Restoring Reality with Spatial Audio and Generative Models

Vivek Jayaram

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:
Steven M. Seitz, Chair
Ira Kemelmacher-Shlizerman, Chair
Shyamnath Gollakota

Program Authorized to Offer Degree:
Computer Science and Engineering

University of Washington

Abstract

Restoring Reality with Spatial Audio and Generative Models

Vivek Jayaram

Co-Chairs of the Supervisory Committee:
Steven M. Seitz
Computer Science and Engineering

Ira Kemelmacher-Shlizerman
Computer Science and Engineering

Everyday, we encounter parts of our reality that are noisy, incomplete, or distorted—whether we're taking a phone call in a noisy cafe, viewing old black and white photos, or trying to better hear a specific instrument in a musical piece. The ability to restore and enhance these signals is essential for improving communication, preserving memories, and creating immersive experiences. This thesis explores new methodologies for signal restoration across both images and audio to address problems like denoising, source separation, inpainting, super-resolution, and colorization. We focus on two different high-level approaches for effective signal restoration: leveraging spatial information to enhance audio experiences and employing generative models to solve challenging inverse problems.

The first part of this work covers spatial audio processing, where we develop novel systems that use multiple microphones for speech enhancement, separation, and spatial audio rendering. We first present a method that uses a multi-microphone array to perform real-time source localization and separation with as many as 4 concurrent speakers. We then showcase custom binaural earbuds that can isolate the wearer's voice on phone calls in real-time using a neural network running on a mobile phone. Finally, we use everyday recordings from those binaural earbuds to improve upon the ability to render sounds to the listener in a spatially

consistent manner.

The second part investigates the use of deep generative models as priors for signal reconstruction. By framing signal restoration tasks as probabilistic inference problems, we apply techniques such as Langevin dynamics and denoising diffusion to efficiently sample from posterior distributions. We first explore the use of score-based models and flow models for source separation of visual signals. We then extend this work to the audio domain by using auto-regressive models for audio source separation and enhancement. Lastly, we present a fast method for solving noisy linear inverse problems using diffusion models.

Together, these contributions demonstrate the powerful ability of both spatial audio processing and deep generative modeling in advancing signal restoration. By tackling practical challenges and pushing the boundaries of theoretical frameworks, this thesis paves the way for more robust communication technologies and immersive audio-visual experiences.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I want to start by thanking my advisors Steve and Ira. They have been integral to my growth as a researcher since the very start of my PhD, providing valuable mentorship, direction, feedback, and support. I first meet Steve when I was in eighth grade at the Washington State MathCounts competition. He was giving the keynote speech showcasing his recently published work on "Building Rome in a Day". It's crazy to think that 10 years later he would become my PhD advisor.

It's amazing to have advisors so excited about new applications of machine learning and determined to usher in the future of technology and communication through new research. Steve and Ira taught me how to present research contributions in a way that would make people excited about the result, through showcasing real-world, seemingly-impossible results that captivate the imagination of readers. Very often I would be stuck on a research problem and a smart insight from one of them would immediately unblock me. That was how my first paper, The Cone of Silence, came into existence; Steve suggested to shift the audio signals before passing them through the neural network, a seemingly simple suggestion that formed the basis of my first paper.

The next person I want to thank is John Thickstun, who was effectively a third advisor to me. He provided a level of close mentorship that was essential to my growth as a researcher, particularly regarding theoretical and algorithmic thinking. When I first started the PhD program, we used to spend hours in whiteboard sessions working through math equations, formulas, drawings, and ideas. I learned most of what I know about generative models from him. It was a shared interest in music that spurred our initial collaboration, but we then turned that into a three-paper collaboration spanning images, speech, and music.

When I first started the program I moved in with two incredible roommates, friends, and research collaborators: Ishan Chatterjee and Maruchi Kim. Coming from hardware backgrounds, they made me rethink what was possible beyond just new machine learning algorithms to new hardware-software combinations. Our ClearBuds collaboration was perhaps the most thorough system I have ever built, and taught me so much about hardware and edge compute. Shyam Gollakota worked closely with us on the ClearBuds project and also served as a valuable resource for discussing and learning about audio related machine learning projects throughout my PhD. He was an incredibly helpful advisor, regularly taking calls at night and on the weekends to help us with the ClearBuds paper. I also want to thank him for being on my reading committee and defense committee.

I want to thank all my amazing labmates, collaborators, and CSE friends, many of whom were dragged into numerous user studies, qualitative evaluations, and lengthy discussions to help me with papers. These include Mek Jenrungrot, Roni Sengupta, Brian Curless, Jingwei Ma, Xiaojuan Wang, Johanna Karras, Luyang Zhu, Keunhong Park, Xuan Luo, Yifan Wang, Ranjay Krishna, Kaitlin Flores, Aleks Holynski, Aditya Kusupati, Matthew Wallingford, Mitchell Wortsman, Vivek Ramanujan, Jason Hoffman, Richard Li, Malek Itani, Bandhav Veluri, Anand Waghmare, Alice Gao, Ben Jones, Amy Zhu, Yuxuan Mei, Shwetak Patel, Sham Kakade, Isaac Tian, JJ Park, Adam Fishman, Benlin Liu, Meng-Li Shish, Mengyi Shan, Roy Or-El, Baback Elmieh, John Akers, David Kessler, Zoran Popovic, Adriana Shulz, Linda Shapiro, and Jeffrey Bilmes. I have learned so much from all of you and have spent countless days, nights, and weekends with you all in the lab, espresso room, anechoic chamber, TGIFs, and GRAIL retreats.

I also thank my parents (the original Dr. Jayaram and Dr. Jayaram) and my sister Harshini, who supported me throughout the entire PhD process. My parents showed me to the rewards and challenges of academic life through their own academic journeys that I witnessed growing up and visiting their research labs regularly. During the Covid lockdowns,

while staying with my family, I completed most of my early papers with their help with recording data, filming demo videos, and giving feedback. The had to deal with me asking for complete silence in the house while I recorded hours of data, sitting for numerous demo video retakes, and evaluating all my weird audio failure cases.

The biggest acknowledgment of all goes to my wife Ellie, who has been an unending pillar of support for me throughout my graduate school experience. She participated in all my user studies, endured nights in the anechoic chamber, and starred in 3 paper teasers and 4 demo videos across Background Matting, Cone of Silence, and HRTF Estimation. She is my biggest champion and has been with me through all the good times and the tough times.

DEDICATION

To my loving wife, Ellie.

Chapter 1

INTRODUCTION

Everyday we encounter parts of our reality that are noisy, incomplete, or distorted. Imagine sitting in a bustling restaurant, straining to hear your conversation partner over the ambient noise. Consider the times you've captured the perfect photo, only to discover an unwanted background distraction that needed to be edited out. Or recall moments on Zoom calls when noise in your environment made it challenging for the other person to hear you clearly.

Restoring and enhancing these signals is a vital problem that spans multiple disciplines and has the potential to significantly impact lives worldwide. This thesis explores new techniques for signal reconstruction and enhancement across the audio and visual domains. We are particularly motivated by improving communication in an increasingly digital world. As more people take calls on-the-go and in noisy environments, the demand for reliable signal processing grows. Additionally, a large number of individuals experience auditory or visual impairments, for whom clearer conversations and higher-quality visuals are especially beneficial. Beyond these practical applications, signal restoration also has creative uses, such as isolating musical components or improving the quality of degraded recordings and photographs.

At its core, signal restoration addresses the challenge of reconstructing a clean signal from noisy or incomplete inputs. The main difficulty lies in the under-constrained nature of this problem, where perfect recovery of the original signal is fundamentally impossible in real-world scenarios (Tropp and Gilbert, 2007). For example, while a recording with minor noise may seem recoverable, imagine attempting to restore a grayscale image to its original color; the true hues present in the scene are unknowable (see Figure 1.1).

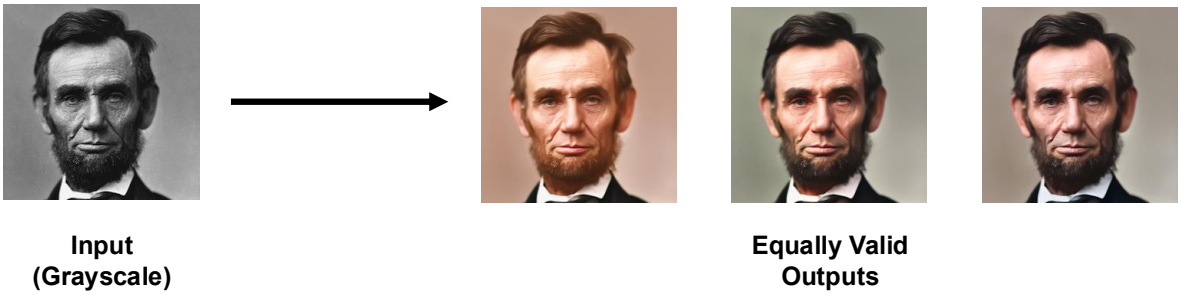


Figure 1.1: We highlight the under-determined nature of signal restoration. We consider a captured photo of Abraham Lincoln which we want to restore using a blue channel constraint proposed in Luo et al. (2021). Using CDIM (see Chapter 9), we generate multiple plausible outputs from a diffusion model. Note that the outputs contain vastly different hues, but each one has a blue channel that exactly matches the captured grayscale photo.

Since mathematical recovery of our clean signal is not possible, we must instead rely on alternate sources of information to best estimate or solve for the desired output. In this thesis we consider two broad approaches to solving signal restoration problems: multi-microphone array processing and generative models. This work is organized into two sections, with the first four chapters focusing on microphone array processing methods and the latter four chapters exploring generative modeling for signal restoration.

Multi-microphone array processing leverages information captured by multiple microphones at different locations to aid in audio restoration tasks. Multi-microphone devices are now widespread, with hundreds of millions of devices sold, from headphones like airpods (Apple Insider, 2021), to VR headsets like Oculus (Statista, 2021), to smart home devices like Alexa (Garfinkle, 2023). For audio enhancement, the target and noise sources are usually at different locations, leading to differences in the captured signal at each microphone. These spatial differences provide valuable cues for isolating the clean source signal. Chapter 2 introduces array processing for speech enhancement and spatial audio, and surveys prior work spanning microphone arrays, beamforming, and head-related transfer functions.

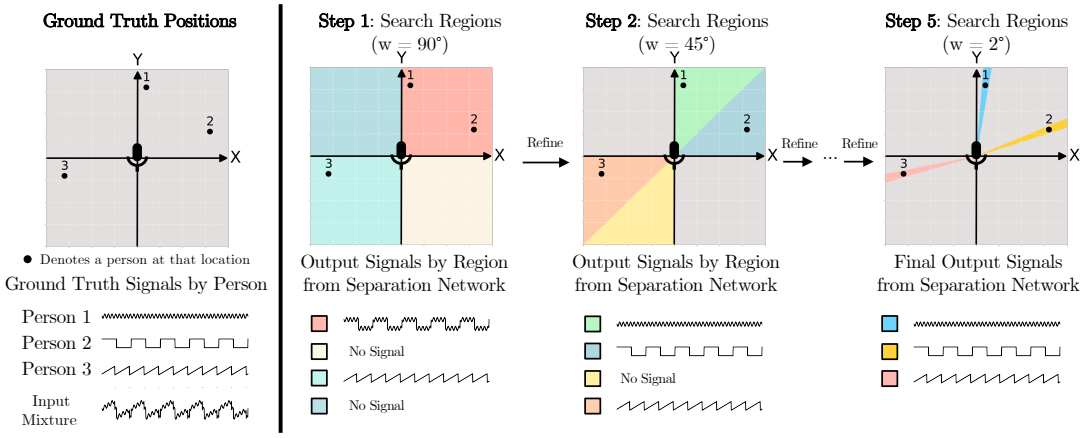


Figure 1.2: The Cone of Silence isolates and localizes an unknown number of speakers using a binary search approach.

In Chapter 3, we introduce the Cone of Silence, a multi-microphone neural network designed to separate and localize multiple speakers in noisy environments. By creating a neural network that isolates sounds from a target spatial direction and window size, we can use a binary search algorithm to separate and localize all speakers in a scene. We show that we can separate 4 simultaneous speakers and locate them spatially to 2° accuracy. This system enables users to focus on a specific speaker, such as the person across the table in a crowded setting. Additionally, we demonstrate its utility for video calls, effectively isolating a speaker's voice against background noise like vacuums, blenders, or other conversations.

We then introduce a new headset called ClearBuds in Chapter 4. These are designed for voice isolation during phone calls in noisy environments. Unlike commercially available earbuds like AirPods, ClearBuds allow time-synchronized streaming of microphone data from both earbuds to a phone, transforming them into a multi-microphone array. This novel system architecture is paired with a binaural speech-enhancement network running directly on the phone. The overall system achieves an end-to-end latency of 109ms, well below the perceptible lag for telephone applications (International Telecommunication Union, 2003). In direct comparisons with the AirPods Pro, ClearBuds achieve better noise suppression based on user evaluation and signal processing metrics. We also run a separate user study

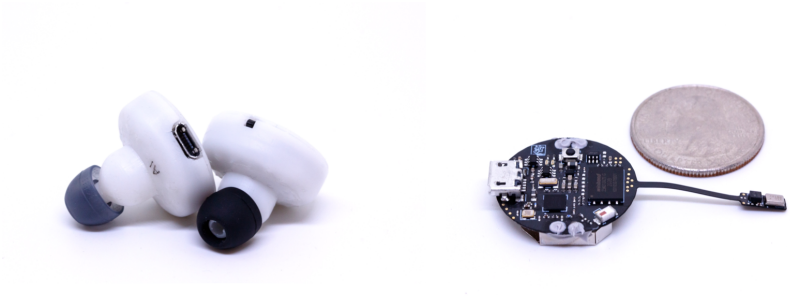


Figure 1.3: An image of the ClearBuds hardware. We built the first binaural headset that could stream time-synchronized audio to a mobile phone.

to validate the strong performance of our noise suppression system against other baseline neural network methods.

In Chapter 5, we use a binaural microphone array inspired by ClearBuds to improve spatial audio experiences for listeners. Accurate spatial audio relies on the Head-Related Transfer Function (HRTF) (Møller, 1992), which describes how sound is modified by a listener's head and ears based on source location. Personalized HRTFs are crucial for immersive audio experiences across AR, VR, gaming and music (Begault and Trejo, 2000; Wenzel et al., 1993). We address the problem of measuring a listener's personalized HRTF, using just a binaural headset and the sounds they hear in-the-wild. By analyzing how sounds change as the user rotates their head through different environments, we can accurately estimate their personalized HRTF. Our results show that our predicted in-the-wild HRTFs closely match ground-truth HRTFs measured in an anechoic chamber. Furthermore, listening studies demonstrate that our personalized HRTFs significantly improve sound localization and reduce front-back confusion in virtual environments.

The second half of this thesis focuses on generative models, which address the underdetermined nature of signal recovery without relying on multiple capture devices. Instead, these methods leverage large datasets of clean signals to train state-of-the-art generative models. Suppose we have a corrupted measurement y, we can use Bayes' rule to guide our

signal restoration:

$$p(x|y) \propto p(x) \cdot p(y|x) \tag{1.1}$$

Generative models act as a prior over the distribution of target signals, $p(x)$, guiding the signal reconstruction towards plausible outputs. By combining these priors with a likelihood function, $p(y|x)$, that assess adherence to the noisy inputs, we can generate samples from the posterior distribution of target signals $p(x|y)$. These methods benefit from the decoupling of the generative modeling task and the signal recovery task; we can use state-of-the-art unconditional models with little or no modification and sample from them in a conditional way for signal reconstruction tasks. We consider flow-based models (Kingma and Dhariwal, 2018a; Kingma et al., 2016) , score-based models (Song and Ermon, 2019; Song et al., 2020b), autoregressive models (Oord et al., 2016; Salimans et al., 2017), and diffusion models (Ho et al., 2020; Song et al., 2020a), all of which can be used for signal recovery. In Chapter 6, we provide an overview of generative modeling techniques and their application to signal restoration.

In Chapter 7, we introduce Bayesian Annealed Signal Informed Separation (BASIS) to address visual source separation. BASIS is a novel sampling algorithm that combines pre-trained generative models with annealed Langevin sampling (Welling and Teh, 2011). We demonstrate BASIS on the separation of up to 4 overlaid images from the MNIST dataset (LeCun, 1998). We also frame image colorization as a source separation task in order to apply our algorithm. This chapter makes a methodological contribution on small-scale images to demonstrate the effectiveness of Markov-chain Monte Carlo methods for posterior sampling. We also show the inherent ambiguity of source separation and propose a new evaluation methodology that considers likelihood under the prior distribution instead of adherence to the original signal.

Chapter 8, extends the BASIS algorithm to posterior sampling from autoregressive models. We first show how to smooth discrete autoregressive models, such as WaveNet (van den Oord et al., 2016a) in the audio domain and PixelCNN++ (Salimans et al., 2017) in the visual domain, in order to computer gradients through the model. This enables us to use these au-

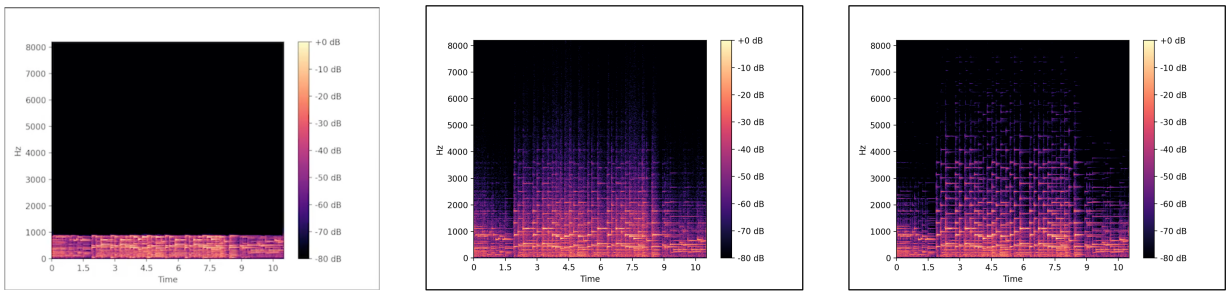


Figure 1.4: Audio super-resolution using the PnF sampling method in Chapter 8. Given an audio samples with 92% of the samples removed, our method can fill in the missing samples. Qualitative audio examples can be found at https://grail.cs.washington.edu/projects/pnf-sampling/.

toregressive models as bayesian priors for linear inverse problems, such as source separation, super-resolution, and inpainting. This autoregressive posterior sampling achieves competitive performance against supervised, task-specific baselines, and we also share qualitative examples on in-the-wild audio restoration problems. And example of audio super-resolution is shown in Figure 1.4.

Finally, Chapter 9 introduces constrained diffusion implicit models (CDIM), a new method for solving noisy linear inverse problems with diffusion models. Compared to previous diffusion based methods for inverse problems, such as DPS (Chung et al., 2022b), CDIM make several key improvements. First, we greatly reduce the inference time by using a learned step size to accelerate convergence. Second, we incorporate Kullback-Liebler divergence (Kullback, 1951) to handle non-gaussian noise distributions like Poisson noise. Lastly, we guarantee exact reconstruction of the input observation, even for out of distribution images. We demonstrate CDIM's efficacy across datasets like FFHQ (Karras et al., 2019) and ImageNet (Russakovsky et al., 2015), solving tasks such as inpainting, super-resolution, deblurring, and sparse recovery.

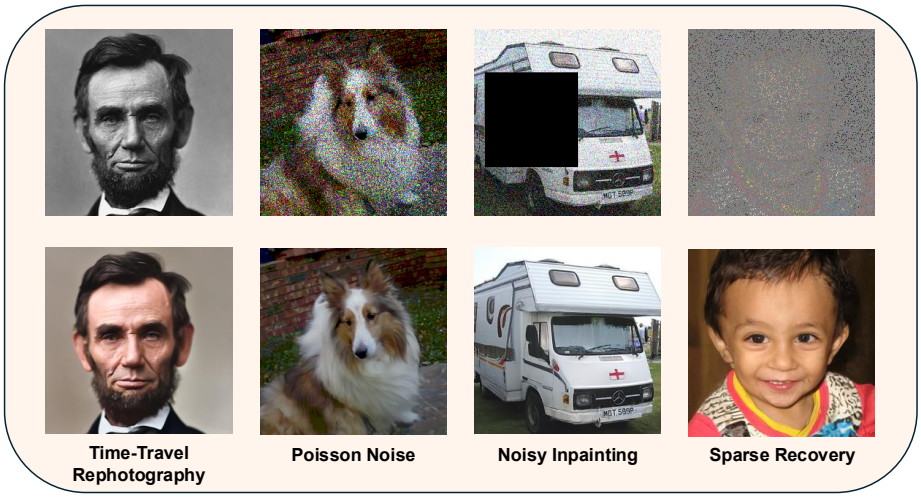


Figure 1.5: Examples of various image restoration tasks using Constrained Diffusion Implicit Models, presented in Chapter 9.

1.1 Publications and Co-authorship

The work presented in this thesis is the result of numerous research collaborations. Various work has been co-authored by peers and has appeared in conference publications and other dissertations. Below I share the publications and contributions of the papers contained in this thesis. An asterisk * denotes equal contribution.

Chapter 3: The Cone of Silence

- **Contributions:** This paper was the result of an equal contribution effort with Teerapat Jenrungrot. I was primarily responsible for the algorithmic development, data collection, and model training, while Teerapat conducted experiments, wrote a significant portion of the paper, and helped frame the contribution.

- **Publication:** The Cone of Silence: Speech Separation by Localization. Teerapat Jenrungrot*, Vivek Jayaram*, Steven M. Seitz, Ira Kemelmacher-Shlizerman. Neural Information Processing Systems (NeurIPS) 2020. Oral Presentation (Top 1% of papers).

Chapter 4: ClearBuds

- **Contributions:** This paper resulted from a cross-disciplinary collaboration with Ishan Chatterjee and Maruchi Kim. I was responsible for the neural network development, data collection, model training, and experimental results. Ishan and Maruchi developed the hardware and firmware and also conducted experiments and user studies.

- **Publication:** ClearBuds: Wireless Binaural Earbuds for Learning-Based Speech Enhancement. Ishan Chatterjee*, Maruchi Kim*, Vivek Jayaram*, Shyamnath Gollakota, Ira Kemelmacher Shlizerman, Shwetak Patel, Steven M. Seitz. International Conference on Mobile Systems, Applications and Services (MobiSys) 2022.

Chapter 5: HRTF Estimation in-the-Wild

- **Contributions:** This work was primarily authored by myself. I conducted all experiments, data collection, model training, and writing.

- **Publication:** HRTF Estimation in-the-Wild. Vivek Jayaram, Ira Kemelmacher-Shlizerman, Steven M. Seitz. ACM Symposium on User Interface Software and Technology (UIST). 2023.

Chapters 7, 8, and 9: Generative Models for Signal Restoration

- **Contributions:** These chapters are the result of three papers produced during a close collaboration with John Thickstun. We are co-authors of the first two papers (Source Separation and PnF Sampling), and John shifted to an advisory role for the last paper (CDIM). In all of these works, I was primarily responsible for model training, data collection, and experiments, while John led the efforts on theoretical contribution and algorithmic development.

- **Publication 1:** Source Separation with Deep Generative Priors. Vivek Jayaram*, John Thickstun*. International Conference on Machine Learning (ICML). 2020.

- **Publication 2:** Parallel and Flexible Sampling from Autoregressive Models via Langevin Dynamics. Vivek Jayaram*, John Thickstun*. International Conference on Machine Learning (ICML). 2021.

- **Publication 3:** Constrained Diffusion Implicit Models. Vivek Jayaram, Ira Kemelmacher-Shlizerman, Steven M Seitz, John Thickstun. (Arxiv Preprint). 2025.

Chapter 2

MULTI-MICROPHONE ARRAY PROCESSING AND SPATIAL AUDIO

The human auditory system benefits fundamentally from using two ears to perceive and interpret sounds (Blauert, 1997). Similarly, audio processing algorithms can harness spatial information for more powerful isolation and processing of sounds within an environment. This chapter introduces the principles of multi-microphone array processing and spatial audio algorithms, establishing a foundation for the methods detailed in the following three chapters. Specifically, we explore a spatial audio system designed to accomplish two complementary objectives: isolating desired sound sources in noisy environments and rendering the desired sound in an immersive way for the listener. These goals are critical across a range of applications, including teleconferencing, hearing aids, virtual reality, and gaming. For instance, imagine sitting in a bustling restaurant with friends. A well-designed spatial audio system would not only isolate their voices but also position them realistically in the soundscape, creating a natural and engaging auditory experience (see Figure 2.1).

We begin by introducing multi-microphone arrays in Section 2.1 and describe their hardware configurations, spatial cues, and practical considerations. Following this, we explore beamforming in Section 2.2, a cornerstone algorithm for array signal processing, discussing its principles, limitations, and modern extensions. Finally, we explore the Head-Related Transfer Function (HRTF) in Section 2.3 and discuss it's importance and methods of estimation. This foundation will prepare readers to appreciate the novel methods and contributions presented in later chapters.

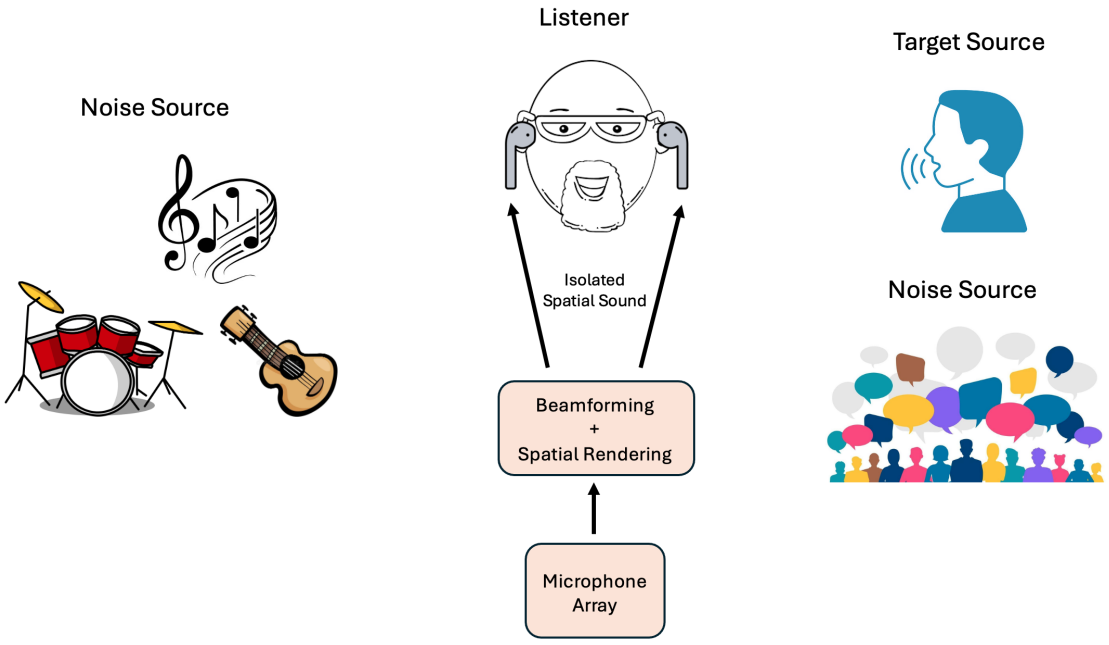


Figure 2.1: Illustration of a spatial audio system for real-time speech enhancement. The system must perform two tasks: isolating the source of interest and rendering it spatially to the listener. The microphone array may either be in the earbuds of the listener or an external capture device.

2.1 Multi-Microphone Capture

Multi-microphone capture devices form the backbone of array signal processing algorithms, enabling the capture of sound from multiple spatial locations. By distributing microphones across a device or environment, these arrays can exploit spatial diversity to better analyze, separate, and enhance sound sources. Early innovations in multi-microphone systems include the seminal work on beamforming for linear arrays (Van Trees, 2002) and microphone array processing for robust speech enhancement (Brandstein and Ward, 2001). More recently, these principles have been integrated into commercial products like smart speakers and headphones, making spatial audio technologies widely accessible.

The physical arrangement of microphones is a critical design consideration, directly impacting the performance and capabilities of spatial audio systems. Common configurations include:

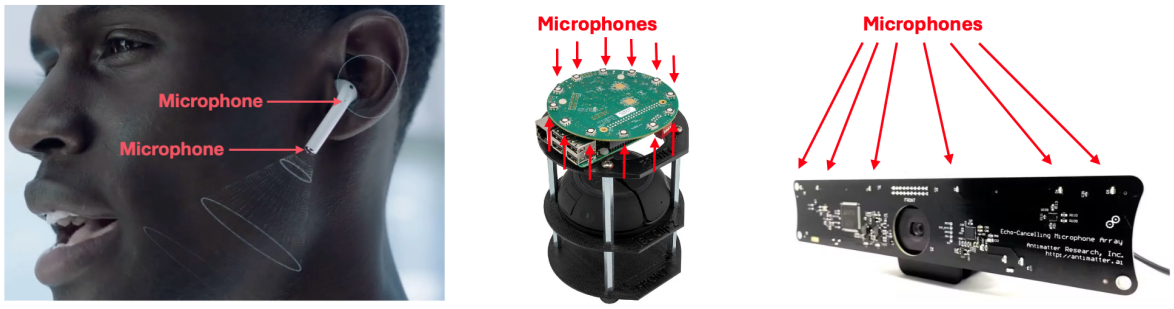
- Linear Arrays: Often used in applications like smart speakers or conference microphones, linear arrays align microphones in a straight line to capture sound with directional resolution along a single plane (Johnson and Dudgeon, 1993). See Figure 2.2 (c).

- Circular Arrays: Circular microphone arrays provide omnidirectional coverage, making them suitable for 360-degree audio applications. Applications in immersive audio and robotics have been extensively studied, with notable contributions by Rafaely (2004) in spherical microphone array processing and McCowan et al. (2001) for meeting transcription systems. See Figure 2.2 (b).

- Binaural Arrays: Found in headsets or earbuds, binaural arrays mimic the human head's geometry, capturing sound as it would naturally reach the ears (Møller, 1992). An early example includes dummy head recordings Blauert (1997). This configuration is particularly valuable for spatial audio rendering and HRTF estimation.

- Planar Arrays: Planar arrays extend the concept of linear arrays into two dimensions, arranging microphones in a grid-like structure. This configuration enables high spatial resolution in both azimuth and elevation, making it ideal for advanced beamforming and 3D sound capture. Planar arrays have been widely studied for applications such as video conferencing, environmental monitoring, and acoustic imaging (Benesty et al., 2008).

Consumer devices have popularized multi-microphone systems in recent years (see Figure 2.2). For instance, Apple's AirPods Pro utilize multiple microphones on each earbud for better voice capture and active noise cancellation. Note that AirPods are not a true binaural



(a) (b) (c)

Figure 2.2: Various examples of multi-microphone arrays in real-world products. (a) Apple AirPods Pro contain two microphones on each earbud. (b) Amazon Alexa contains twelve microphones in a circular array. (c) Acusis S Linear Microphone Array, a video conferencing product with a camera and six microphones in a linear configuration.

microphone array given that only one earpiece (e.g. left or right) can capture and stream audio at a time. Another popular example is Amazon's Echo smart speakers, which use circular microphone arrays to detect voice commands even in noisy environments. These systems demonstrate the practical utility of multi-microphone arrays, though challenges such as device size constraints, power consumption, and microphone synchronization remain critical considerations.

2.1.1 Interaural Time and Level Differences

Multi-microphone systems exploit spatial diversity to enhance audio processing. Two key spatial audio cues—Interaural Time Difference (ITD) and Interaural Level Difference (ILD)—enable the localization and separation of sound sources:

- ITD: This describes the time delay between when a sound reaches one microphone and another, due to distance traveled at the speed of sound. This is manifested by a phase offset in the two signals. The pioneering work in Jeffress (1948) laid the

groundwork for understanding ITD as a key mechanism in human auditory localization. An early spatial filtering algorithm that relied on ITD was developed in Brandstein and Silverman (1997).

- ILD: This describes the difference in sound intensity between microphones, arising due to head-shadowing effects or spatial attenuation. Rayleigh (1907) first described the difference in sound intensity between ears based on where a sound was located. Blauert (1997) later extensively documented the role of ILD in human spatial hearing through recordings with a dummy head.

2.1.2 Geometry and Spatial Resolution

Another crucial aspect of multi-microphone arrays is their spatial resolution, defined as how finely the array can distinguish sound sources located at different spatial positions. Spatial resolution depends on factors including the microphone geometry, the sampling rate of the system, the angle and distance of the sources, and the frequency content of the sound signals.

For two microphones separated by a distance d, such as shown in in Figure 2.3, time delay between the captured signals is

$$\Delta t = \frac{d \sin \theta}{c} \tag{2.1}$$

Here c is the speed of sound, and θ represents the azmuthal angle of the sound source relative to the broadside of the microphone array axis (see Figure 2.4). A source aligned with the microphones ($\theta = 90°$) provides the maximum time differences, while one equidistant to each microphone ($\theta = 0°$) results in no detectable time delay. Spatial resolution referes to the smallest angular separation $\Delta\theta$ that allows discrimination between two sources located at at θ_1 and $\theta_2 = \theta_1 + \Delta\theta$. With perfect sensors, infinite sampling rate, and no noise, you could resolve sources separated by arbitrarily small angles ($\Delta\theta \to 0$) because the wavefronts are mathematically distinct. In practice, perfect resolution is unattainable due

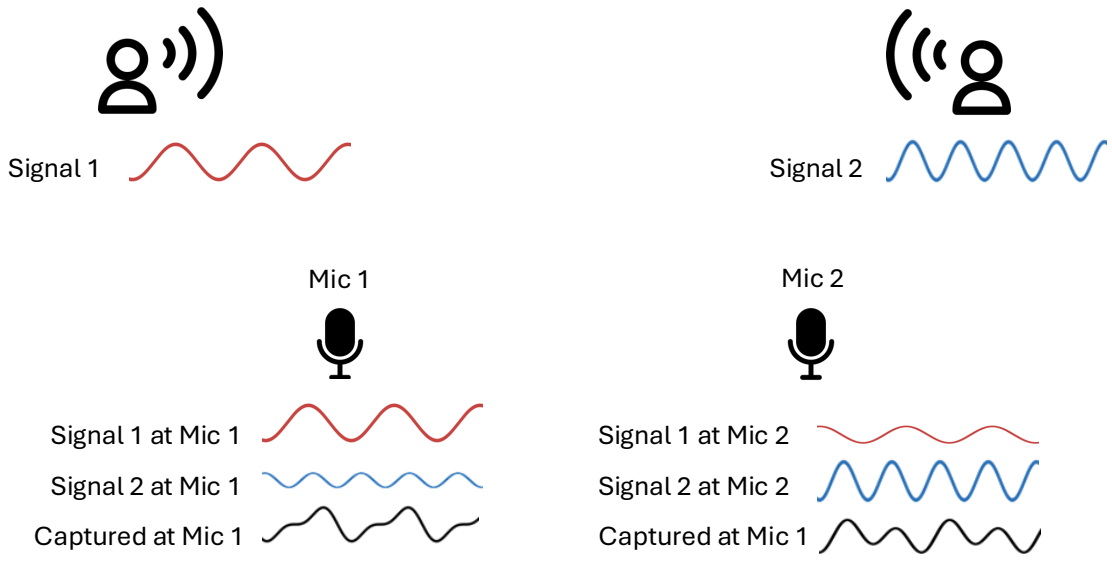


Figure 2.3: An illustration of how interaural time differences and level differences create spatial cues at different microphones. Suppose we have two people talking, illustrated with Signal 1 and Signal 2. Those signals experience different attenuation and phase changes when they arrive at the two microphones. This results in a different overall captured signal at each microphone, providing information about the location and content of each speaker.

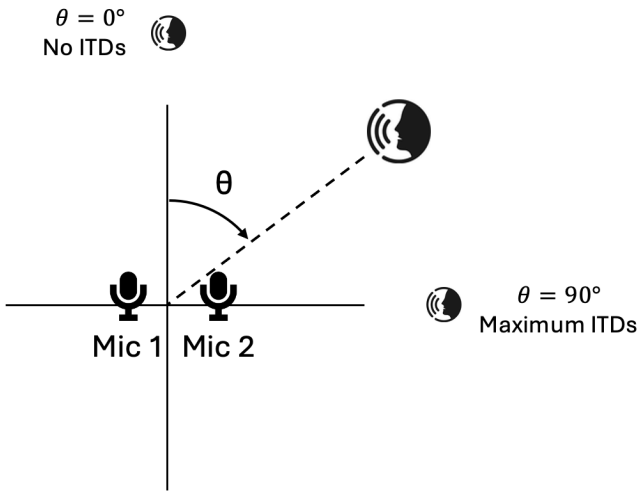


Figure 2.4: We measure the angle θ relative to the broadside axis of the microphone array. At $\theta = 0°$, there are no ITDs since the sound reaches the microphones at the same time. At $\theta = 90°$, the sound experiences the maximum ITDs across the two microphones.

to sensor imperfections, finite sampling rates, and environmental noise. Assume there is a smallest difference in the wavefronts Δt_{min} that we can realistically detect. This is inversely proportional to the frequency of the sound wave as high frequency sources produce more pronounced wavefront differences across the same distance. Mathematically, we can express this as

$$\Delta t_{min} \propto \frac{1}{f} = \frac{\lambda}{c} \tag{2.2}$$

where λ is the wavelength of the sound, and we have used the fact that $c = \lambda f$ for propagating waves. Substituting this back into equation 2.1 and considering a linear array with N microphones each d distance apart, we have

$$\Delta\theta \propto \sin^{-1}\left(\frac{\lambda}{dN}\right) \tag{2.3}$$

This metric, known as the beamwidth or spatial resolution, improves with more microphones (N), larger microphone spacing (d), and higher sound frequencies (lower λ). However, to avoid spatial aliasing, the microphone spacing must satisfy $d < \lambda/2$ (Alien and Berkley, 1976). For example, the microphone array in The Cone of Silence, with a diameter of 6.4 cm, can resolve sounds up to approximately 27 kHz without spatial aliasing. However the microphone array in ClearBuds, given an average human inter-ear distance of 15cm (Errede, 2002), could only resolve sounds up to 11kHz.

For a circular microphone array with N microphones evenly distributed along a radius R, the spatial resolution is similarly determined:

$$\Delta\theta \propto \sin^{-1}\left(\frac{\lambda}{2\pi RN}\right) \tag{2.4}$$

Larger radii, more microphones, and higher frequencies yield better resolution, analogous to linear arrays. Circular arrays also benefit from isotropic coverage, making them ideal for 360-degree sound capture (Rafaely, 2004).

The microphone sampling rate also affects spatial resolution by limiting the smallest detectable time delay Δt_{min}. For a system with a sampling rate of 48 kHz, each sample

represents a time interval of $\frac{1}{48000} \approx 20.8\mu s$. Consider two microphones spaced 6.4cm apart. A sound source at $\theta_1 = 0°$ ($\Delta t_1 = 0$) can only be distinguished from a source at $\theta_2 \approx 8°$, where the ITDs results in a single-sample delay ($\Delta t_2 = 20.8\mu s$). Higher sampling rates or increased microphone separation reduce this minimum angle, enhancing spatial resolution.

This interplay of geometric factors, frequency content, and sampling rate highlights the inherent trade-offs in designing multi-microphone systems. Practical implementations must balance these considerations against constraints such as size, power consumption, and cost.

2.2 Beamforming

Beamforming is a fundamental technique in spatial audio and microphone array processing, aiming to extract sound from a particular direction while suppressing interference from others. Although a traditional definition of beamforming only considers combining signals for constructive and destructive interference, we refer to it here more broadly as the task of directional sound capture and isolation. This task is similar to the formulations which we later explore in The Cone of Silence (Chapter 3) and ClearBuds (Chapter 4). By leveraging the spatial configuration of multiple microphones, beamformers apply direction-dependent filtering that coherently sums signals from a desired source and attenuates noise and competing sources. Over the years, beamforming strategies have evolved from simple fixed filters to sophisticated adaptive and subspace-based methods, each addressing different challenges in complex acoustic environments. We discuss a variety of algorithms from the delay-and-sum beamformer (Van Veen and Buckley, 1988b; Cox et al., 1987) to adaptive approaches like the Minimum Variance Distortionless Response (MVDR) beamformer (Capon, 1969) and its generalization, the Linearly Constrained Minimum Variance (LCMV) beamformer (Frost, 1972a; Cox et al., 1987; Brandstein and Ward, 2001). We then consider a subspace-based approach known as MUSIC (Schmidt, 1986), which employs high-resolution techniques for direction-of-arrival (DOA) estimation and can inform source separation strategies as well. Finally we discuss deep learning extensions of beamforming which rely on CNNs, RNNs, and transformers (Xiao et al., 2016; Swietojanski et al., 2014; Bai et al., 2024).

2.2.1 Delay-and-Sum Beamformer

One of the earliest and most intuitive beamforming strategies is the delay-and-sum beamformer, which aligns signals in time before summation (see Figure 2.5). Suppose we have an array of N microphones, and a plane wave arriving from a direction of interest. If $x_n(t)$ is the signal at the n-th microphone, the array output $y(t)$ of a delay-and-sum beamformer is given by

$$y(t) = \frac{1}{N} \sum_{n=1}^{N} x_n(t - \tau_n), \tag{2.5}$$

where τ_n is the delay applied to the n-th microphone signal to compensate for the source direction. By choosing τ_n to align the wavefront from the desired source direction, constructive interference for that source occurs while signals from other directions add less coherently, thus providing spatial selectivity (Van Veen and Buckley, 1988b).

While elegant in its simplicity, delay-and-sum beamforming lacks robustness and frequency-dependent control. It does not explicitly minimize noise or interference, leading to limited performance in reverberant and noisy environments. Nonetheless, it laid the groundwork for more advanced methods by illustrating the basic principle of spatial filtering via phase alignment. The Cone of Silence builds on delay-and-sum beamforming by aligning the desired source before using a deep neural network. In addition, ClearBuds uses the same principle that the desired speaker is temporally aligned across microphones and the interfering sources are not.

2.2.2 Minimum Variance Distortionless Response (MVDR) Beamformer

Adaptive beamformers improve upon fixed designs by continuously adjusting their filter coefficients to enhance performance under changing conditions. The MVDR beamformer (Capon, 1969) was a seminal adaptive solution that minimizes output power while ensuring that a signal arriving from a desired direction is passed without distortion. In the frequency domain, let $\mathbf{x}(\omega) = [X_1(\omega), X_2(\omega), \ldots, X_N(\omega)]^T$ denote the vector of microphone signals and $\mathbf{w}(\omega) = [W_1(\omega), W_2(\omega), \ldots, W_N(\omega)]^T$ be the beamforming weights. The MVDR beamformer

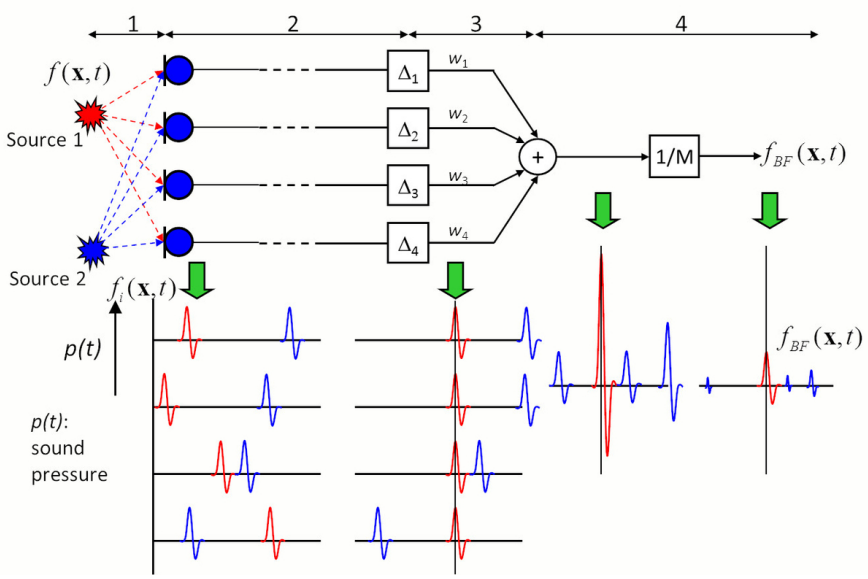


Figure 2.5: An illustration of delay-and-sum beamforming from Meyer et al. (2011). There are two sound sources, designated with red and blue. We calculate the necessary time offsets so that the red source signal is aligned across the microphones. Then we sum the signals, producing constructive intereference with the desired source and destructive interference with the undesired blue source. Finally we normalize the output to the original scale.

solves

$$\min_{\mathbf{w}(\omega)} \mathbf{w}^H(\omega)\mathbf{R}(\omega)\mathbf{w}(\omega) \quad \text{subject to} \quad \mathbf{w}^H\mathbf{a}(\theta) = 1 \tag{2.6}$$

where $\mathbf{R}(\omega)$ is the spatial covariance matrix of the input signals $\mathbf{x}$ at frequency ω, and $\mathbf{a}(\theta)$ is a steering vector corresponding to the desired direction θ. The closed form solution for the beamforming weights is:

$$\mathbf{w}(\omega) = \frac{\mathbf{R}^{-1}(\omega)\mathbf{a}(\theta)}{\mathbf{a}^H(\theta)\mathbf{R}^{-1}(\omega)\mathbf{a}(\theta)} \tag{2.7}$$

The MVDR solution is well-known for its excellent interference rejection, but it can be sensitive to estimation errors in the covariance matrix and microphone miscalibration (Brandstein and Ward, 2001; Doclo and Moonen, 2007). Subsequent enhancements, such as diagonal loading and other regularization techniques, were introduced to improve numerical stability and robustness against model mismatch (Cox et al., 1987). Still, traditional MVDR beamforming is fundamentally limited by assumptions of stationarity and linear signal models, which may not hold in non-stationary or highly reverberant conditions.

2.2.3 LCMV Beamforming

The LCMV beamformer generalizes MVDR by imposing multiple linear constraints, allowing the beamformer to simultaneously maintain distortionless response in one or more directions and, if desired, create destructive interference toward known noise sources. The Frost beamformer (Frost, 1972a) was an early and influential implementation of the LCMV approach.

At a high level, the LCMV beamformer solves:

$$\min_{\mathbf{w}(\omega)} \mathbf{w}^H(\omega)\mathbf{R}(\omega)\mathbf{w}(\omega) \quad \text{subject to} \quad \mathbf{w}^H\mathbf{A} = \mathbf{d}^H \tag{2.8}$$

where $\mathbf{A} = [\mathbf{a}_1(\theta_1), .., \mathbf{a}_L(\theta_L)]$ is a matrix of L steering vectors and $\mathbf{d} = [d_1, d_2, ..., d_L]^T$ is the desired responses for the corresponding directions. Solving this constrained optimization

yields weights that suppress interference while preserving the desired signal. Similar to MVDR, LCMV can be solved easily in closed form with the solution

$$\mathbf{w}(\omega) = \mathbf{R}^{-1}\mathbf{A}(\mathbf{A}^H\mathbf{R}^{-1}\mathbf{A})^{-1}\mathbf{d} \tag{2.9}$$

(Frost, 1972a; Cox et al., 1987; Brandstein and Ward, 2001). By supporting multiple constraints, the LCMV approach can produce more flexible spatial filters than MVDR, for example steering multiple simultaneous look-directions or forming suppression at known interference locations. However, like MVDR, LCMV methods can suffer when the covariance matrix is poorly estimated, and they assume that the directional properties of the scene are well-characterized.

2.2.4 MUSIC Beamformer

While MVDR and LCMV rely on direct optimization with steering vectors, the MUltiple SIgnal Classification (MUSIC) algorithm (Schmidt, 1986) takes a different approach. MUSIC is a high-resolution direction-of-arrival (DOA) estimation technique that uses the eigenstructure of the spatial covariance matrix. Instead of solving a direct minimization problem for beamformer weights, MUSIC projects potential steering vectors onto the noise subspace of the array covariance matrix. Peaks in the resulting pseudospectrum indicate directions of incoming signals.

Let $\mathbf{R}$ be the array covariance matrix. By performing an eigen-decomposition:

$$\mathbf{R} = \mathbf{U}_s\mathbf{\Lambda}_s\mathbf{U}_s^H + \mathbf{U}_n\mathbf{\Lambda}_n\mathbf{U}_n^H, \tag{2.10}$$

we separate the signal subspace ($\mathbf{U}_s$) associated with the largest eigenvalues ($\mathbf{\Lambda}_s$) from the noise subspace ($\mathbf{U}_n$) associated with the smallest eigenvalues ($\mathbf{\Lambda}n$). For a candidate direction θ, let $\mathbf{a}(\theta)$ be the corresponding steering vector. The MUSIC pseudospectrum is given by:

$$P_{\text{MUSIC}}(\theta) = \frac{1}{\mathbf{a}^H(\theta)\mathbf{U}_n\mathbf{U}_n^H\mathbf{a}(\theta)}. \tag{2.11}$$

Sharp peaks of $P_{\text{MUSIC}}(\theta)$ identify DOAs of the impinging signals. Once the DOAs are estimated, a beamformer can be constructed to enhance signals from those directions. MUSIC is known for its high-resolution DOA estimation, outperforming conventional methods like delay-and-sum in scenarios with closely spaced sources. However, it requires accurate covariance matrix estimation and a clear separation between signal and noise subspaces, and it can be computationally more complex than simpler beamforming approaches.

2.2.5 Limitations of Traditional Beamforming

The beamforming methods discussed here—delay-and-sum, MVDR, LCMV, and MUSIC—represent core approaches in spatial audio processing. These methods represent early approaches to isolating sounds based on their direction, an important concept explored in this thesis. Although these classical beamformers have proven effective in numerous applications, they face fundamental limitations when confronted with highly dynamic and reverberant environments, or non-stationary sources. Their reliance on accurate steering vectors, and linear signal models can lead to performance degradation if assumptions are violated or in scenarios with fewer microphones. Moreover, these methods do not inherently account for the semantic content of the underlying sound sources, a key insight that deep learning based methods exploit.

2.2.6 Moving Beyond Traditional Methods: Data-Driven and Neural Beamformers

The recent surge in machine learning has spawned beamforming strategies that learn directly from data rather than relying purely on analytic models. Early attempts integrated statistical models of speech and noise into the optimization frameworks (Gannot et al., 2001; Doclo and Moonen, 2006), culminating in multi-channel Wiener filtering strategies that combine beamforming and spectral enhancement.

More recently, deep learning-based beamforming has emerged as a powerful approach (Heymann et al., 2016; Sainath et al., 2017; Nakatani et al., 2020; Abd El-Fattah et al., 2014). Neural beamformers often operate in the time-frequency domain, learning masks

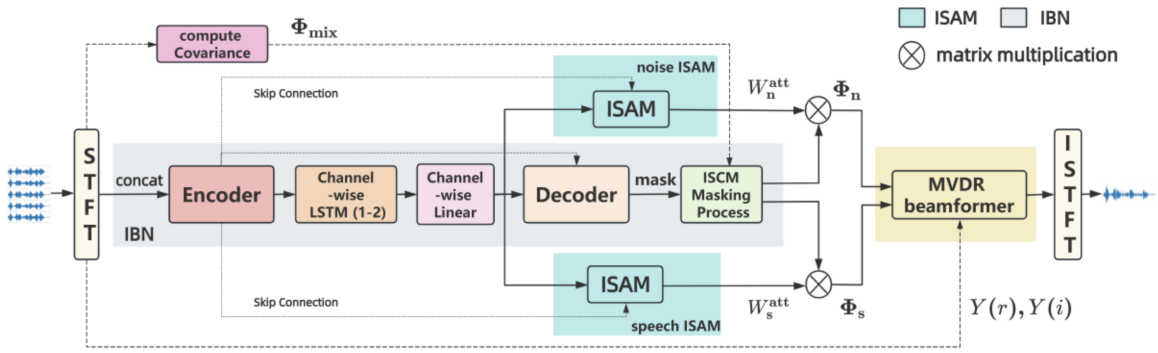


Figure 2.6: A transformer-based deep learning beamformer (Bai et al., 2024). This architecture combines deep learning with traditional methods like covariance estimation and MVDR.

that emphasize desired signals. For example, a neural beamformer may first estimate a complex spectral mask $\mathbf{M}(\omega)$ for each microphone channel, and then produce beamforming weights by:

$$\mathbf{w}_{\mathrm{NN}}(\omega) = \mathrm{NN}\big(\mathbf{X}(\omega)\big), \tag{2.12}$$

where $\mathbf{X}(\omega)$ are the complex spectra of the input signals and $\mathrm{NN}(\cdot)$ denotes a deep neural network which can learn a detailed function from data. The output weights $\mathbf{w}_{\mathrm{NN}}(\omega)$ are then multiplied with the original signal to isolate the source of interest while suppressing interfering sources. Such a network can learn non-linear, complex mappings that outperform linear beamformers in challenging conditions. It can implicitly handle aspects like reverberation, varying source positions, and microphone imperfections by training on diverse datasets. These methods also take advantage of the latest deep learning architectures, using convolutional neural nets (CNNs) (Swietojanski et al., 2014), recurrent neural nets (RNNS) (Xiao et al., 2016), and Transformers (see Figure 2.6) (Bai et al., 2024; Guo et al., 2023). Yet, these advantages do not come without trade-offs. Data dependency, model interpretability, computational costs, and generalization to unseen conditions remain major hurdles. Models often need large and diverse training sets to handle the variability of real-world scenarios. Ensuring reliable performance when microphone configurations, acoustic conditions, or user

anatomies deviate from those seen during training is an ongiong challenge.

2.3 Head-Related Transfer Function

We now consider the second challenge of our spatial audio system which is to render the isolated sound source in a realistic and spatially consistent manner. The Head-Related Transfer Function (HRTF) is a crucial element for this. It encapsulates the frequency-dependent filtering effects imposed by the human anatomy—specifically, the head, torso, and pinnae—on incoming sound waves before they reach the eardrums (see Figure 2.7. Understanding and accurately modeling HRTFs is essential for creating spatially realistic audio experiences, as they provide the spectral cues that allow humans to localize and externalize sound sources in three-dimensional space (Møller et al., 1995b; Algazi et al., 2001; Blauert, 1997). With precise HRTFs, virtual audio systems can convincingly place sound sources anywhere around a listener, enhancing the immersion in applications such as virtual and augmented reality, gaming, teleconferencing, and hearing-aid fitting (Begault and Trejo, 2000; Wenzel et al., 1993).

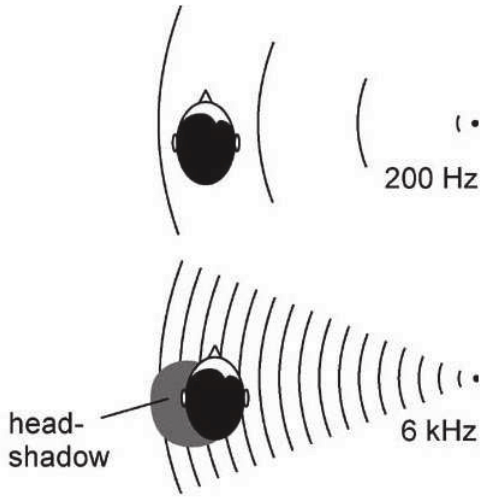


Figure 2.7: An illustration of the head filtering effect of the HRTF from (Wall et al., 2008). We show that different frequencies of signals may be attenuated by the head differently

2.3.1 Importance of HRTFs in Spatial Audio

From a perceptual standpoint, the human auditory system infers sound direction from subtle spectral notches and peaks introduced by the interplay of the outer ear shape, head size, and torso reflections (Rayleigh, 1907; Blauert, 1997). These frequency-dependent alterations differentiate sound reaching the left and right ears, enabling listeners to judge the azimuth, elevation, and distance of sound sources. Accurate HRTFs allow binaural rendering systems to synthesize these cues, effectively "tricking" the listener's brain into perceiving sound originating from specific spatial locations without requiring a full surround-sound setup.

Mathematically, an HRTF can be expressed in the frequency domain as a complex-valued function:

$$H(\omega, \phi, \theta) = \frac{P_{\text{eardrum}}(\omega, \phi, \theta)}{P_{\text{free-field}}(\omega)}, \tag{2.13}$$

where $P_{\text{eardrum}}(\omega, \phi, \theta)$ is the sound pressure at the listener's eardrum for a plane wave arriving from azimuth ϕ and elevation θ, and $P_{\text{free-field}}(\omega)$ is the reference pressure of the same wave measured in the free-field (i.e., without the listener present) (Wightman and Kistler, 1989). In practice, HRTFs are often measured at discrete spatial sample points and frequencies, forming a high-dimensional data set known as the HRTF dataset.

2.3.2 Traditional Methods of HRTF Estimation

Traditionally, HRTFs have been obtained through direct acoustic measurements using specialized equipment and controlled environments. In a standard measurement procedure, a listener or a mannequin (such as a KEMAR dummy head) is placed in an anechoic chamber (see Figure 2.8), and test signals (often exponential sweeps or maximum length sequences) are played from a dense grid of loudspeakers surrounding the subject (Algazi et al., 2001; Møller, 1992). The recorded responses at the subject's ears are then processed to yield measured HRTFs.

These measurement campaigns are time-consuming, expensive, and difficult to personalize. Even with dummy heads that approximate the average human ear shape and head size,

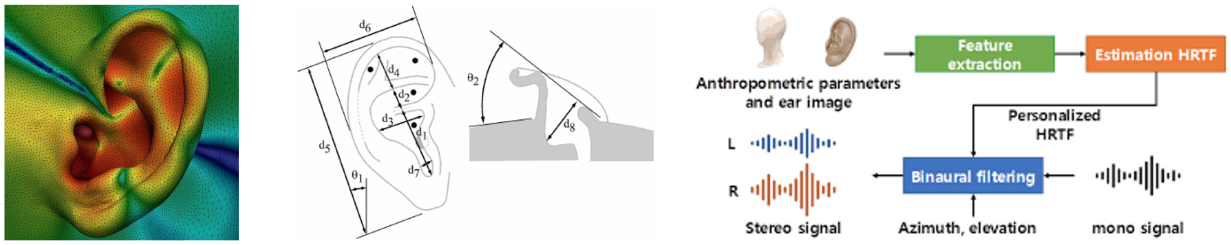
Figure 2.8: A user getting their HRTF measured in an anechoic chamber.Image from Southampton (2003).

the resulting HRTFs may not perfectly match an individual's unique anatomy. Inter-subject variability can cause perceptible differences in the spatial accuracy of binaural rendering, with some listeners experiencing inside-the-head localization or reduced externalization due to the mismatch between their personal HRTF and the measured one (Middlebrooks, 1999b).

To reduce this burden, interpolation and spatial sampling techniques have been introduced to estimate HRTFs at unmeasured directions from a limited set of measured points. These methods rely on assumptions about the smoothness and spatial coherence of HRTFs (Cheng and Wakefield, 1999), but often still require extensive initial measurements.

2.3.3 Modern Methods of HRTF Estimation

Recent approaches have sought to alleviate the limitations of traditional HRTF measurement by developing computational, data-driven, and personalized estimation techniques. Modern methods typically fall into a few categories.



(a) (b) (c)

Figure 2.9: Several modern HRTF estimation approachs. (a) Mesh2HRTF (Ziegelwanger et al., 2015) which simulates the acoustic field through a mesh of the ear and head. (b) CIPIC database (Algazi et al., 2001) which contains many anthropometric measurements for nearest neighbor comparison or principal component analysis. (c) Neural network HRTF prediction from ear images and anthropometric measurements (Lee et al., 2019).

Anthropometric and Geometric Modeling One line of research relates physical dimensions of the listener's head, torso, and pinnae to expected HRTF patterns. Parametric models approximate the external ear as a set of geometric shapes, and then use numerical acoustic simulations (e.g., Boundary Element Methods) to compute HRTFs (Katz, 2001; Huttunen et al., 2014; Ziegelwanger et al., 2015). These methods are computationally intensive but can yield highly accurate results for a given individual. Anthropometric regression approaches fit functions that map measurements of the listener's ear and head shape to HRTF features, thereby enabling personalization without direct acoustic measurements (Zotkin et al., 2003, 2002). While promising, these methods depend on the availability and quality of anthropometric data, and their accuracy can degrade for individuals whose physical features deviate significantly from the training set.

Database-driven Methods The availability of large HRTF databases, such as the CIPIC dataset (Algazi et al., 2001), has enabled statistical approaches to HRTF estimation. Principal component analysis (PCA) and other dimensionality-reduction techniques extract the

main variations in HRTFs across subjects and directions, allowing interpolation or approximation of a new subject's HRTF from limited reference measurements or anthropometric cues (Middlebrooks, 1999b). These methods work well for general trends but struggle with outliers or unique anatomical features. Furthermore, the reliance on existing datasets limits the diversity of represented populations, which could bias results toward specific demographics. Expanding and diversifying HRTF datasets remains an open challenge.

Machine Learning and Deep Learning Approaches Building on the success of neural networks in complex regression tasks, deep learning models have been employed to predict a subject's HRTF from a small number of acoustic measurements (Javeri et al., 2022), anthropometric features (Zhi et al., 2022; Zhao et al., 2022), or even ear images (Mohan et al., 2003). Neural networks can learn non-linear mappings between input features (e.g., ear shape parameters) and HRTF responses at arbitrary source directions, reducing the need for dense measurements. These methods can also integrate perceptual feedback, adjusting the predicted HRTFs until they yield the best subjective match for the user. However, their success hinges on the availability of large, diverse training datasets and robust validation protocols. Generalization to unseen populations or edge cases is a persistent challenge, as is the interpretability of the models. Additionally, computational costs for training and inference may limit their applicability in real-time systems.

Considering these challenges, Chapter 5 presents HRTF estimation in-the-wild, which creates a personalized HRTF just from sounds captured by earbuds in everyday environments. As a listener rotates their head through real-world environments, we build up a personalized HRTF that avoids the need for complex scans or specialized sound labs.

Chapter 3

THE CONE OF SILENCE

The ability of humans to separate and localize sounds in noisy environments is a remarkable phenomenon known as the "cocktail party effect." However, our natural ability only goes so far – we may still have trouble hearing a conversation partner in a noisy restaurant or during a call with other speakers in the background. One can imagine future earbuds or hearing aids that *selectively* cancel audio sources that you don't want to listen to. As a step towards this goal, we introduce a deep neural network technique that can be steered to any direction at run time, cancelling all audio sources outside a specified angular window, aka *cone of silence* (CoS) (CoS, 1960).

But how do you know what direction to listen to? We further show that this directionally sensitive CoS network can be used as a building block to yield simple yet powerful solutions to 1) sound localization, and 2) audio source separation. Our experimental evaluation demonstrates state of the art performance in both domains. Furthermore, our ability to handle an unknown number of potentially moving sound sources combined with fast performance represents additional steps forward in generality. Audio demos can be found at our project website.[1]

We are particularly motivated by the recent increase of multi-microphone devices in everyday settings. This includes headphones, hearing aids, smart home devices, and many laptops. Indeed, most of these devices already employ directional sensitivity both in the design of the individual microphones and in the way they are combined together. In practice however, this directional sensitivity is limited to either being hard tuned to a fixed range of directions (e.g., cardioid), or providing only limited attenuation of audio outside that

[1] https://grail.cs.washington.edu/projects/cone-of-silence/

range (e.g., beam-forming). In contrast, our CoS approach enables true *cancellation* of audio sources outside a specified angular window that can be specified (and instantly changed) in software.

Our approach uses a novel deep network that can separate sources in the waveform domain within any angular region $\theta \pm \frac{w}{2}$, parameterized by a direction of interest θ and angular window size w. For simplicity, we focus only on azimuth angles, but the method could equally be applied to elevation as well. By exponentially decreasing w, we perform a binary search to separate and localize all sources in logarithmic time (Figure 3.1). Unlike many traditional methods that perform direction based separation, we can also ignore background source types, such as music or ambient noise. Qualitative and quantitative results show state-of-the-art performance and a direct applicability to a wide variety of real world scenarios. Our key contribution is a logarithmic time algorithm for simultaneous localization and separation of speakers, particularly in high levels of noise, allowing for arbitrary number of speakers at test time, including more speakers than seen during training. We strongly encourage the reader to view our supplementary results for a demo of our method and audio results.

3.1 Related Works

Source separation has seen tremendous progress in recent years, building upon the multi-microphone capture techniques (Section 2.1) and traditional beamforming approaches (Section 2.2) discussed earlier. While traditional beamforming methods provide a foundation for directional audio processing, learning-based approaches have increasingly demonstrated superior performance.

Recent approaches have moved beyond the traditional beamforming methods described in Section 2.2 to incorporate learning techniques that improve over classical methods such as Cardoso (1998); Nesta et al. (2010). These learning-based approaches fall into several categories:

Unsupervised source modeling methods train models for each source type and apply them to mixtures using techniques like Non-negative Matrix Factorization (NMF) Raj et al.

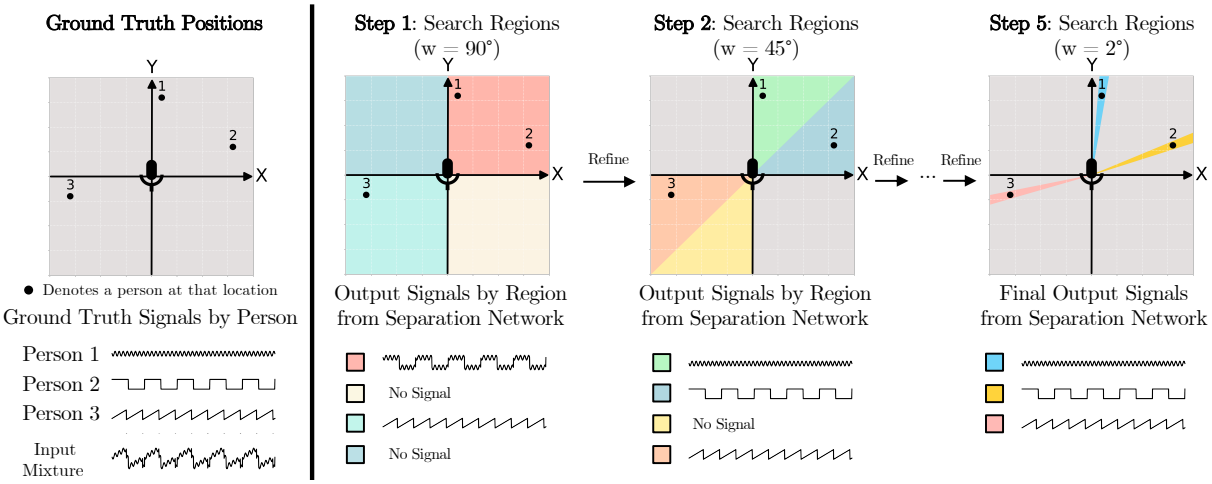


Figure 3.1: Overview of *Separation by Localization* running binary search on an example scenario with 3 sources. Each panel shows the spatial layout of the scene with the microphone array located at the center. During Step 1, the algorithm performs separation on candidate regions of 90°. The quadrants with no sound get suppressed and disregarded. The algorithm continues doing separation on smaller partitions of candidate regions until reaching the final step where the angular window size is 2°.

(2010); Mohammadiha et al. (2013b), clustering (Tzinis et al., 2019; Sawada et al., 2010; Luo et al., 2017), or Bayesian methods (Itakura et al., 2018; Jayaram and Thickstun, 2020; Benaroya et al., 2005). Supervised source modeling methods train models for each source using annotated isolated signals of specific source types, such as pitch information for music (Bertin et al., 2010). Separation-based training methods Halperin et al. (2018); Jansson et al. (2017); Nugraha et al. (2016) employ deep neural networks to learn source separation from mixtures given ground truth signals as training data, commonly known as the mix-and-separate framework.

Two significant trends have emerged in source separation research that inform our work. The first is waveform-domain processing. Direct operation on audio waveforms (Stoller et al., 2018b; Luo and Mesgarani, 2018, 2019) has shown performance improvements over frequency-

domain spectrogram techniques Hershey et al. (2016); Xu et al. (2018); Weng et al. (2015); Tzinis et al. (2019); Yoshioka et al. (2018); Chen et al. (2018); Zhang and Wang (2017). The second is multi-channel approaches. Methods leveraging multi-microphone arrays (Yoshioka et al., 2018; Chen et al., 2018; Gu et al., 2020) and binaural recordings (Zhang and Wang, 2017; Han et al., 2020a) outperform single-channel techniques. Multimodal approaches that combine audio-visual techniques Zhao et al. (2018); Rouditchenko et al. (2019) have also shown promise.

Sound localization, often posed as Direction of Arrival (DOA) estimation (Grondin and Glass, 2019; Nadiri and Rafaely, 2014; Pavlidi et al., 2013), complements the spatial audio capture techniques discussed in Section 2.1. Beyond the subspace methods like MUSIC (Schmidt, 1986) described earlier, other popular approaches include additional beamforming variants (DiBiase, 2000), alternative subspace methods (Wang and Kaveh, 1985; Di Claudio and Parisi, 2001; Yoon et al., 2006), and sampling-based techniques (Pan et al., 2017). Building on the neural beamforming concepts introduced in Section 2.2, deep neural networks have also been employed for multi-source DOA estimation (He et al., 2018; Adavanne et al., 2018).

A key challenge in real-world applications is that the number of speakers is often unknown or varies over time. Many methods require a priori knowledge about the number of sources (Luo and Mesgarani, 2018; Luo et al., 2020a). Recent deep learning approaches that address separation with an unknown number of speakers include Higuchi et al. (2017), Takahashi et al. (2019), and Nachmani et al. (2020), though these methods typically employ additional models to predict speaker count, and some like Nachmani et al. (2020) use different separation models for different numbers of speakers.

While the directional beamforming approaches discussed in Section 2.2 provide a foundation for source separation, they typically require knowing the direction of interest in advance (Adel et al., 2012). Without a known DOA for each source, these methods must perform a computationally expensive linear sweep of the entire angular space. Previous work on joint localization and separation includes expectation-maximization approaches Traa and

Smaragdis (2014); Mandel et al. (2009); Asano and Asoh (2004); Dorfan et al. (2015); Deleforge et al. (2015); Mandel et al. (2007), Directional NMF Traa et al. (2015), and Bayesian inference methods based on inter-microphone phase differences Johnson et al. (2018). Our method improves on these by combining deep learning in the waveform domain with an efficient logarithmic search strategy, addressing the limitations of both traditional beamforming and previous joint approaches.

3.2 Method

In this section we describe our Cone of Silence network for angle based separation. The target angle θ and window size w are learned independently; Separation at θ is handled entirely by a pre-shift step, while an additional network input is used to produce the window of size w. We also describe how to use the network for *separation by localization* via binary search.

Problem Formulation: Given a known-configuration microphone array with M microphones and $M > 1$, the problem of M-channel source separation and localization can be formulated in terms of estimating N sources $\mathbf{s}_1, \ldots, \mathbf{s}_N \in \mathbb{R}^{M \times T}$ and their corresponding angular position $\theta_1, \ldots, \theta_N$ from an M-channel discrete waveform of the mixture $\mathbf{x} \in \mathbb{R}^{M \times T}$ of length T, where

$$\mathbf{x} = \sum_{i=1}^{N} \mathbf{s}_i + \mathbf{bg}. \tag{3.1}$$

Here **bg** represents the background signal, which could be a point source like music or diffuse-field background noise without any specific location.

In this paper we explore circular microphone arrays, but we also describe possible modifications to support linear arrays. The center of our coordinate system is always the center of the microphone array, and the angular position of each source, θ_i, is defined based on this coordinate system. In the problem formulation we assume the sources are stationary, but we describe how to handle potentially moving sources in Section 3.3.5. In addition, we only focus on separation and localization by azimuth angle, meaning that we assume the

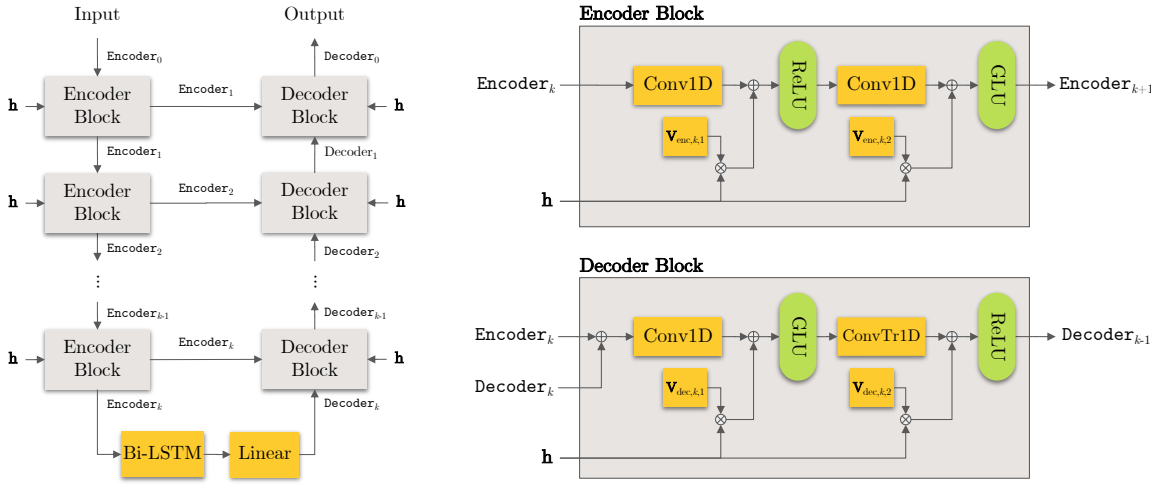


Figure 3.2: (*left*) our network architecture, (*top-right*) the encoder block, (*bottom-right*) the decoder block. In all diagrams, **h** refers to the global conditioning variable corresponding to an angular window size w.

sources have roughly the same elevation angle. As we show in the experimental section, this assumption is valid for most real world scenarios.

3.2.1 Cone of Silence Network (CoS)

We propose a network that performs source separation given an angle of interest θ and an angular window size w. The network is tasked with separating speech only coming from azimuthal directions between $\theta - \frac{w}{2}$ and $\theta + \frac{w}{2}$ and disregarding speech coming from other directions. In the following sections we describe how to create a network with this property. Figure 3.2 shows our proposed network architecture. θ and w are encoded in a shifted input $\mathbf{x}'$ and a one-hot vector $\mathbf{h}$ as described in Section 3.2.1 and Section 3.2.1 respectively.

Base Architecture

Our CoS network is adapted from the Demucs architecture Défossez et al. (2019a), a music separation network, which is similar to the Wave U-Net architecture Jansson et al. (2017).

We extend the original Demucs network to our problem formulation by modifying the number of input and output channels to match the number of microphones.

There are several reasons why this base architecture is well suited for our task. As mentioned in Section 3.1, networks that operate on the raw waveform have been recently shown to outperform spectrogram based methods. In addition, Demucs was specifically designed to work at sampling rates as high as 44.1kHz, while other speech separation networks operate at rates as low as 8kHz. Although human speech can be represented at lower sampling rates, we find that operating at higher sampling rates is beneficial for capturing small time difference of arrivals between the microphones. This would also allow our method to be extended to high resolution source types, like music, where a high sampling rate is necessary.

Target Angle θ

In order to make the network output sources from a specific target angle θ, we use a shifted mixture $\mathbf{x}' \in \mathbb{R}^{M \times T}$ based on θ. We found that that this worked better than trying to directly condition the network based on both θ and w. $\mathbf{x}'$ is created as follows: by calculating the time difference of arrival at each microphone, we shift each channel in the original signal $\mathbf{x}$ such that signals coming from angle θ are temporally aligned across all M channels in $\mathbf{x}'$.

We use the fact that the time differences of arrival (TDOA) between the microphones are primarily based on the azimuthal angle for a far-field source. This assumption is valid when the sources are roughly on the same plane as the microphone array Valin et al. (2003). Let c be the speed of sound, sr be our sampling rate, p_θ be the position of a far-field source at angle θ, and $d(\cdot, \cdot)$ be a simple Euclidean distance. The TDOA in samples for the source to reach the i-th microphone is

$$T_{\texttt{delay}}(p_\theta, \texttt{mic}_i) = \left\lfloor \frac{d(p_\theta, \texttt{mic}_i)}{c} \cdot sr \right\rfloor \tag{3.2}$$

In our experiments, we chose $\texttt{mic}_0$ as our canonical position, meaning that $\mathbf{x}'_0 = \mathbf{x}_0$ and all other channels $\mathbf{x}'_i$ are shifted to align with $\mathbf{x}'_0$.

$$\mathbf{x}'_i = \texttt{shift}(\mathbf{x}_i, T_{\texttt{delay}}(p_\theta, \texttt{mic}_0) - T_{\texttt{delay}}(p_\theta, \texttt{mic}_i)) \qquad i = 1, \ldots, M-1 \tag{3.3}$$

`shift` is a 1-D shift operation with one sided zero padding. This idea is similar to the first step of a Delay and Sum Beamformer Johnson and Dudgeon (1992). We then train the network to output sources which are temporally aligned in $\mathbf{x}'$ while ignoring all others. For $M > 2$, these shifts are unique for a specific angle θ, so sources from other angles will not be temporally aligned. If $M = 2$ or the mic array is linear, then sources at angles θ and $-\theta$ have the same per-channel shift leading to front-back confusion.

Angular Window Size w

Although the network trained on shifted inputs as described in Section 3.2.1 can produce accurate output for a given angle θ, it requires prior knowledge about the target angle θ for each source of interest. In addition, real sources are not perfect point sources and have finite width, especially in the presence of slight movements.

To solve these problems, we introduce a second variable, an angular window size w which is passed as a global conditioning parameter to the network. This angular window size facilitates the application of this network for a fast binary search approach. It also allows the localization and separation of moving sources. By using a larger angular window size and a smaller temporal input waveform, it is possible to localize and separate moving sources within that window.

Motivated by the global conditioning framework in WaveNet Oord et al. (2016), we use a one-hot encoded variable $\mathbf{h}$ to represent different window sizes. In our experiments, we use $\mathbf{h}$ of size 5 corresponding to window sizes from the set $\{90°, 45°, 23°, 12°, 2°\}$. By passing $\mathbf{h}$ to the network with our shifted input $\mathbf{x}'$, we can explicitly make the network separate sources from the region $\theta \pm \frac{w}{2}$. We embed $\mathbf{h}$ to all encoder and decoder blocks in the network, using a learning linear projection $\mathbf{V}_{\cdot,k,\cdot}$ as shown in Figure 3.2. Formally, the equations for the

encoder block and decoder block can be written as follows:

$$\begin{aligned}\texttt{Encoder}_{k+1} = \text{GLU}(&\mathbf{W}_{\text{encoder},k,2} * \text{ReLU}(\mathbf{W}_{\text{encoder},k,1} * \texttt{Encoder}_k \\ &+ \mathbf{V}_{\text{encoder},k,1}\mathbf{h}) + \mathbf{V}_{\text{encoder},k,2}\mathbf{h}),\end{aligned} \tag{3.4}$$

$$\begin{aligned}\texttt{Decoder}_{k-1} = \text{ReLU}(&\mathbf{W}_{\text{decoder},k,2} *^{\top} \text{GLU}(\mathbf{W}_{\text{decoder},k,1} * (\texttt{Encoder}_k + \texttt{Decoder}_k) \\ &+ \mathbf{V}_{\text{decoder},k,1}\mathbf{h}) + \mathbf{V}_{\text{decoder},k,2}\mathbf{h}).\end{aligned} \tag{3.5}$$

The notation $\mathbf{W}_{\cdot,k,\cdot} * \mathbf{x}$ denotes a 1-D convolution between the weights for the layer of an encoding/decoding block at level k and an input $\mathbf{x}$. The notation $*^{\top}$ denotes a transposed convolution operation. Empirically we found that passing $\mathbf{h}$ to every encoder and decoder block worked significantly better than passing it to the network only once. Evidence that the CoS network learns the desired window size is presented in Figure 3.4.

Network Training

Consider an input mixture $\mathbf{x}$ of N sources $\mathbf{s}_1, \ldots, \mathbf{s}_N$ with the corresponding locations $\theta_1, \ldots, \theta_N$ along with a target angle θ_t and window size w. The network is trained with the following objective function:

$$\mathcal{L}(\mathbf{x}; \mathbf{s}_1, \ldots, \mathbf{s}_N, \theta_t, w) = \left\| \tilde{\mathbf{x}}' - \sum_{i=1}^{N} \mathbf{s}'_i \cdot \mathbb{I}\left(\theta_t - \frac{w}{2} \leq \theta_i < \theta_t + \frac{w}{2}\right) \right\|_1 \tag{3.6}$$

where $\mathbf{x}'$ and $\mathbf{s}'_i$ are the shifted signals of the input mixture and ground truth signal as described in Section 3.2.1 based on the target angle θ_t. $\tilde{\mathbf{x}}'$ is the output of the network using the shifted signal $\mathbf{x}'$ and the angular window w. $\mathbb{I}(\cdot)$ is an indicator function, indicating whether $\mathbf{s}_i$ is present in the region $\theta_t \pm \frac{w}{2}$. If no source is present in the region $\theta_t \pm \frac{w}{2}$, the training target is a zero tensor $\mathbf{0}$.

3.2.2 Localization and Separation via Binary Search

By starting with a large window size w and decreasing it exponentially, we can perform a binary search of the angular space in logarithmic time, while separating the sources simultaneously. More concretely, we start with our initial window size $w_0 = 90°$, our initial target

angles $\theta_0 = \{-135°, -45°, 45°, 135°\}$, and our observed M-channel mixture $\mathbf{x} \in \mathbb{R}^{M \times T}$. In the first pass we run the network $\text{CoS}(\mathbf{x}', w_0)$ for all $\theta_0^i \in \theta_0$. This first step is the quadrant based separation illustrated in Step 1 of Figure 3.1. Because regions without sources will produce empty outputs, we can discard large angular regions early on with a simple cutoff. We then regress on a smaller window size, $w_1 = 45°$ and the new candidate regions $\theta_1 = \bigcup_i \{\theta_0^i \pm \lfloor \frac{w_0}{2} \rfloor\}$ for θ_0^i regions with high energy outputs from $\text{Cos}(\mathbf{x}', w_0)$. We continue to regress on smaller window sizes until reaching the desired resolution. The complete algorithm is written below and shown in Figure 3.1.

Algorithm 1 Separation by Localization via Binary Search

Require: M-channel input mixture $\mathbf{x} \in \mathbb{R}^{M \times T}$ and microphone positions $\{\texttt{mic}_i\}_{i=0}^{M-1}$

Ensure: Separated signals and their locations

Initialize L, $w_{0,\ldots,L-1}$, and θ_0.
for $\ell = 0$ to $L - 1$ **do**
    $\theta_{\ell+1} \leftarrow \{\}$
    for all $\theta_\ell^i \in \theta_\ell$ **do**
        $\mathbf{x}' \leftarrow \textsc{PreShift}(\mathbf{x}, \theta_\ell^i, \{\texttt{mic}_j\}_{j=0}^{M-1})$
        $\tilde{\mathbf{x}}' \leftarrow \text{CoS}(\mathbf{x}', w_\ell)$
        if $\tilde{\mathbf{x}}' \neq \emptyset$ **then**
            Update $\theta_{\ell+1}$ by adding $\theta_\ell^i \pm \lfloor \frac{w_\ell}{2} \rfloor$.
        end if
    end for
end for
return Non-max suppression on sources at $\theta \in \theta_L$

To avoid duplicate outputs from adjacent regions, we employ a non-maximum suppression step before outputting the final sources and locations. For this step, we consider both the angular proximity and similarity between the sources. If two outputted sources are physically

close and have similar source content, we remove the one with the lower source energy. For example, for outputs $(\tilde{\mathbf{x}}'_i, \theta_i)$ and $(\tilde{\mathbf{x}}'_j, \theta_j)$ with $\|\tilde{\mathbf{x}}'_i\| > \|\tilde{\mathbf{x}}'_j\|$, we remove $(\tilde{\mathbf{x}}'_j, \theta_j)$ if $|\theta_i - \theta_j| < \epsilon_\theta$ and $\|\tilde{\mathbf{x}}'_i - \tilde{\mathbf{x}}'_j\| < \epsilon_x$.

3.2.3 Runtime Analysis

Suppose we have N speakers and the angular space is discretized into $r = \frac{360^\circ}{w}$ angular bins. The binary search algorithm runs for at most $\mathcal{O}(\log r)$ steps and requires at most $\mathcal{O}(N)$ forward passes on every step. Thus, the total number of forward passes is $\mathcal{O}(N \log r)$ while a linear sweep always runs in $\mathcal{O}(r)$ forward passes.

In most cases, $N \ll r$, so the binary search is clearly superior. For instance, when operating at a 2° resolution, the average number of forward passes our algorithm takes to separate 2 voices in the presence of background is 32.64, compared to 180 for a linear sweep. A forward pass of the network on a single GPU takes .03s for a 3s input waveform at 44.1kHz, meaning that the binary search algorithm in this scenario could keep up with real-time while the linear search could not.

3.3 Experiments

In this section, we explain our synthetic dataset and manually collected real dataset. We show numerical results for separation and localization on the synthetic dataset and describe qualitative results on the real dataset.

3.3.1 Synthetic Dataset

Numerical results are demonstrated on synthetically rendered data. To generate the synthetic dataset, we create multi-speaker recordings in simulated environments with reverb and background noises. All voices come from the VCTK dataset Veaux et al. (2016b), and the background samples consist of recordings from either noisy restaurant environments or loud music. The train and test splits are completely independent and there are no overlapping

identities or samples. We chose VCTK over other widely used datasets like LibriSpeech Panayotov et al. (2015) and WSJ0 Garofalo et al. (2007) because VCTK is available at a high sampling rate of 48 kHz compared to 16 kHz as offered by others. In the supplementary materials, we show results and comparisons with lower sampling rates.

To synthesize a single example, we create a 3-second mixture at 44.1 kHz by randomly selecting N speech samples and a background segment and placing them at arbitrary locations in a virtual room of a randomly chosen size. We then simulate room impulse responses (RIRs) using the image source method Allen and Berkley (1979) implemented in the `pyroomacoustics` library Scheibler et al. (2018). To approximate a diffuse-field background noise, the background source is placed further away, and the RIR for the background is generated with high-order images, causing indirect reflections off room walls Vorländer (2007). All signals are convolved with the corresponding RIRs and rendered to a 6-channel circular microphone array ($M = 6$) of radius 2.85 in (7.25 cm). The volumes of the sources are chosen randomly in order to create challenging scenarios; the input SDR is between -16 dB and 0 dB for most of the dataset. For training our network, we use 10,000 examples with N chosen uniformly between 1 and 4, inclusively, at random, and for evaluating we use 1,000 examples with N dependent on the evaluation task.

3.3.2 Source Separation

To evaluate the source separation performance of our method, we create mixtures consisting of 2 voices ($N = 2$) and 1 background, allowing comparisons with deep learning methods that require a fixed number of foreground sources.

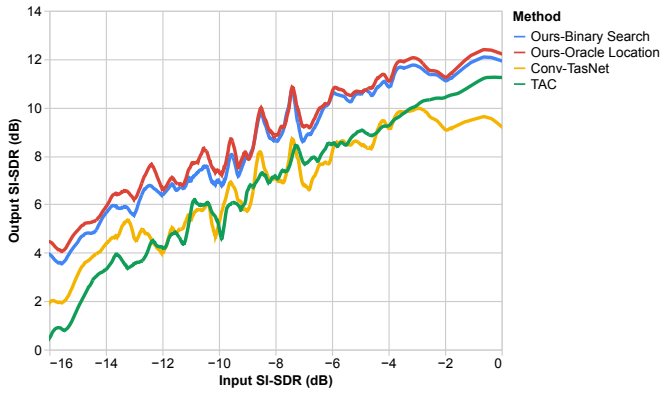
We use the popular metric *scale-invariant signal-to-distortion ratio* (SI-SDR) Le Roux et al. (2019). When reporting the increase from the input to output SI-SDR, we use the label SI-SDR improvement (SI-SDRi). For deep learning baselines in the waveform domain we chose TAC Luo et al. (2020a), a recently proposed neural beamformer, and a multi-channel extension of Conv-TasNet Luo and Mesgarani (2019), a popular speech separation network. For this multi-channel Conv-TasNet, we changed the number of input channels to match the

Table 3.1: Separation Performance. Larger SI-SDRi is better. The SI-SDRi is computed by finding the median of SI-SDR increases from Figure 3.3.

Method	**SI-SDRi** (dB)
Waveform-based	
Conv-TasNet Luo and Mesgarani (2019)	15.526
TAC Luo et al. (2020a)	15.121
Ours - Binary Search	**17.059**
Ours - Oracle Location	17.636
Spectrogram-based	
Oracle IBM Stöter et al. (2018); Wang (2005)	13.359
Oracle IRM Stöter et al. (2018); Liutkus and Badeau (2015)	4.193
Oracle MWF Stöter et al. (2018); Duong et al. (2010)	8.405

number of microphones in order to process the full mixture. To compare with spectrogram based methods, we use oracle baselines based on the time-frequency representation like Ideal Binary Mask (IBM), Ideal Ratio Mask (IRM), and Multi-channel Wiener Filter (MWF). For more details on oracle baselines, please refer to Stöter et al. (2018). Table 3.1 and Figure 3.3 show the comparison between our proposed system and the baseline systems.

Notice that our method strongly outperforms the best possible results obtainable with spectrogram masking, and is slightly better than recent deep-learning baselines operating on the waveform domain. Furthermore, our network can accept explicitly known source locations (given by *Ours-Oracle Location*), allowing the separation performance to improve further when the source positions are given.



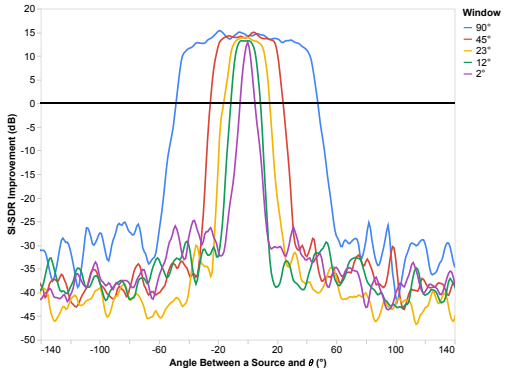


Figure 3.3: (left) Input SI-SDR vs Output SI-SDR for waveform based methods. Some methods are not shown to improve the visibility.

Figure 3.4: (right) Evidence that the network amplifies voices between $\theta \pm \frac{w}{2}$ and suppresses all others.

3.3.3 Source Localization

To evaluate the localization performance of our method, we explore two variants of the same dataset in Section 3.3.2. The first set contains 2 voices and 1 background, exactly as in the previous section, and the second contains 2 voices with no background, a slightly easier variation. Here, we report the CDF curve of the angular error, i.e., the fraction of the test set below a given angle error.

For baselines, we choose popular methods for direction of arrival estimation, both learning-based systems He et al. (2018) and learning-free systems Schmidt (1986); DiBiase (2000); Wang and Kaveh (1985); Di Claudio and Parisi (2001); Pan et al. (2017); Yoon et al. (2006). For the scenario with 2 voice sources and 1 background source, we let the learning-free baseline algorithms localize 3 sources and choose the 2 sources closest to the ground truth voice locations. This is a less strict evaluation that does not require the algorithm to distinguish between a voice and background source. For the learning-based method He et al. (2018), we retrained the network separately for each dataset in order to predict the 2 voice loca-

Table 3.2: Localization Performance

Method	**Median Angular Error**	
	2 Voices	2 Voices + BG
Learning-free		
MUSIC Schmidt (1986)	82.5°	36.8°
SRP-PHAT DiBiase (2000)	6.2°	46.4°
CSSM Wang and Kaveh (1985)	30.1°	36.3°
WAVES Di Claudio and Parisi (2001)	16.4°	32.1°
FRIDA Pan et al. (2017)	6.9°	18.5°
TOPS Yoon et al. (2006)	2.4°	11.5°
Learning-based		
MLP-GCC He et al. (2018)	1.0°	41.5°
Ours	**2.1°**	**3.7°**

tions, even in the presence of a background source. Figure 3.5 shows the CDF plots for both scenarios.

Our method shows state-of-the-art performance in the simple scenario with 2 voices, but some baselines show similar performance to ours. However, when background noise is introduced, the gap between our method and the baselines increases greatly. Traditional methods struggle, even when evaluated less strictly than ours, and MLP-GCC He et al. (2018) cannot effectively learn to distinguish a voice location from background noise.

3.3.4 Varying Number of Speakers

To show that our method generalizes to an arbitrary number of speakers, we evaluate separation and localization on mixtures containing up to 8 speakers with no background. We train

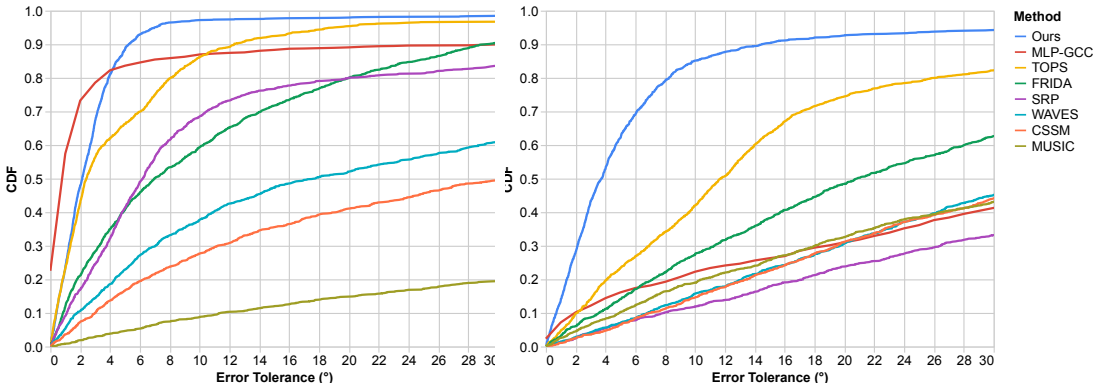


Figure 3.5: Localization Performance: (*Left*) error tolerance curve on mixtures of 2 voices, (*right*) error tolerance curve on mixtures with 2 voices and 1 background.

the network with mixtures of 1 background and up to 4 voices and evaluate the separation results with median SI-SDRi and the localization performance with median angular error. For a given number of speakers N, we take the top N outputs from the network and find the closest permutation between the outputs and ground truth. We report the results in Table 3.3. Notice that we are reporting results on scenarios where there are more speakers than seen during training.

We report the SI-SDRi and median angular error, together with the precision and recall of localizing the voices within 15° of the ground truth when the algorithm has no information about the number of speakers. We remark that as the number of speakers increases, the recall drops as expected. The precision increases are due to the fact that there are fewer false positives when there are many speakers in the scene. The results suggest that our method generalizes and works even in scenarios with more speakers than seen in training.

3.3.5 Results on Real Data and Moving Sources

Dataset: To show results on real world examples, we use the ReSpeaker Mic Array v2.0 Seed (2016), which contains $M = 4$ microphones in a circle of radius 1.27 in (32.2 mm). Although

Table 3.3: Generalization to arbitrary many speakers. We report the separation and localization performance as the number of speakers varies.

Number of Speakers N	2	3	4	5	6	7	8
SI-SDRi (dB)	13.9	13.2	12.2	10.8	9.1	7.2	6.3
Median Angular Error	2.0°	2.3°	2.7°	3.5°	4.4°	5.2°	6.3°
Precision	0.947	0.936	0.897	0.912	0.932	0.936	0.966
Recall	0.979	0.972	0.915	0.898	0.859	0.825	0.785

a network trained purely on synthetic data works well, we find that it is useful to fine-tune with data captured by the microphone. To do this we recorded VCTK samples played over a speaker from known locations, and also recorded a variety of background sounds played over a speaker. We then created mixtures of this real recorded data and jointly re-trained with real and fully synthetic data. Complete details of this capture process are described in the supplementary materials.

Results: In the supplementary videos[2] we explore a variety of real world scenarios. These include multiple people talking concurrently and multiple people talking while moving. For example, we show that we can separate people on different phone calls or 2 speakers walking around a table. To separate moving sources, we stop the algorithm at a coarser window size (23°) and use inputs corresponding to 1.5 seconds of audio. With these parameters, we find that it is possible to handle substantial movement because the angular window size captures each source for the duration of the input. We then concatenate sources that are in adjacent regions from one time step to the next. Because our real captured data does not have precise ground truth positions or perfectly clean source signals, numerical results are not as reliable as the synthetic experiment. However, we have included some numerical results on real data

[2]Available at https://grail.cs.washington.edu/projects/cone-of-silence/

in the supplementary materials.

3.3.6 Limitations

There are several limitations of our method. One limitation is that we must reduce the angular resolution to support moving sources. This in contrast to specific speaker tracking methods that can localize moving sources to a greater precision Traa and Smaragdis (2013); Qian et al. (2017).

Another limitation is that in the minimal two-microphone case, our approach is susceptible to front-back confusion. This is an ambiguity that can be resolved with binaural methods that leverage information like HRTFs Keyrouz (2017); Ma et al. (2017)

A final limitation is that we assume the microphone array is rotationally symmetric. For ad-hoc microphone arrays, our pre-shift method would still allow for separation from a known position. However, the angular window size w would have to be learned dependent on θ, making the binary search approach more difficult.

Chapter 4

CLEARBUDS

In the previous chapter, we showed how The Cone of Silence could isolate a person's voice during a phone call when they were in a noisy environment. One major downside, however, was the requirement of a 4 channel circular microphone array which is not commonly found in phones. In contrast, more people than ever are taking calls on-the-go using wireless earbuds, with 100 million AirPods sold in 2020 (Apple Insider, 2021). While these earbud systems offer unprecedented convenience compared to circular microphone arrays, their mobility raises an important technical challenge: these devices are harder to work with for the end goal of speech isolation. Available products like AirPods only allow sound capture from one earpiece at a time, greatly limiting the spatial information captured. Furthermore, processing the speech in real-time using only the compute of a mobile phone presents a significant challenge. To address this, we build our own earbuds that allow capture from the left and right earbuds across the head, and pair it with a novel speech enhancement network that runs in real-time on a mobile phone.

Source separation of acoustic signals is a long-standing problem where the conventional approach for decades has been to perform beamforming using multiple microphones. Signal processing-based beamformers that are computationally lightweight can encode the spatial information but do not effectively capture acoustic cues (Van Veen and Buckley, 1988a; Krim and Viberg, 1996; Chhetri et al., 2018). Recent work, including that presented in Chapter 4, has shown that deep neural networks can encode both spatial and acoustic information and hence can achieve superior source separation with gains of up to 9 dB over signal processing baselines (Subakan et al., 2021; Luo and Mesgarani, 2019). However, these neural networks are computationally expensive. The existing binaural (i.e., using two microphones) neural

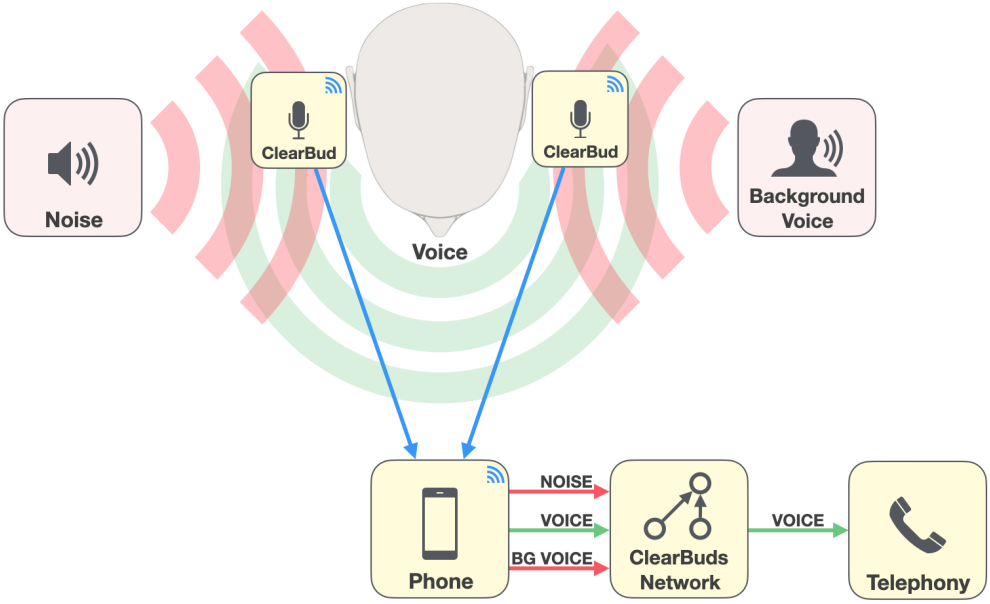


Figure 4.1: ClearBuds System Overview. Our goal is to isolate a user's voice from background noise (e.g., street sounds or other people talking) by performing source separation using a pair of custom designed, synchronized, wireless earbuds.

networks can not meet the end-to-end latency required for telephony applications and have not been evaluated with real earbud data. Commercial end-to-end systems, like Krisp, use neural networks on a cloud server for single-channel speech enhancement, with implications to cost and privacy.

We present the first mobile system that uses neural networks to achieve real-time speech enhancement from binaural wireless earbuds. Our key insight is to treat wireless earbuds as a binaural microphone array, and exploit the specific geometry – two well-separated microphones behind a proximal source – to devise a specialized neural network for high quality speaker separation. In contrast to using multiple microphones on the same earbud to perform beamforming, as is common in Apple AirPods and other hearing aids, we use microphones across the left and right earbuds, increasing the distance between the two microphones and thus the spatial resolution.

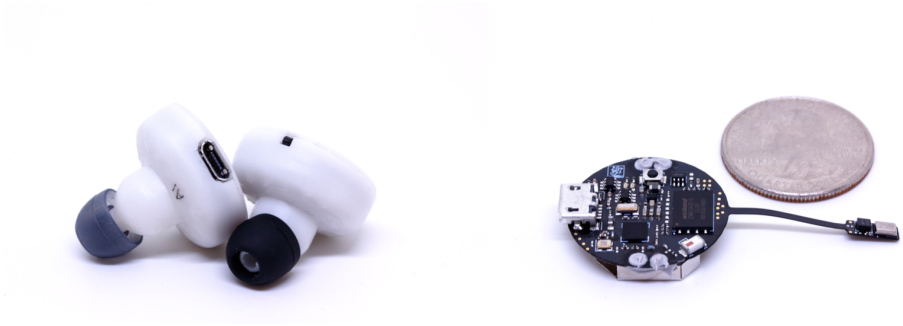


Figure 4.2: ClearBuds hardware inside 3D-printed enclosure and when placed beside a quarter.

To achieve this system, we make three technical contributions spanning earable hardware and neural networks.

- **Synchronized binaural earables.** We designed a binaural wireless earbud system (Fig. 4.2) capable of streaming two time-synchronized microphone audio streams to a mobile device. This is one of the first systems of its kind, and we expect our open-source earbud hardware and firmware to be of wider interest as a research and development platform. Existing earable platforms such as eSense Kawsar et al. (2018) do not support time-synchronized audio transmission from two earbuds to a mobile device. We designed our DIY hardware using open source eCAD software, outsourced fabrication and assembly ($2K for 50 units), and 3D printed the enclosures.

- **Lightweight cascaded neural network.** We introduce a lightweight neural network that utilizes binaural input from wearable earbuds to isolate the target speaker. To achieve real-time operation, we start with the Conv-TasNet source separation network Luo and Mesgarani (2019) and redesign the network to achieve a 90% re-use of the computed network activations from the previous time step for each new audio segment (see Section 4.2.2). While these optimizations make this network real-time, they

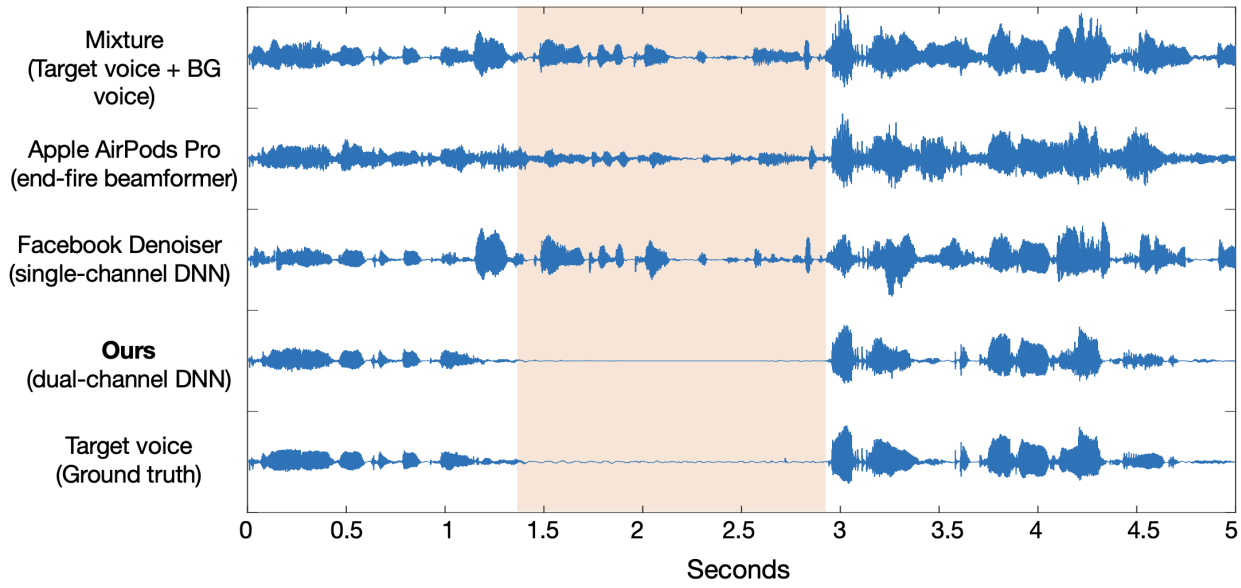


Figure 4.3: Performance with multiple people talking. We use spatial cues to separate background voices from the target speaker, even when the background voice is louder than the target voice. This is evident when the target speaker is silent but background voice continues to talk (highlighted in orange). Apple AirPods Pro uses an endfire beamformer to partially suppress background voice. The mono-channel Facebook Denoiser (Demucs) is unable to suppress the background voice. Clearbud's network removes the background voice, approaching ground truth.

also introduce artifacts in the audio output (i.e., crackling, static). Interestingly, these artifacts have little effect on traditional metrics, like Signal-to-Distortion Ratio (SDR), but have a noticeable effect on subjective listening scores (see 4.4.2). These artifacts however are often visible in a frequency representation of the audio. To address this, we combine our mobile temporal model with a real-time spectrogram-based frequency masking neural network. We show that by combining the two networks and creating a lightweight cascaded network, we can reduce artifacts and improve the audio quality further.

- **Network training for in-the-wild generalization.** Training the network in a supervised way requires clean ground truth speech samples as training targets. This is difficult to obtain in fully natural settings since the ground truth speech is corrupted with background noise and voices. Training a network that generalizes to in-the-wild scenarios also requires the training data to mimic the dynamics of real speech as closely as possible. This includes reverb, voice resonance, and microphone response. Synthetically rendered spatial data is the easiest type of data to obtain, but most different from real recordings, while real speakers wearing the headset in an anechoic chamber provide the best ground-truth training targets, but are the most costly to obtain. Synthetic data can simulate various reverb and multi-path that are not captured in an anechoic chamber. Our training methodology uses large amounts of synthetic data simulated in software, small amounts of hardware data with speakers embedded into a foam mannequin head and small amounts of data from human speakers wearing the earbuds in an anechoic chamber (see 3.3.1) to create a neural network that generalizes to users and multi-path environments not in the training data.

We combine our wireless earbuds and neural network to create ClearBuds, an end-to-end system capable of (1) source separation for the intended speaker in noisy environments, (2) attenuation and/or elimination of both background noises and external human voices, and (3) real-time, on-device processing on a commodity mobile phone paired to the two earbuds. Our results show that:

- Our binaural wireless earbuds can stream audio to a phone with a synchronization error less than 64μs and operate continuously on a coin cell battery for 40 hours.

- Our system outperforms Apple AirPods Pro by 5.23, 8.61, and 6.94 dB for the tasks of separating the target voice from background noise, background voices, and a combination of background noise and voices respectively.

- Our network has a runtime of 21.4ms on iPhone 12, and the entire ClearBuds system

operates in real-time with an end-to-end latency of 109ms. For telephony applications, an ear-to-mouth latency of less than 200ms is required for a good user experience (International Telecommunication Union, 2003).

- In-the-wild evaluation with eight users in various indoor and outdoor scenarios shows that our system generalizes to previously unseen participants and multipath environments, that are not in the training data.

- In a user study with 37 participants who spent over 15.4 hours and rated a total of 1041 in-the-wild audio samples, our cascaded network achieved a higher mean opinion score and noise suppression than both the input speech as well as a lightweight Conv-TasNet.

Qualitative audio examples can be found at `https://clearbuds.cs.washington.edu`.

4.1 Related Work

Building upon the multi-microphone capture techniques discussed in Section 2.1 and the beamforming approaches detailed in Section 2.2, we now focus on the practical implementation of these concepts in wireless earbuds. While recent advances in neural networks have shown promising results, few have been demonstrated within the constraints of wireless earbuds. By creating a wireless network between two earbuds, we demonstrate that our real-time, two-channel neural network can outperform current real-time speech enhancement approaches for wireless earbuds.

4.1.1 Beamforming Techniques

Traditional beamforming techniques, as described in Section 2.2, remain popular in commercial devices such as smart speakers (Amazon, 2018), mobile phones (Samsung, 2014), and earbud devices like Apple AirPods (Apple, 2018) due to their computational efficiency. However, as discussed in Section 2.2, beamforming performance is limited by the geometry of the microphone array and the distance between microphones (Van Veen and Buckley, 1988a;

InvenSense, 2013). The form factor of devices like AirPods restricts both the number of microphones on a single earbud and the available distance between them, limiting the gain of the beamformer—a challenge directly related to the spatial resolution constraints outlined in Section 2.1.

While binaural arrays across two earbuds could provide better performance in principle by increasing the effective array aperture, current wireless architectures are limited to streaming from a single earbud at a time (Telephony and Group, 2020). Furthermore, adaptive beamformers such as MVDR are sensitive to sensor placement tolerance and steering vector accuracy (Zhang and Wang, 2017; Brandstein and Ward, 2001). Traditional beamforming also leverages only spatial or spectral cues and does not use acoustic cues (e.g., structure in human speech) and perceptual differences to discriminate sources—information that machine learning methods can successfully leverage.

4.1.2 Single-channel Deep Speech Enhancement

Many deep learning techniques operate on spectrograms to separate the human voice from background noise (Xu et al., 2015; Mohammadiha et al., 2013a; Duan et al., 2012; Nikzad et al., 2020; Choi et al., 2019; Weninger et al., 2015; Fu et al., 2019). However, recent works instead operate directly on the time domain signals (Luo and Mesgarani, 2019; Germain et al., 2018; Pascual et al., 2017; Defossez et al., 2020; Macartney and Weyde, 2018), yielding performance improvements over spectrogram approaches. Commercial noise suppression software like Krisp and Google Meet have successfully deployed single-channel models in real-time and are available for use on mobile phones and desktop computers, but processing is performed on the cloud. Fedorov et al. (2020) achieves low-power speech enhancement using long short-term memory (LSTM), but it is for a single-channel network, not for multichannel source separation. Further, single-channel models cannot effectively capture spatial information and fail to isolate the intended speaker when there are multiple speakers (see Fig. 4.3), highlighting the importance of the multi-microphone approaches discussed in Section 2.1.

4.1.3 Multi-channel Source Separation and Speech Enhancement

Multi-channel methods have been shown to perform better than their single-channel source separation counterparts (Yoshioka et al., 2018; Chen et al., 2018; Zhang and Wang, 2017; Gu et al., 2020; Tzirakis et al., 2021; Jenrungrot et al., 2020). Binaural methods, which leverage the interaural time and level differences, have also been used for source separation (Sun et al., 2019; Han et al., 2020b; Li et al., 2011; Reindl et al., 2010) and localization (van Hoesel et al., 2008; Lyon, 1983; Kock, 1950). Han et al. (2020b) reduces the look-ahead time in the network to make it causal in behavior but has not been demonstrated to run on a mobile device.

Our method improves on existing binaural methods by combining time-domain neural network with spectrogram-based frequency masking networks and optimizing them to enable real-time processing on a phone. Recent works such as Tan et al. (2019, 2021); Shankar et al. (2020) use multiple microphones on a smartphone for speech enhancement. However, neither of them demonstrates evaluation with real data, where artifacts because of network optimizations can affect user performance.

In contrast, we demonstrate the first system that achieves real-time speech enhancement using microphones on two wireless earbuds. Further, as the distance between the earbuds is larger than the distance between microphones on a typical mobile phone, we can attain a better spatial resolution than a mobile phone implementation, while also retaining the ability to speak hands-free. More recent works tackle the problem of real-time directional hearing using eye trackers and wearable headsets. For example, Wang et al. (AAAI 2022) uses a hybrid network that combines signal processing with neural networks, but shows that their technique performs poorly in binaural scenarios (i.e., two microphones) and requires four or more microphones. In contrast, we focus on the problem of speech enhancement and create the first real-time end-to-end hardware-software neural-network based system using wireless synchronized earbuds.

4.1.4 Earbud Computing and Platforms

There has been recent interest in earbud computing (Ma et al., 2021b; Kawsar et al., 2018; Min et al., 2018; Powar and Beresford, 2019; Yang and Choudhury, 2021b) to address applications in health monitoring (Chan et al., 2019; Bui et al., 2021; Chan et al., 2022), activity tracking (Ma et al., 2021a) and sensor fusion with EEG signals (Ceolini et al., 2020). The eSense platform (Kawsar et al., 2018; Min et al., 2018) has enabled research in sensing applications with earables. OpenMHA (Pavlovic et al., 2018; Herzke et al., 2017) is an open signal processing *software* platform for hearing aid research. Neither of these platforms support time-synchronized audio transmission from two earbuds, which is a critical requirement for achieving speech enhancement in binaural settings. In contrast, we created open-source wireless earbud hardware that can support synchronize wireless transmission from the two earbuds.

4.2 ClearBuds Design

We first introduce our lightweight neural network architecture. We then describe system design of our hardware platform and our synchronization algorithm.

4.2.1 Problem Formulation

Suppose we have a 2 channel microphone array with one microphone on each ear of the wearer. The target voice is speaking with a signal $s_0 \in \mathbb{R}^{2\times T}$ in the presence of some background noise $\mathbf{bg}$ or other non-target speakers $s_{1..N}$. There may also be multi-path reflections and reverberations $\mathbf{r}$ which we would also like to reduce, i.e., $\mathbf{x} = \sum_{i=0}^{N} \mathbf{s}_i + \mathbf{bg} + \mathbf{r}$. Our goal is then to recover the target speaker's signal, s_0, while ignoring the background, reverbations, or other speakers. We also must do so in a real-time way, meaning that the a mixture sample $\mathbf{x}_t$ received at time t must be processed and outputted by the network before $t + \mathbf{L}$ for some defined latency $\mathbf{L}$. We refer to the non-target speakers as "background voices". These background voices may be at any location in the scene, including very close

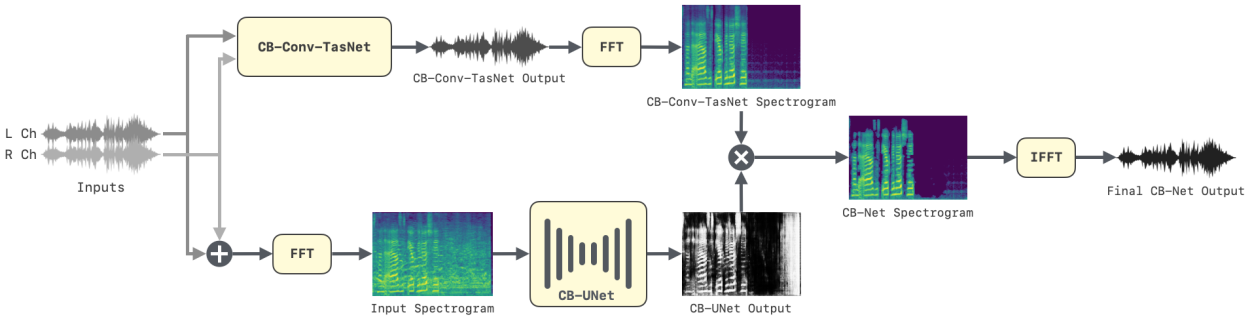


Figure 4.4: Network Diagram of CB-Net. Our network contains a time-domain component, shown in the top as CB-Conv-TasNet, and a frequency domain component, shown on the bottom by CB-UNet.

to the target speaker and their angle can change with time and motion.

4.2.2 Neural Network Architecture Motivation

Our network needs to perform in real-time on a mobile device with minimal latency. This is challenging for several reasons. First, the processing device has a much lower compute capacity, especially compared to cloud GPUs. Additionally, the network should separate non-speech noises as well as unwanted speech. To do this, it must learn spatial cues and human voice characteristics. Finally, the resulting output should maximize the quality from a human experience perspective while minimizing any artifacts the network might introduce.

Our network, which we call *ClearBuds-Net* or *CB-Net*, is a cascaded model that operates in both time and frequency domains. The full network architecture is illustrated in Fig. 4.4 and contains two main sub-components: A dual-channel time domain network called *CB-Conv-TasNet*, and a frequency based network called *CB-UNet*.

CB-Conv-TasNet

The first component of separation method is a time domain network that is based on a multi-channel extension of Conv-TasNet (Luo and Mesgarani, 2019). This is a network in

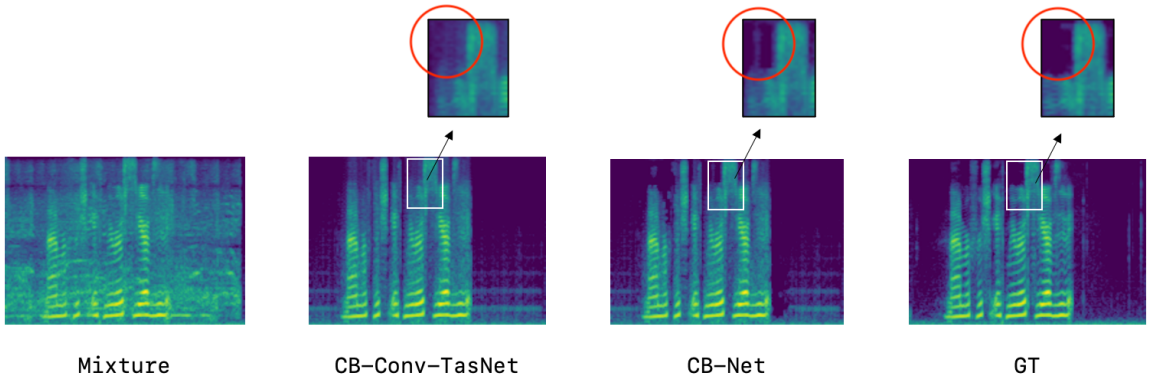


Figure 4.5: The spectrograms above show the motivation behind a combined time and frequency domain method. The output of the time-domain component, CB-Conv-TasNet, contains artifacts, particularly at high frequencies. Although subtle, these artifacts are perceptible by human listeners. CB-Net is able to reduce these artifacts by using a frequency-domain network (CB-UNet) that masks unwanted frequencies.

the waveform domain that has a Temporal Convolution Network (TCN) structure, lending itself to a causal implementation with intermediate layer caching (Paine et al., 2016). We use depthwise separable convolutions (Howard et al., 2017) to further reduce the number of parameters and make the design real-time. We call this network CB-Conv-TasNet since it is an optimized version of the original Conv-TasNet.

A key feature of the time domain approach is that it can easily capture spatial cues in the network. In our application, the desired source is always physically between two microphones, thus the voice signal will reach the microphones roughly at the same time. In contrast, background or other speakers are typically not temporally aligned and will reach one microphone earlier or later. By feeding two time synchronized channels into the neural network, this spatial alignment of the sources can be learned from time differences in the signal. This is similar to a delay-and-sum beamforming effect, except the sum is replaced with a deep network.

CB-UNet

The output of our lightweight CB-Conv-TasNet often contains audible artifacts (i.e., crackling, static) that reduce the listening experience. Interestingly, these artifacts have little effect on traditional metrics, like Signal-to-Distortion Ratio (SDR), but have a noticeable effect on subjective listening scores (see 4.4.2). These artifacts are often visible in a frequency representation of the audio. Fig. 4.5 shows how CB-Conv-TasNet alone contains noticeable artifacts when compared to the ground truth. To address this, we cascade a lightweight causal UNet (Ronneberger et al., 2015a) which operates on the mel-scale spectrogram of the input audio. This network, which we call CB-UNet, produces a binary mask which is applied to the output of CB-Conv-TasNet. The combined output, shown in Fig. 4.5 as CB-Net, reduces these artifacts. The mean opinion scores in our evaluation shows the strength of the cascaded CB-Net when compared to the time-domain component only.

4.2.3 Neural Network Detailed Description

CB-Conv-TasNet

The input to the network is a binaural mixture given by $\mathbf{x} \in \mathbb{R}^{2\times T}$. The first step is an encoder that transforms the mixture $\mathbf{x}$ into $\mathbb{R}^{N\times T/L}$ with a 1D convolution of size L and stride L. This is followed by a ReLU layer. The encoder's outputs are next fed into a temporal convolution network that consists of stacks of 1-D convolutions with increasing dilation factors. We use 14 convolution layers with dilation factors of 1,2,4,..,64 repeated twice, with a ReLU nonlinearity and skip connection after each convolution. The encoder output is multiplied with the output of the temporal conv-net, before being fed through a fully connected Decoder layer which transforms the output back into $\mathbb{R}^{2\times W}$.

In a real-world implementation, we do not have access to the full waveform, but only packets of data at a time. Furthermore, we must process these packets with limited access to future input samples. Given 15.625 kHz sampling rate, we choose to process packets of 350 samples at a time (22.4ms), which is our window size W. We also use $2W$, or 700

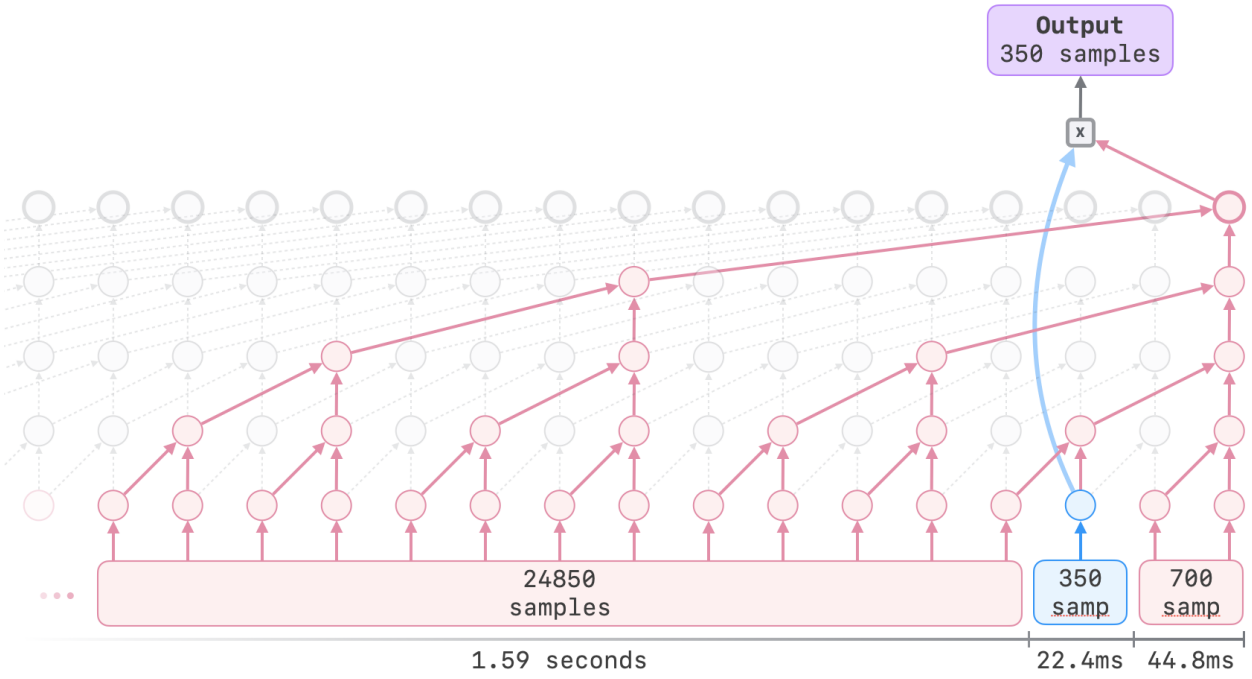


Figure 4.6: CB-Conv-TasNet, the time-domain component of CB-Net. Given a packet of 350 samples (22.4ms) highlighted in blue, we use 1.5s of past input and 44.8ms of future input to output the separation results. Our caching scheme works as follows: When we receive a new 350ms samples, all intermediate activations (circles in the diagram) slide to the left, and we compute only the rightmost column of outputs.

samples of lookahead time (44.8ms) and 1.5s of past samples. Since we have no padding in the temporal convolution net, the network starts with this large temporal context and outputs exactly $\mathbb{R}^{1\times W}$ samples, corresponding to the desired output for our input packet of W samples. When we receive the next packet of size W, all intermediate activation from the encoder and temporal conv-net can be shifted over by W/L samples and re-used. We chose $L = 50$, but any divisor of W would work. Re-using intermediate outputs from previous packets saves over 90% of the compute time for a new packet in our network.

CB-UNet

The frequency domain network is a mono-channel network that outputs a binary mask for each time-frequency bin. The input $\mathbf{x} \in \mathbb{R}^{1\times T}$ is a summation of the binaural left and right channel, which is the equivalent of a broadside beamformer. We first run a STFT, which is a mel-scale fourier transform with hop size of 350, a window size of 1024 including zero padding on the edges, and a 128 bin mel-scale output. The network input is a spectrogram of 64 time bins and 128 frequency bins, corresponding to a receptive field of 22400 samples, or $1.43s$. In order to maintain the causality requirement, we use the same lookahead strategy as the time-domain network where we allow 700 samples of lookahead for a target packet of 350 samples. The UNet architecture contains 4 downsampling and upsampling layers, starting with 64 channels and doubling the number of channels at each subsequent layer. The downsampling layers contain a depthwise separable convolution followed by a 2×2 max pooling, and the upsampling layers contain a depthwise separable convolution followed by a transposed convolution for upsampling. The output is a sigmoid function, which is then thresholded to return a binary mask in $[0, 1]^{128\times 64}$. When outputting a spectrogram mask on an $\mathbb{R}^{128\times 64}$ input, we predict a mask over the entire input even though we only need the output for a specific slice of 350 samples, or a $\mathbb{R}^{128\times 1}$ mask. Further optimizations could be made by caching intermediate outputs or only computing the mask for the target samples. However CB-UNet's run-time was so small compared to the rest of the network that these optimizations were not necessary.

Combining the Outputs

At each time step, the output of CB-Conv-Tasnet is an audio waveform in $x \in \mathbb{R}^{1\times 350}$, and the output of CB-UNet is a spectrogram mask in $\mathbf{M} \in \mathbb{R}^{128\times 64}$. We run the same fourier transform on the buffered conv-tasnet outputs to produce a spectrogram $\mathbf{X} \in \mathbb{R}^{128\times 64}$. Our output can then be computed by $iSTFT(\mathbf{M} \otimes \mathbf{X})$. Our empirical results show that this gives the best results compared to other methods such as ratio masking.

Training

CB-Conv-TasNet is trained with an $L1$-based loss over the waveform along with the multi-resolution spectrogram loss. Formally, provided s_0 is our target speaker and x' is the output from the network, our loss is:

$$L(s_0, x') = \|s_0 - x'\|_1 + L_{sc}(s_0, x') + L_{mag}(s_0, x')$$
$$L_{sc}(s_0, x') = \frac{\|STFT(s_0) - STFT(x')\|_F}{\|STFT(s_0)\|_F}$$
$$L_{mag}(s_0, x') = \|\log(STFT(s_0) - \log(STFT(x'))\|_1$$

STFT denotes the magnitude of the short time Fourier transform, and F denotes the Frobenius norm. L_{sc} and L_{mag} represent spectral convergence and magnitude losses, which gave better results than L1 loss alone.

For training CB-UNet, for each time frequency bin, the training target $\mathbf{M}$ is 1 if the target voice is the dominant component, and 0 otherwise. Formally, $\mathbf{M}(f,t) = [\mathbf{S}_0(f,t) \geq \mathbf{S}_i(f,t)], \quad \forall i = (1..n)$. The network is then trained with the binary cross entropy of the output compared to the target mask.

Hyperparameters and Training Details

We use a learning rate of 3×10^{-4} along with the ADAM optimizer (Kingma and Ba, 2014) for training the network. The network was trained on a single Nvidia TITAN Xp GPU. Because of the small size of the network, training could be completed within a single day and generally required ≈50 epochs to reach convergence.As an additional data augmentation step we make the following perturbations to the data: High-shelf and low-shelf gain of up to $2dB$ are randomly added using the `sox` library.

4.3 Training methodology

Training the network in a supervised way requires clean ground truth speech samples as training targets. This is difficult to obtain in fully natural settings since the ground truth

speech is corrupted with background noise and voices. Training a network that generalizes to in-the-wild scenarios also requires the training data to mimic the dynamics of real speech as closely as possible. This includes reverb, voice resonance, and microphone response. Synthetically rendered spatial data is the easiest type of data to obtain, but most different from real recordings, while real speakers wearing the headset in an anechoic chamber provide the best ground-truth training targets, but are the most costly to obtain. Synthetic data can simulate various reverb and multipath that are not captured in an anechoic chamber. We adopt a hybrid training methodology where we first train on a large amount of synthetic data and fine-tune on real data recorded with our hardware. Our training method is based on the commonly used mix-and-separate framework Zhao et al. (2018), where clean speech and noise samples are recorded separately and combined randomly to form noisy mixtures. Our results show that our network trained this way generalizes to naturally recorded noisy data in real-world environments.

4.3.1 Synthetic Data

This type of data is the easiest to obtain, since a wide variety of voice types and physical setups can be generated instantly. Many machine learning baselines, e.g., Luo et al. (2020b); Jenrungrot et al. (2020); Tzirakis et al. (2021), only train and evaluate on synthetic data generated in this manner. To generate the synthetic dataset, we create multi-speaker recordings in simulated environments with reverb and background noises. All voices come from the VCTK dataset Veaux et al. (2016a) (110 unique speakers with over 44 hours), and background sounds come from the WHAM! dataset (Wichern et al., 2019), with 58 hours of recordings from a variety of noise environments such as a restaurant, crowd, and music.

To synthesize a single example, we create a 3 second mixture as follows: two virtual microphones are placed 17.5 cm apart, which is the average distance between human ears (Risoud et al., 2018). The target speaker's voice is placed at the center between the two virtual microphones, and a second voice is placed randomly between 1 and 5 meters away and at a random angle. A randomly chosen background noise is also placed in the scene.

We then simulate room impulse responses (RIRs) for a randomly sized room using the image source method implemented in the pyroomacoustics library (Allen and Berkley, 1979; Scheibler et al., 2018). The room is rectangular with sides randomly chosen between 5 and 20 meters, and the RT60 values are randomly chosen between 0 and 1 second. All signals are convolved with the RIR and rendered to the two channel microphone array. The volumes of the background are randomly chosen so that the input signal-to-distortion ratio is roughly between -5 and 5 dB. For training, we use 10,000 mixtures generated in this manner.

4.3.2 Hardware Data

While a large amount of synthetic data can be easily rendered to train the network, it does not contain characteristics such as the microphone response of physical hardware and imperfections in the time-of-arrival. To address this, we also train on a set of recorded voice samples from our earbuds. We set up a foam mannequin head with an artificial mouth speaker (Sony SBS-XB12) that plays VCTK samples as the spoken ground truth. For background voice recordings, the speaker is placed in varying locations within a one meter radius of the foam head. Physically recorded background noise is provided by binaural version of the WHAM! dataset (Wichern et al., 2019), which was recorded in real environments using a binaural mannequin like ours. We record 2 hours each of clean speech, and background voices. 2000 random mixtures are then created for training.

4.3.3 Human Data

The spoken hardware data above still does not contain natural voice resonance since it is played out of an electronic speaker. Furthermore, the background sounds recorded by a mannequin wearing earbuds still misses some of the physical filtering of the human body. To better capture desired output of real scenarios, we collect a ground-truth speech dataset in an anechoic chamber with human speakers (5 male, 4 female) and a noise dataset in real environments with human listeners. For the voice data, each human speaker wore our ClearBuds prototypes, and uttered 15 minutes of text from Project Gutenberg in the

Figure 4.7: In-the-wild experiments in various scenarios (crowded cafe, busy intersection, outdoor plaza, classroom) were conducted across 8 users and indoor and oudoor environments, all unseen in our training dataset.

anechoic chamber. The purpose of this anechoic data is to provide clean training targets for the network, modelling the resonance of human speakers wearing our hardware. For the real world noise dataset, individuals wore ClearBuds and recorded various noisy scenarios such as washing dishes, loud indoor/outdoor restaurants, and busy traffic intersections. 2000 random mixtures of clean voice and recorded noise were generated for this dataset.

Our network is jointly trained using all these datasets. Note that testing and evaluation is done *outside* the anechoic chamber.

4.4 Experiments and Results

We first compare our end-to-end system performance against a commercial wireless earbud system. We then present an in-the-wild evaluation of our system. Next, we compare numerical results against various speech enhancement baselines. Finally, we present system-level evaluations. Our work is approved by the IRB.

4.4.1 Comparison with Beamforming Earbuds

We evaluate our end-to-end system against the Apple AirPods Pro headset connected to a iPhone 12 Pro in a repeatable physical set up. In our evaluation, as is typical, there is no overlap between training and test datasets.

Procedure. We use the popular metric *scale-invariant signal-to-distortion ratio* (SI-SDR) (Roux et al., 2018). While SI-SDR provides a repeatable metric used in the acoustic community, it requires a clean, sample-aligned ground truth (target voice) as the basis for evaluation. Therefore, we create a repeatable soundscape for our test setup where a sample-aligned ground truth can be obtained. A foam mannequin head with a speaker (Sony SBS-XB12) inserted into its artificial mouth uttered one hundred VCTK samples with identities and samples unseen in the training set. The mannequin wore ClearBuds and AirPods Pro in subsequent experiments, and the outputs of the two systems could be directly compared. Ambient environmental sound (from WHAM! dataset) was played via four monitors (PreSonus Eris E3.5) positioned to fill 3 meter by 4 meter room, and background voice (also VCTK) was played from a monitor positioned 0.4 meters from head on the right.

All speakers were driven through a common USB interface (PreSonus 1810c) ensuring the same time-alignment and loudness between the two test conditions. Since Apple AirPods Pro beamforming cannot be toggled on and off, we cannot calculate an SI-SDR increase (SI-SDRi), and therefore report output SI-SDR. To establish the ground truth voice against which to calculate SI-SDR, we record clean target voice through each headset. Ambient noise SNR ranged between 0dB and 16dB with respect to target voice. Qualitatively, this sounded like a second person speaking loudly in a noisy bar or cafe. Finally, background voice SNR ranged between 6dB and 12dB, qualitatively sounding like a person speaking from a meter or two away.

Results. We report output SI-SDR from the two systems in Fig. 4.8. To calculate output SI-SDR, we align individual one second chunks and take the logarithmic mean across 250 chunks. We find that ClearBuds achieves higher output SI-SDR across all test conditions when compared to the beamforming utilized by the Apple AirPods Pro. For a qualitative comparison of AirPods Pro versus ClearBuds performance with human speakers, see video: https://clearbuds.cs.washington.edu/videos/airpods_comparison.mp4.

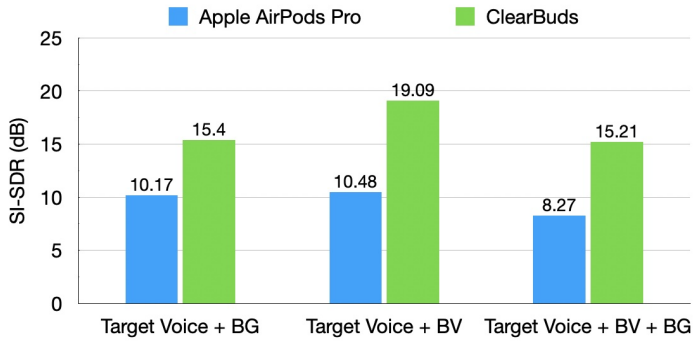


Figure 4.8: Comparison with AirPods Pro. Reporting the output SI-SDR (note: not SI-SDR increase). ClearBuds exceeds in three conditions: target voice plus background noise (BG), target voice plus background voice (BV), target voice plus background voice and noise.

4.4.2 In-the-Wild Evaluation

We perform in-the-wild evaluation in indoor and outdoor scenarios with users not in the training data. The procedure and results are described in the following sections.

In-the-wild experiments. Eight individuals (four male, four female, mean age 25) with a variety of accents wore a pair of ClearBuds and read excerpts from Project Gutenberg while in four noisy environments: a coffee shop, a noisy intersection, an outdoor plaza, and a classroom (see Fig. 4.7). The environments featured ringing phones, cross-talk from other people, ambient music, a crying baby, opening/closing doors, driving vehicles, and street noise, amongst other common sounds. These experiments were uncontrolled in that the background voices and noise were naturally occurring sounds that are typical to these real-world scenarios and were mobile.

Evaluation procedure. In-the-wild evaluation precludes access to clean, sample-aligned truth to compute SI-SDR. Instead, the common (and expensive) procedure is to perform a user study and compute the mean opinion score. Since this is a time-consuming process, prior works on binaural networks, e.g., Luo et al. (2020b); Tan et al. (2019); Jenrungrot

et al. (2020), avoid in-the-wild evaluation. Since our goal is to design and evaluate an in-ear system in real scenarios, we recruit thirty-seven participants (11 female, 26 male, mean age 29) for a user study. Each participant listened to between 6 and 11 in-the-wild audio samples (avg. 9.38 samples, each between 10–60 seconds). Each speech sample was processed and presented three ways: (1) the original input, (2) CB-Conv-TasNet, and (3) CB-Net, yielding a total of $37 \times 9.38 \times 3 = 1{,}041$ rating samples.

Participants were encouraged to use audio equipment they would typically use for a call. Fourteen used earbuds, thirteen used computer speakers, seven used headphones, and three used phone speakers. The study took about 25 minutes per participant. As is typical with noise suppression systems, participants were asked to give ratings in two categories: the intrusiveness of the noise and overall quality (mean opinion score - MOS):

1. **Noise suppression:** How INTRUSIVE/NOTICEABLE were the BACKGROUND sounds? 1 - Very intrusive, 2 - Somewhat intrusive, 3 - Noticeable, but not intrusive, 4 - Slightly noticeable, 5 - Not noticeable

2. **Overall MOS:** If this were a phone call with another person, How was your OVERALL experience? 1 - Bad, 2 - Poor, 3 - Fair, 4 - Good, 5 - Excellent

Results: Fig. 4.9 shows the noise intrusiveness and MOS values for the original microphone, CB-Conv-TasNet, and CB-Net. As expected, applying CB-Conv-TasNet to the original audio helped suppress noise dramatically, increasing opinion score from 2.02 (slightly better than *2 - Somewhat intrusive*) to 3.28 (between *3 - Noticeable, but not intrusive* and *4 - Slightly noticeable*) ($p < 0.01$). The light-touch, spectrogram-masking clean up method featured in CB-Net increased noise suppression opinion score significantly ($p < 0.001$) to 3.77, indicating the method did indeed further suppress perceptually annoying noise artifacts. Importantly, this step also increased overall MOS. While users only slightly preferred ($p < 0.05$) CB-Conv-TasNet (2.67) to the original input (2.49) due to artifacts introduced, they more significantly ($p < 0.001$) preferred our CB-Net (3.10), an increase of 0.61 opinion score points from the

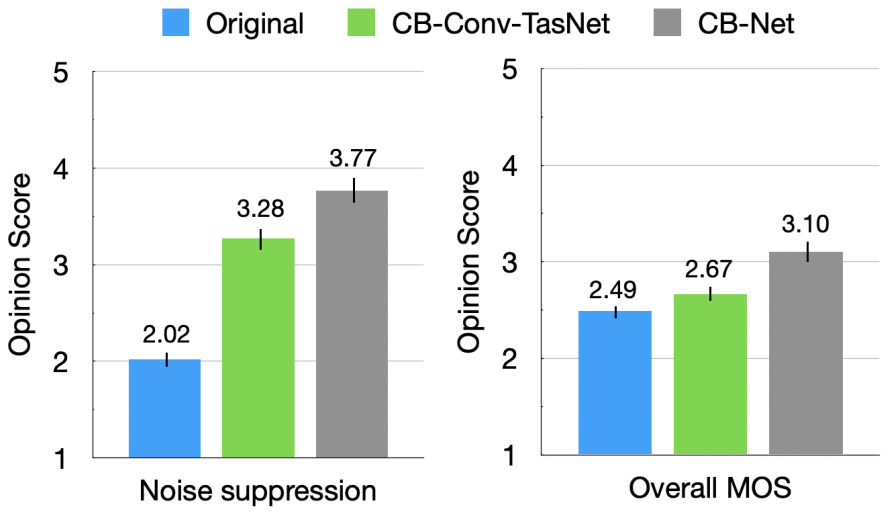


Figure 4.9: In-the-wild study results. Noise suppression indicates perceived quality of background noise reduction (higher is less intrusive). Overall MOS indicates overall perceived quality. Error bars are 95% CI.

input. For context, in the flagship ICASSP 2021 Deep Suppression Noise Challenge (Reddy et al., 2021), with state-of-the-art, real-time algorithms run on a quad-core desktop CPU, the winning submission increased MOS by 0.57 (Reddy et al., 2021) from input.

Note that in our in-the-wild experiments, the background noise and voices were not static. The speakers themselves can also be mobile. Our network was able to adaptively remove the background noise and achieve speech enhancement with mobility.

4.4.3 Benchmarking our Neural Network

The conventional evaluation in the machine learning and acoustic community is to evaluate models and techniques on synthetic data against baselines. For completeness, we compare our method against a variety of speech enhancement baselines using the synthetic dataset. For evaluation, an additional 1000 mixtures of 3 seconds each were generated such that there was no overlapping identities or samples between the train and test splits.

Evaluation Procedure: For comparisons to other baseline methods, we use the popular SI-SDR and PESQ metrics. Unlike the AirPods experiment, where the original noisy mixture could not be recorded since AirPods beamforming cannot be toggled off, here we compute SI-SDR of the ground truth relative to both the input noisy mixture and then to the network output. When reporting the increase from the input SI-SDR to output SI-SDR, we use the SI-SDR improvement (SI-SDRi).

For a deep learning baseline in the waveform domain, we choose the causal Demucs model (Defossez et al., 2020). This is a single channel method which was recently shown to outperform many other deep learning baselines and runs real-time on a laptop CPU. We also compare with Dual-signal Transformation LSTM Network (DTLN) (Westhausen and Meyer, 2020). This method also runs on a laptop or mobile phone in real-time. To compare with spectrogram based methods, we use the oracle baselines, ideal ratio mask (IRM) and ideal binary mask (IBM) (Stöter et al., 2018; Wang, 2005), that use the ground truth voice to calculate the best possible result that can be obtained by masking a noisy spectrogram.

As an ablation study, we report results with each individual component of the network, *CB-Conv-TasNet* and *CB-UNet*. We also show results when the multi-channel part of our network, *CB-Conv-TasNet*, only has access to one microphone, labeled as *CB-Conv-TasNet Single Mic*. This explicitly shows the advantage of using two microphones. There are only a few deep learning methods that tackle binaural speech separation for mobile processing, and the most relevant ones, such as Tan et al. (2019) and Han et al. (2020b), do not have publicly available code to test against.

Results: As shown in Table 4.1, our binaural method is comparable to the best possible results that can be obtained by a spectrogram masking method (IBM, IRM). We also show an improvement over waveform based deep learning methods that only use a single microphone input. In particular, the improvement is greatest when there are two speakers present (Target Voice + Background Voice). This is because single channel methods can only rely on voice characteristics, whereas our network also uses spatial cues to separate the speaker

Table 4.1: Benchmarking our neural network. We show results for a target voice speaking in three noise scenarios: (1) Background noise (BG), (2) Background voice (BV), and (3) Background noise and background voice (BG and BV). CB-Conv-TasNet performs slightly better on synthetic data, but as shown in Fig. 4.9, does not generalize as well to in-the-wild scenarios. This demonstrates the importance of evaluating networks on real in-the-wild hardware data.

	SI-SDR increase (SI-SDRi)			Output PESQ		
Method	Target with BG	Target with BV	Target with BV + BG	Target with BG	Target with BV	Target with BV + BG
CB-Net	10.41	10.56	9.35	2.08	2.68	1.81
CB-Conv-TasNet	11.19	11.01	9.68	2.24	2.58	1.91
CB-Conv-TasNet Single Mic	6.15	0.13	2.34	1.82	1.84	1.53
CB-UNet	3.21	0.78	1.82	1.60	2.10	1.50
DTLN Westhausen and Meyer	7.02	0.06	2.13	2.08	1.95	1.67
Causal Demucs Defossez et al.	6.62	-0.03	2.11	1.80	1.88	1.43
Ideal Ratio Mask (IRM, oracle)	11.41	11.53	12.04	2.53	3.00	2.44
Ideal Binary Mask (IBM, oracle)	9.97	11.05	10.85	2.30	2.90	2.21

of interest. Although *CB-Net* shows similar or worse performance to *CB-Conv-TasNet*, subjective evaluation on in-the-wild hardware data shows that *CB-Net* is far superior to human listeners (see 4.4.2).

Examples of the synthetic dataset, outputs from all the methods and qualitative comparisons against Krisp, a commercial noise suppression system, can be found linked from our project website: `https://clearbuds.cs.washington.edu`.

We further report the runtime of our network variations in Table 4.2

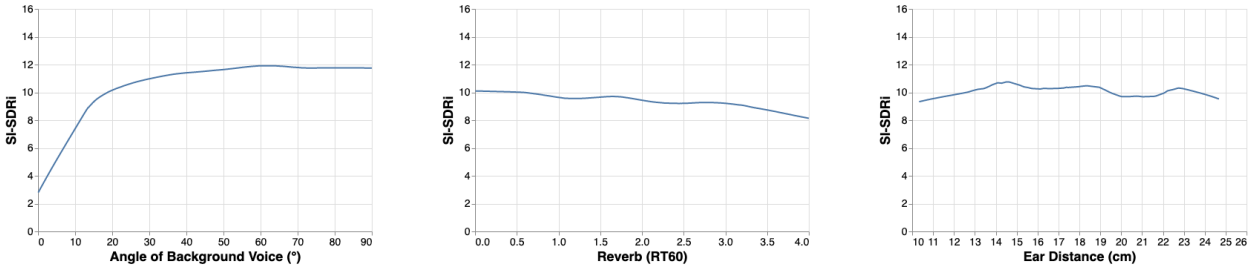
4.4.4 Additional Network Ablations

We numerically evaluate various aspects of the design by changing the angle of background voice, reverberance in the environment, and microphone separation.

Table 4.2: Neural network run time on smartphones

Device	Conv-TasNet	CB-Conv-TasNet	**CB-Net**
iPhone 12 Pro	155.5ms	17.5ms	21.4ms
iPhone 11	165.4ms	18.6ms	22.7ms
iPhone XS	241.5ms	27.2ms	33.0ms
FLOPs/packet	1078M	97M	131M

Angle of Background Voice: The ability of our network to separate the target voice from a background voice is based on utilizing the time difference of arrival to the binaural microphones. Because we only have two microphones, this ability is limited when the background voice is in the front-back plane of the speaker. In this case, the background voice will arrive at each microphone simultaneously, and there will be no spatial cues to separate the two voices. To illustrate this effect, we graph the separation performance as a function of the angle of the background voice in Fig. 4.10a.



(a) (b) (c)

Figure 4.10: (a) Performance against angle of background voice in presence of significant multipath. (b) Performance against amount of reverberation in an indoor room. RT60 (in seconds) measures how long sound takes to decay by 60 dB in a space with a diffuse soundfield. (c) Performance as distance between ears increases.

Multipath and Reverberant Environments: While our in-the-wild experiments show the performance in various indoor and outdoor environments, we benchmark our system in different reverberant conditions, including those more reverberant than seen during training. Synthetically generated mixtures are generated using the pyroomacoustics library with the RT60 value randomly chosen between 0 and 4s. We generate 200 examples and plot the SI-SDRi compared to the RT60 in Fig. 4.10b. Our method shows only a slight decrease in performance as the reverberation of the environment increases. Because the target speaker is physically close to the microphone array, our setup is generally less affected by reverberations than other kinds of source separation problems where the target speaker may be further away.

Separation Between Microphones: Our in-the-wild evaluation across 8 participants showed generalization across facial features. Here, we benchmark our method to different head sizes where the distance between the microphones may be different. We generate 200 synthetic samples, where the distance between the microphones is randomly chosen between 10 and 25 cm. Because the target speaker is in the middle of the microphone array, the target signal will arrive at both mics simultaneously regardless of the microphone distance. Fig. 4.10c show little change in performance even with microphone distances greatly different than used during training.

4.5 Limitations & Future work

The first limitation is that the user must be wearing both wireless earbuds to benefit from our binaural noise suppression network. Second, with only two microphones, there is an opportunity for background voices to remain in the uplink channel if the voice is within a few degrees of the target speaker's sagittal plane (see Fig. 4.10a). The underlying assumption of our network is that the mouth is in the middle of the user's ears, though as seen in Fig. 4.10c and our in-the-wild evaluation, some variance is permissible.

While we minimize the power consumption of the ClearBud hardware, we shift the processing and therefore power consumption to the more powerful mobile phone. Performing

network computation on the mobile phone over a cloud GPU is an enhancement in terms of user privacy and security so that sensitive voice data is not transmitted to the cloud. While mobile chips are becoming more power efficient, an alternative design to explore is to run our neural network on a plugged-in edge device (e.g., router), minimizing computation while achieving similar latency.

Future work could integrate two microphones in each earbud, so that each earbud could beamform toward the user's mouth prior to processing in the neural network. We also had to develop a custom wireless audio protocol to stream two microphones to a single phone. While this prevents this architecture from being deployed on today's commodity wireless earbuds, adoption may be imminent as Bluetooth 5.2 shows promise with the introduction of Multi-Stream Audio and Audio Broadcast FAQs.

Our network could also be deployed on other multi-microphone mobile or resource-constrained edge systems such as smart watches, augmented reality glasses, or smart speakers to allow for enhanced voice control or telephony in noisy environments. The hardware and firmware for Clearbuds could be leverage to produce wireless, synchronized microphone arrays for telephony, acoustic activity recognition or for swarm robot localization and control.

Chapter 5

HRTF ESTIMATION IN THE WILD

Up until now, we have explored multi-microphone array processing mainly for the applications of speech isolation and localization. However, microphone arrays can also be valuable in creating realistic spatial audio listening experiences. Spatial audio is an important aspect of many audio applications, including virtual and augmented reality, gaming, music, and audio for film and television (Begault and Trejo, 2000; Wenzel et al., 1993). The fundamental challenge of spatial audio is to create the perception that sound is coming from any location in space, even though the sound is played back through headphones. Humans are remarkably good at perceiving the location of incoming sounds in the real world, with as little as 3.5° error even in noisy environments (Makous and Middlebrooks, 1990). This ability is achieved through the Head Related Transfer Function (HRTF), which is the direction-dependent filtering of sound by the head, ears, and torso. By using a listener's HRTF to render virtual sounds, it is possible to create an immersive audio experience that simulates sound coming from any position in 3D space. The HRTF is comprised of two components: interaural time differences (ITD) and interaural level differences (ILD). While both components are important for accurate spatial localization, this work focuses on the frequency dependent ILDs, also called spectral features which describe the different frequencies arriving at each ear. These are more easily obtainable from in-the-wild recordings and have been shown to be more important for HRTF personalization compared to ITDs (Wenzel et al., 1993).

A key problem is that HRTFs vary significantly from person to person, and using a personalized HRTF is necessary to create high fidelity spatial audio. This is because using someone else's HRTF or a generic HRTF will lead to localization errors and an unpleasant listening experience (Wenzel et al., 1993; Middlebrooks, 1999a). Despite its importance,

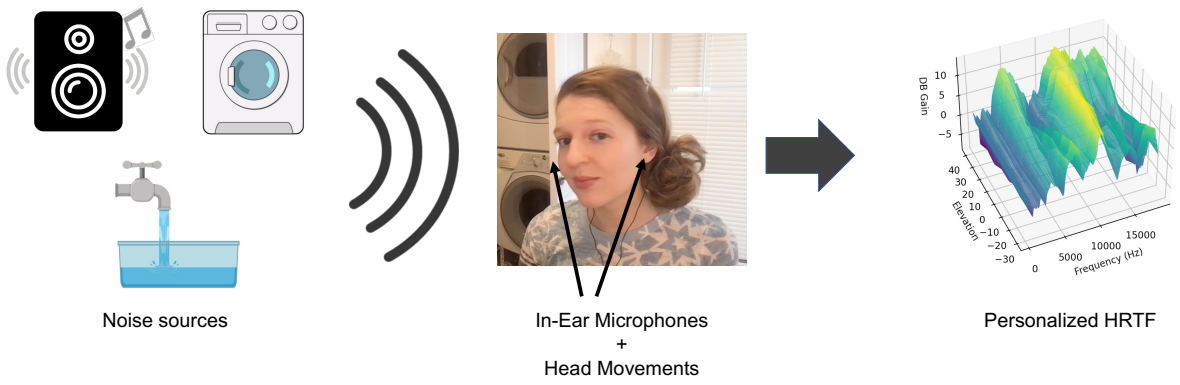


Figure 5.1: Our method uses binaural recordings of everyday noises along with head tracking information to create a personalized HRTF for the listener.

accurately measuring an individual HRTF is a difficult task. This is due to the fact that HRTF is a complex, dynamic phenomenon that is affected by a variety of factors, including an individual's ear shape, head size, and general anatomy. Traditional methods require the listener to sit in an anechoic chamber while sine-sweeps are played from all possible angles. Other methods involve taking complex 3D scans of the head and ears along with anatomical measurements. This problem of personalized spatial audio has also received increasing attention from companies, such as Apple, Sony, and Logitech, which have recently developed methods to create personalized HRTFs through head scans, imaging, and user feedback (Apple, 2023c; Sony, 2023; Logitech, 2023). Despite these advances, achieving high-fidelity spatial audio remains an ongoing challenge and an active area of research and development.

We are particularly motivated by the rapid proliferation of earbuds systems, with 100 million AirPods sold in 2020 alone (Apple Insider, 2021). These systems typically contain a microphone in each ear as well as a head tracking IMU, making them ideal for capturing personalized HRTFs. Although AirPods only capture data from one earpiece at a time, the ClearBuds hardware in Chapter 4 allows time-synchronized data capture from both

ears, making them perfect for measuring HRTFs. As more and more people use earbuds, we envision a future where collecting data for personalized HRTFs is as simple as wearing earbuds and moving around in different environments. By analyzing the changes in sound arriving at the listener's ears over time, we can infer their personalized HRTF and use it across a wide range of spatial audio applications. This approach has the potential to be more efficient and less burdensome than traditional methods that require 3D scans or anthropometric measurements.

As a step towards that, in this chapter we present a method for measuring individualized HRTFs that leverages environmental sounds recorded by the listener in everyday settings. Our approach is designed for scenarios where there is a single stationary noise source, and we demonstrate its effectiveness using a wide range of noise sources such as music, home appliances, and outdoor sounds. By analyzing the recorded sounds, we can extract features that are specific to the listener's HRTF and use them to construct a personalized HRTF. Our method utilizes machine learning along with synthetic and real training data in order to predict the frequency-dependent filtering of a subject from natural recordings.

To validate our approach, we conducted user studies with real listeners and show three key experimental results. First, our predicted HRTFs closely match the ground truth HRTF recorded in an anechoic chamber. Second, our HRTFs significantly improve the sound localization accuracy of users in a virtual auditory display when compared to a generic HRTF. Third, our HRTFs greatly reduce front-back confusion when used to render sounds. Overall, our proposed method of measuring individualized HRTFs in-the-wild has the potential to offer a more efficient and less burdensome alternative to traditional methods, and we hope that it will inspire further research and development in this field. Video demos and code can be found on our project website.[1]

[1] https://grail.cs.washington.edu/projects/hrtf-in-the-wild/

Figure 5.2: Example from Southampton (2003) of an anechoic chamber and speaker array for measuring HRTFs

5.1 Related Works

Building upon the HRTF foundations discussed in Section 2.3, this chapter focuses on developing methods to personalize HRTFs in practical, accessible ways. As explained in Section 2.3, accurate HRTFs are essential for creating spatially realistic audio experiences, but traditional measurement methods present significant challenges.

Traditional methods of measuring an individual's HRTF involve dense acoustic measurement in an anechoic chamber (Sridhar et al., 2017; Algazi et al., 2001; Møller et al., 1995b; Watanabe et al., 2014; Southampton, 2003), as described in Section 2.3. While these approaches provide high fidelity HRTFs, they remain time-consuming and resource-intensive, requiring the listener to visit a specialized lab for measurement. This limitation was a primary motivation for the development of alternative approaches outlined in Section 2.3.

To simplify this process, methods have been proposed that only use a single loudspeaker Reijniers et al. (2017, 2020); Li and Peissig (2017), aligning with the practical motivations discussed in Section 2.3. In these works, a reference signal is played from a stationary loudspeaker while the subject rotates their head through different directions under the measurement of an IMU or other head tracking device. This approach greatly simplifies the HRTF measurement process, and our method builds on this idea of measuring the listener's motion relative to the noise source.

Acoustic simulations on 3D scans of individuals represent another category of HRTF estimation methods, extending the computational approaches discussed in Section 2.3. The algorithms described in Ziegelwanger et al. (2015); Huttunen et al. (2014, 2007); Meshram et al. (2014c,a) use 3D mesh data with boundary-element methods to simulate the diffraction of sound waves through the head and ear, similar to the Mesh2HRTF approach referenced in Figure 2.9 of Section 2.3. It has also been shown that HRTFs can be calculated directly from a point cloud of the head Sridhar and Choueiri (2017). Although not published, the method released by Apple Apple (2023c) uses the depth sensor to create a 3D scan of the head. Despite their advantages, these methods still suffer from several drawbacks, including the need for an accurate 3D mesh and the privacy concerns associated with 3D scanning and imaging.

Following the anthropometric and geometric modeling approaches described in Section 2.3, it is possible to estimate the HRTF directly from anthropometric measurements, given the availability of large HRTF datasets with associated head and ear measurements Algazi et al. (2001); Watanabe et al. (2014). The works in Zotkin et al. (2002, 2003) show positive results when selecting the HRTF with the closest anthropometric measurements to a new user, leveraging the CIPIC database approach illustrated in Figure 2.9 of Section 2.3. Other works Hu et al. (2006); Zhao et al. (2022); Chun et al. (2017); Chen et al. (2019) use regression methods or deep learning to predict HRTF features from these anthropometric measurements, including works like Zhi et al. (2022); Zhao et al. (2022); Mohan et al. (2003) that use images of the ears along with anthropometric measurements. These approaches extend the machine learning and deep learning approaches outlined in Section 2.3, but still face challenges in obtaining accurate measurements.

Building on the modern methods discussed in Section 2.3, recent approaches have been proposed to measure HRTFs acoustically in less controlled environments. The method in Diepold et al. (2010) proposed measuring the HRTF from everyday recordings, but uses a third microphone in the room as a way to record the clean reference signal. Another method Zandi et al. (2022) allows the user to play sine sweeps from their smartphone, but

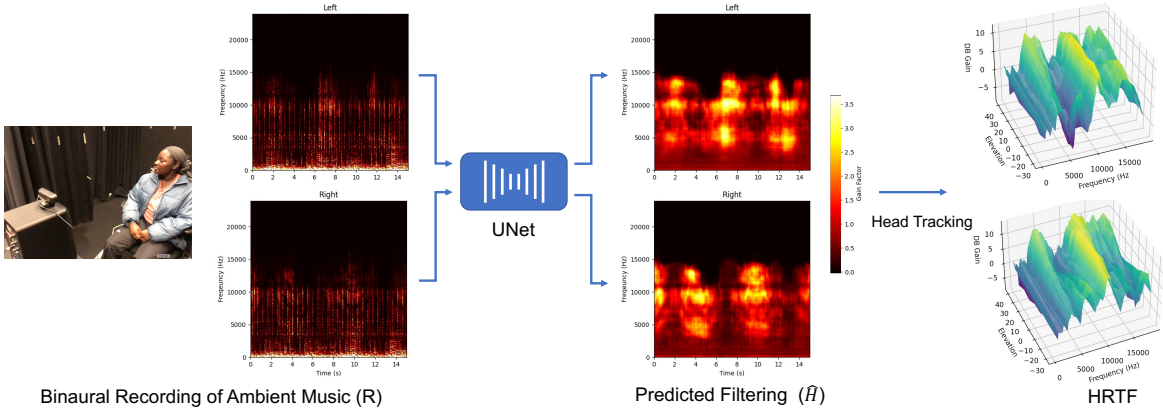


Figure 5.3: An overview of our method. We use binaural recordings of in-the-wild sounds to predict the filtering from the HRTF at each time step. We then use the head tracking data to map this predicted filtering to the user's location dependent HRTF.

requires capturing this signal from at least 60 unique locations around the head. Similarly, the method in Yang and Choudhury (2021a) asks the user to play predefined sounds from their smartphone while they move the phone around their head. Finally, the method in Yamamoto and Igarashi (2017) allows users to answer pairwise comparison questions in a listening study to determine the best HRTF.

In contrast to these methods, our approach offers several key advantages. First, we don't require an anechoic chamber or specialized speakers. Second, we don't require any 3D scans or imaging of the head. Finally, the recordings are collected passively from sound sources in the environment. In our method, the user is in an everyday environment and we capture their natural head movements after an initial calibration step. This aligns with the need for more accessible personalization approaches highlighted in Section 2.3, and addresses the limitations of traditional measurement methods while maintaining perceptual accuracy. Our approach is designed to be less cumbersome than existing methods that involve answering questions, moving a smartphone around, or taking detailed head measurements and scans.

5.2 Method

Suppose that a listener is wearing earbuds which contain a microphone in each ear as well as a head tracking IMU. The listener may be in the presence of a sound source s, and the microphones at each ear will pick up the binaural recording given by r_l and r_r for left and right respectively. We also use the 3DoF head rotation: $\boldsymbol{\theta}_h(t)$ which describes the head rotation at any given moment in time t This data is available from recent airpod devices (Apple, 2023b), or could be captured through the webcam. Our goal is to learn the HRTF solely from r_l, r_r, and $\boldsymbol{\theta}_h(t)$. We use the uppercase notation S, R_l, and R_r to refer to the time-frequency representation of the audio, and the lowercase to refer to the waveform representation. Similarly we use H to denote the filtering function imposed by the listener during a recording, and the time domain version, the head related impulse response (HRIR) is written as h. In this work, we limit the method to scenarios with a single stationary sound source, and its position relative to the head is written as $\theta_s(t)$, and $\varphi_s(t)$.

Under the simplest assumptions, the captured audio is a convolution between the HRIR and the original source. For example, the recorded audio r can be written as

$$r = h * s \tag{5.1}$$

In the frequency domain, this is

$$R = H \cdot S \tag{5.2}$$

or equivalently

$$H = R/S \tag{5.3}$$

Furthermore, the recording may include multi-path signals and other ambient noise, denoted as ϵ, which add ambiguity to the scenario. Breaking this down by left and right separately we get

$$H_l = (R_l - \epsilon_l)/S \tag{5.4}$$

$$H_r = (R_r - \epsilon_r)/S \tag{5.5}$$

As we can see, this is a highly underdetermined problem, since we do not have access to the original sound source S or multipath contributions ϵ, only the captured recordings R_l and R_r. Therefore, at any given moment, we would not know whether a given frequency was modified by the listener's HRTF or by the emitting sound source.

Our goal is then to predict H_l and H_r from R_l and R_r without access to the actual ground truth source S. We can then use H_l and H_r along with the head tracking information to create a listener specific HRTF, which is a function of the source location and frequency: $HRTF(\theta_s, \varphi_s, f)$

5.2.1 Deep Network

To solve this problem, we can use the fact that most sound sources have a repeated or predictable frequency distribution over time which can be learned. Furthermore, the recording from both ears together provide clues towards the relative filtering at each ear. We frame this problem as a supervised learning problem and use a deep network for this prediction task. Deep networks can learn the underlying structure of sounds such as speech and various noise sources, solving one of the ambiguities. Secondly, these networks can use the temporal information of the source along with the data captured at both ears to predict which frequencies are being modified by the HRTF instead of by the sound source or multipaths.

Our network is a modification of the Unet Convolutional Neural Network (Ronneberger et al., 2015b) with an initial convolutional block comprising 32 features, and composed of of 4 downsampling and 4 upsampling convolutional blocks. R_l and R_r are produced using the magnitude of a short-time Fourier transform of the captured audio. They are concatenated channel wise, and feed through the network to produce 2 channels of output of the same dimension. The output represents the predicted level change at a certain frequency due to the HRTF as a scalar factor. We found that learning the filtering function as multiplicative gain was easier than dB due to the fact that cutting out a frequency would require learning a dB gain of $-\infty$. Training details are described in Section 5.3.

5.2.2 Source Localization and Head Tracking

Head tracking through an IMU or camera can provide the 3DoF rotation angle of the head. However, because the sound source may not be located directly in front of the user, it is also necessary to know the location of the signal relative to the user. In our system, we require an initial localization input from the user. They are asked to point their head directly towards the sound source (or directly away for sounds coming from behind). They then press a button which allows the system to record the initial location of the sound source, $\theta_s(0)$ and $\varphi_s(0)$. During the rest of the recording process, the rotation matrix of the head orientation can be applied to the initial source location to give the relative position of the sound source at that time, $\theta_s(t)$ and $\varphi_s(t)$.

It may also be possible to infer the initial source location using localization algorithms, but we leave that as future work as the manual localization by the user is quick and very accurate.

5.2.3 HRTF Estimation from Aggregated Results

By aggregating predictions across many recordings with different sound sources and head rotations, we can obtain a more accurate and full representation of the listener's HRTF. Let F be the number of frequency bins in the spectrogram representation. For each binaural recording $R \in \mathbb{R}^{2\times T\times F}$, we use a deep network to predict the filtering function $\hat{H} \in \mathbb{R}^{2\times T\times F} = \text{UNet}(R)$.

To build a model of the listener's HRTF, we first initialize an empty HRTF for all source locations and frequencies. Then, we use the UNet to predict how the listener's HRTF filtered the sound source for each recording. If the entirety of a recording contains minimal energy at a given frequency, we assume that this frequency was absent from the source signal and do not use it. Finally, we use the known relative location of the source over time to create a HRTF prediction for each location-frequency bin. Our method does not explicitly solve for directions with no data, but in such scenarios, we could use HRTF extrapolation/interpolation methods

which have shown good results when we only have a sparse HRTF (ben hur et al., 2020; Ito et al., 2022)

Across time steps, the predicted filtering function for the same location-frequency bin may vary due to the changes in the underlying sound source, reverb, or other effects not modeled such as doppler effects. Because of this, we average the predicted HRTF values at each location-frequency bin to obtain the listener's HRTF magnitude at each location and frequency. One of the advantages of our method is that over time, we can collect more and more information about the HRTF and use that to produce a better estimate. We explored both the mean and median and found that the mean worked better.

To obtain the head-related impulse response (HRIR) for use in spatial audio applications, we also need the phase information which describes the interaural time differences (ITDs). We use ITDs from a generic HRTF and apply inverse fast Fourier transform (IFFT) to obtain the HRIR. Although some previous works in similar domains (Richard et al., 2022; Steinmetz et al., 2021) predict the phase as well as the magnitude of the impulse response, we found that phase was much harder to predict in a reverberant environment due to multipath effects. At many frequencies, the captured phase was completely different from the actual ITD phase due to multipath interference. Our user studies also showed that generic ITDs still produced a strong ability to localize sounds. The full algorithm is described in Algorithm 2.

5.2.4 System Implementation

Our method is general and designed to work with any device that supports binaural recordings and head tracking. This could include earbuds, VR headsets, or smart glasses. However, with the exception of certain headsets paired with certain phones (Schoon, 2023; Apple, 2023a), these devices do not currently expose the required functionality to third-party developers. We therefore built our own physical system with commercially available hardware.

For the binaural recordings, we used the Sound Professionals SP-TFB-2 in-ear Binaural Microphones (Professionals, 2022). These wired headphones are capable of capturing frequencies up to 20kHz. It's noteworthy that our microphones, unlike those used in numerous

Algorithm 2 Create HRTF, HRIRs from Binaural Recordings

for $\forall\theta, \forall\varphi, \forall f$ **do**
    $H\hat{R}TF(\theta, \phi, f) \leftarrow []$ ▷ Initialize empty HRTF
end for
for $R \in$ `Recordings` **do**
    $\hat{H} \leftarrow \text{UNet}(R)$ ▷ Network inference
    for $t \in 0..T,\ f \in 0..F$ **do**
        if $R(f)$`.mean()` $> \epsilon$ **then**
            $H\hat{R}TF(\theta_s(t), \phi_s(t), f).\text{append}(\hat{H}(t, f))$
        end if
    end for
end for
for $\forall\theta, \forall\varphi, \forall f$ **do** ▷ Use phase from generic HRTF
    $|H\hat{R}TF(\theta, \phi, f)| \leftarrow H\hat{R}TF(\theta, \phi, f)$`.mean()`
    $\angle H\hat{R}TF(\theta, \phi, f) \leftarrow \angle HRTF_{\texttt{generic}}(\theta, \phi, f)$
end for
$HRIR(\theta, \varphi) =$ `iFFT`$(HRTF(\theta, \varphi))$

previous studies such as Sridhar et al. (2017); Sridhar and Choueiri (2017); Algazi et al. (2001) are positioned at the entrance of the ear canal rather than fully blocking it. Our research demonstrates that it is feasible to generate an accurate HRTF even without perfect microphone placement. Extending this methodology to commercial earbuds would require re-training with data captured using those specific headphones to learn their unique transfer function.

For head tracking, we used the face pose detector provided by the Google MediaPipe Library (Google, 2023). This algorithm uses a forward facing webcam to detect the 3DoF head position, and is based on BlazeFace (Bazarevsky et al., 2019) and AttentionMesh (Grishchenko et al., 2020). The head tracking runs in less than 10`ms` on a Macbook pro, and

we use a HRTF with bin size $\theta = 5°$ and $\varphi = 5°$. This means that as long as the user is not rotating their head faster than $\sim 300°/s$, the head tracking will assign the sound to the correct HRTF bin.

Figure 5.4: Left: Our head tracking implementation uses the webcam to determine the 3DoF head rotation during recording. The normal vector is drawn in blue to help visualize the direction the head is pointing. Right: An image of the binaural microphone used in our implementation. The microphone sits near the ear canal.

5.3 Data and Training

To train our network, we adopt an approach that combines synthetic and real data. We begin with large amounts of synthetically rendered data, which enables us to learn from a wide range of noise types and simulated environments, including multi-path scenarios. However, such data does not capture all nuances of real-world audio and fails to generalize completely to actual recordings. To address this, we incorporate real data, which is more time-consuming to collect but provides more effective training for the network. By leveraging both sources of data, we are able to benefit from the strengths of each approach. This mix of synthetic and real training data has been explored in previous works as well (Chatterjee

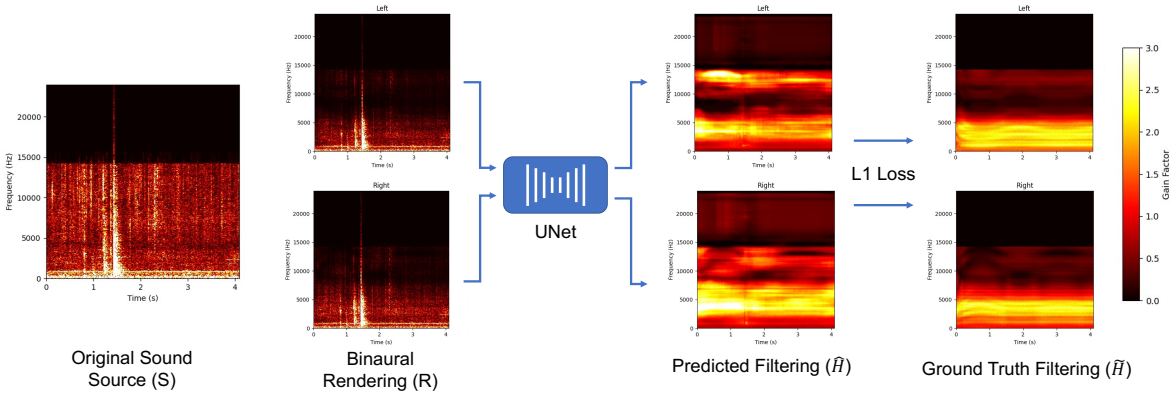


Figure 5.5: The process for training the network. We use create binaural renderings of a sound source with simulated multi-path environments. We then use the ground truth filtering of the HRTF to train the network with an L1 loss between the predicted filtering, $\hat{H}$ an the ground truth filtering $\tilde{H}$. The real training data is used in an identical way except R is a binaural recording, not a binaural rendering, and we don't have access to the original sound source S.

et al., 2022; Jenrungrot et al., 2020; Seib et al., 2020). Below, we describe the two data sources in more detail.

5.3.1 Synthetic Training Data

We first train the network on synthetically rendered spatial data. For HRTFs, we use the RIEC dataset Majdak et al. (2013), which contains 109 HRTFs measured in an anechoic chamber for different individuals. This was split into a training set of 64 and a test set of 45 HRTFs. To render sounds, we use the Steam Audio C++ API which allows realistic sound rendering for moving sources with custom HRTFs and multi-path environments.

The noise sources come from the WHAM! dataset (Wichern et al., 2019) and AudioSet dataset (Gemmeke et al., 2017). These datasets contain a wide variety of noise sources such as music, speech, appliances, and machinery. Sound sources without sufficient frequency

ranges (requiring a minimal energy up to at least 5khz) and without sufficient regularity (e.g. impact only sounds) were filtered out. Some example of sound categories that were removed included chewing, clapping, snapping, and whistling. Some of the most effective noise sources included water, kitchen appliances, pop music, and machinery. For both datasets, a 80/20 train/test split was maintained. None of the audio samples used for training the network were used during any of the synthetic or real evaluations

Each generated recording was 3s long and created by placing the sound source at a random azimuth and random elevation 1.5m away from the listener while the listener moved their head in a random direction at a random speed. Multipaths were simulated by create walls at distances between 2 and 10 meters from the listener with RT60 values from 0.4 to 0.9 seconds. For each recording we also obtained the ground truth filtering at the source locations to use as a training label, $\tilde{H}$. The train set contains 10,000 generated examples, and the test set contains 1,000 examples.

5.3.2 Real Training Data and Anechoic HRTFs

Although large amounts of synthetic data can be easily collected, a network solely trained on synthetic data does not perform well in real-world scenarios. To address this issue, we augment the training data with in-the-wild recordings that more closely resemble the acoustic environments and sounds that would be encountered by users during inference. The main challenge is that the ground-truth filtering function is required as a training supervised label for the model. In order to generate these labels, we measure the ground-truth HRTFs of the subjects in an anechoic chamber. These HRTFs can also be used as a baseline to evaluate the in-the-wild inferred HRTFs.

Our anechoic HRTF measurement procedure is most similar to the method described in Reijniers et al. (2020). Subjects are seated in an anechoic chamber while a single loudspeaker emits a reference signal. They are instructed to move their head slowly to cover a broad range of azimuth and elevation angles. Unlike Reijniers et al. (2020), we place the speaker at 3 different elevation angles when capturing the ground-truth HRTF. This better captures

the filtering effects of the torso at different sound elevations which are not captured by simply rotating the head up and down with respect to a single speaker location. Furthermore, we use a broadband Gaussian noise signal instead of a sine sweep, as we only care about the frequency-dependent level differences and not the ITDs. This allows us to capture the filtering across all frequencies at each time step. The speaker used is the KRK Classic 5 Studio Monitor which contains 2 drivers. To account for an imperfect speaker response, a reference signal $\tilde{S}$ is first recorded. The ground truth filtering function is then obtained by dividing the recording R by $\tilde{S}$. This also has the effect of cancelling out any frequency response imposed by the microphones as both R and $\tilde{S}$ contain the same microphone response. A full discussion of speaker and microphone compensation is provided in Langendijk and Bronkhorst (2000).

After collecting the anechoic HRTFs, we generated real training data for the neural network by having 2 subjects listen to 1 hour of noise sources, from the training partition of our audio datasets, played back through the loudspeaker in regular environments. The speaker location was known, and the training label $\tilde{H}$ could be generated from the anechoic HRTFs.

5.3.3 Training Details

All recordings were captured at 48kHz sample rate. Each training example contained 3s of binaural audio, and mini-batch size 32 was used. The STFT was conducted with a window size of 2048. Training occured on a Nvidia Titan Xp GPU and took approximately 10 hours for 100 epochs of training. Data augmentation techniques included random left-right flip, random volume changes, and the addition of random noise. Samples from the real and synthetic dataset were randomly sampled with equal probability

5.4 Results

We evaluate the effectiveness of our method through a user study, and present three key results to show the strength of the method. First, we show that our predicted HRTFs closely

match the ground-truth HRTFs. Second, we demonstrate that our HRTFs improve localization by listeners in a virtual environment. Finally, we show that our HRTFs significantly reduce front-back confusion with rendered sounds.

5.4.1 User Study and In-the-Wild HRTF

8 individuals with regular hearing abilities (4 male, 4 female, mean age 28) participated in the user study. First, we measured their ground-truth HRTF in an anechoic chamber as described in Section 5.3.2. Next, we used our in-the-wild method to measure their HRTF in a regular environment. The subjects were in a normal sized reverberant room, that was not particularly quiet. The background noise in the room was measured to be around $50\mathtt{dB}$ due to electric hum and other noises. Next, a variety of noise types were played from a loud speaker in the room. This included music, running water, kitchen appliances, and other sounds from the test partition of the WHAM! and AudioSet datasets. The speaker was placed at 3 elevations and a variety of azimuth angles relative to the listener at distances that varied from $1 - 3\mathtt{m}$. The listener was instructed to rotate their head through a normal range of angles as they listened to the audio sounds. In total, roughly 15 minutes of audio were captured per user across all the locations.

5.4.2 Comparison with Ground-Truth HRTF

The first metric we use to evaluate the correctness of our HRTFs is the agreement with the ground-truth HRTF. A visual comparison between the two is shown in Figure 5.6 which plots the results for a given subject at four consecutive elevations and $\theta = 0°$. To evaluate the similarity quantitatively, at every azimuth and elevation, we compute the Log-Spectral Distortion (LSD) in dB which is given by

$$LSD(\hat{H}, \tilde{H}) = \sqrt{\frac{1}{F}\sum_{f=1}^{F}\left(20\log_{10}\left|\frac{\tilde{H}(k)}{\hat{H}(k}\right|\right)^2} \tag{5.6}$$

We then report the median value across all azimuth and elevations in table 1. Our method

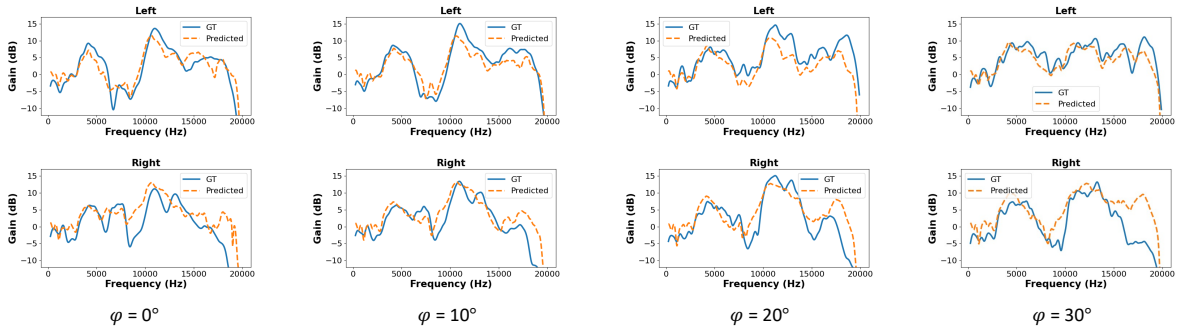


Figure 5.6: We plot the ground-truth HRTF and predicted HRTF for a given test subject for $\theta = 0°$ and 4 elevations. The HRTF that we create for the user closely matches the ground truth, even though the magnitude of some notches and peaks may not be exactly correct.

is compared with several other methods as well. For the random method, we average the LSD when comparing the ground-truth HRTF with all other HRTFs in the RIEC database. For a generic HRTF, we used the KEMAR HRTF (Gardner et al., 1994) which contains measurements for a dummy head commonly used as a generic HRTF model. We also share the results reported in Hu et al. (2006) and Zandi et al. (2022). It's important to note that the method in Zandi et al. (2022) achieves it's reported results when the HRTF is measured at 1138 unique locations done actively by the user, and the method in Hu et al. (2006) requires detailed anthropometric measurements of the ear, head, and torso.

5.4.3 Localization in a Virtual Auditory Display

To evaluate the effectiveness of our HRTFs in spatial audio applications, we created a virtual auditory display where sounds could be rendered spatially with dynamic head tracking. Our method aimed to replicate the experimental method described in ben hur et al. (2020). A sound reproduction system was implemented in Unity and Steam Audio where a virtual sound was placed at an arbitrary location and played back to the listener through headphones. The listener could then move their head, with the sound location adjusting accordingly based

Method	LSD (dB)
Random RIEC Subject	8.23
Generic HRTF	7.32
Zandi et. al Zandi et al. (2022)	4.5
Ours	**4.38**
Hu et. al Hu et al. (2006)	3.5

Table 5.1: Log-spectral distortion between ground-truth HRTF and the output HRTF for several methods. We note that the method in Hu et al. (2006) requires additional physical measurements and the method in Zandi et al. (2022) requires significantly more active input from the user.

on head tracking information sent to Unity via a UDP connection. Overall the system's latency from the head movement to the sound update was less than 30ms which is below the perceptual lag for binaural listening (Yairi et al., 2008; Sankaran et al., 2016). Listeners were placed in a room with a grid of angular markings on 3 sides of them at 10° intervals for both azimuth and elevation. A white noise stimulus was played for a maximum of 5 seconds during which the listeners could make exploratory head movements within a maximum of 30° of the forward facing angle. They were then asked to indicate their perceived source location by pointing at the best grid location. Experiments were conducted for sources in the front hemisphere and back hemisphere separately with sources coming from random locations in $\theta \in [-70°, 70°]$ and $\varphi \in [-30°, 40°]$. A brief calibration period was used where the listener could see the ground truth location for the first 4 examples while making the exploratory head movements. Each subject then evaluated 20 random locations for each candidate HRTF.

Results are reported in Figure 5.7. For both total angular error and elevation error, listeners performed significantly better with our method ($p < 0.01$) compared to a generic

HRTF. In addition, the localization error with our method was close to that of the anechoic ground-truth HRTF. We note that, although the mean azimuth error was better with our method and the ground-truth HRTF compared to a generic HRTF, it was not statistically significant ($p > 0.05$). We hypothesize that this is because ITDs are the primary method used by humans for azimuth inference, and both the ground-truth HRTF and our method contained generic ITDs with only personalized spectral features. Statistical significance was computed with an independent-samples t test between the two candidate distributions.

5.4.4 Front Back Confusion

The last experiment conducted was a front-back confusion test using rendered sounds. A short white noise stimulus was rendered at a random location using a candidate HRTF and played back to the listener through headphones. The listener then had to predict whether the source was coming from the front or back hemisphere. The locations used were $\theta \in [-70°, 70°]$ in the front and back, and $\varphi \in [-30°, 40°]$. Like the previous experiment, the listener received the ground truth answer for the first 4 locations. However, unlike the previous experiment, the listener was not allowed to make exploratory head movements and had to predict front or back based on the rendering alone. Each subject then evaluated 30 random locations per HRTF before moving on to the next HRTF. The results are shown in table 2 which once again show a significant improvement ($p < 0.01$) when using our method compared to a generic HRTF.

5.5 Limitations and Conclusion

Our method shows a strong ability to solve for a listener's HRTF using only binaural recordings of in-the-wild sounds and relative head tracking information. However, there are several limitations that need to be acknowledged.

First, our method was only demonstrated with a single stationary noise source. Such scenarios are limited in everyday settings, and solving for the HRTF with multiple sources or moving sources would present additional challenges. It would be necessary to localize moving

Method	Front-back confusion rate
Generic	29.0% ± 5.4
Ours	**14.8% ± 4.6**
GT HRTF	9.6% ± 4.2

Table 5.2: Front-back confusion with rendered sounds. We report the percent of times the listeners made an error, along with the first standard deviation

sources and separate the contributions to the recording from multiple sources. Second, the user still has to actively localize the sources at the beginning of each recording, which presents an additional burden compared to a fully passive HRTF estimation method. This could be resolved by using a binaural localization method, and erroneous localizations could be compensated through outlier detection methods. Finally, the microphones in commercial earbuds are often not exactly at the ear canal entrance. The effect of the earbud on the HRTF would need to be taken into account through careful measurements of the earbud system. Despite these limitations, our method for HRTF estimation has immense potential as wireless earbuds proliferate among everyday users. We show strong performance on a variety of real-world user studies, and we hope that our method can be incorporated into commercial earbud systems in the near future.

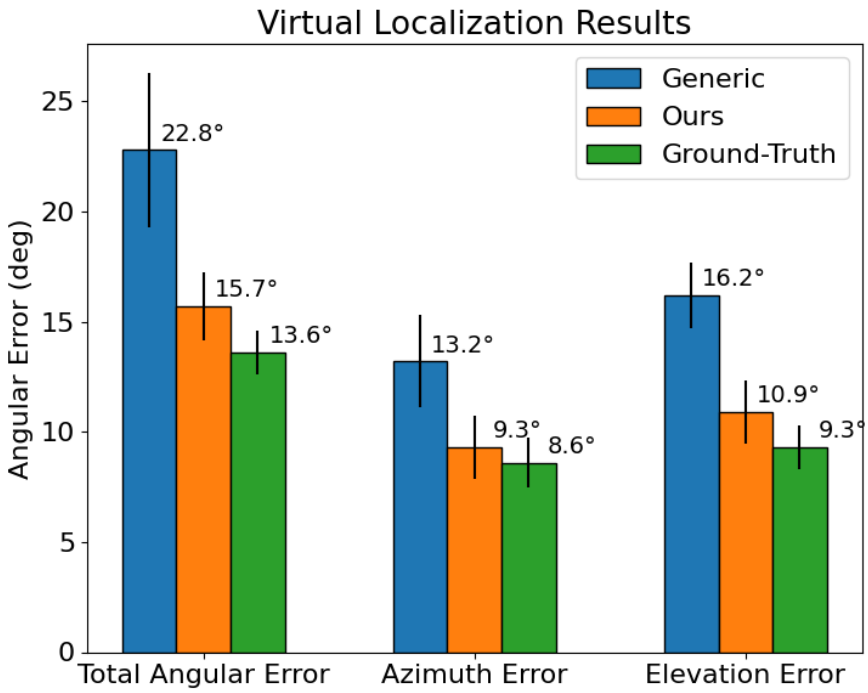


Figure 5.7: Localization results for the virtual auditory display experiment. Results are reported for 3 different experiments: a generic HRTF, the HRTF predicted using our method, and the ground-truth anechoic HRTF described in Section 5.3.2. For each experiment, we first show the total angle difference between the source and prediction. We then show the prediction error broken down by azimuth and elevation error. Results are averaged over all subjects and trials. Error bars shown are the first standard error of the mean.

Chapter 6

GENERATIVE MODELS

The second half of this thesis focuses on using generative models to solve signal restoration problems. Unlike microphone array-based methods, generative approaches can eliminate the need for specialized capture devices, offering a versatile solution that can extend across audio, images, and other modalities.

Generative models leverage a large number of examples to learn the characteristics of a desired data distribution. We can then use these generative models to facilitate signal restoration by sampling from them in a way that aligns with the observation of a noisy or incomplete signal. This chapter serves as an introduction to generative models within this specific context of signal restoration. We focus on introducing the classes of generative models that will be used in subsequent chapters.

Section 6.1 begins by exploring the foundational principles of generative models and their relationship to data distributions. We then provide an overview of key generative modeling paradigms: flow-based models (Section 6.2), autoregressive models (Section 6.3), score-based models (Section 6.4), and diffusion models (Section 6.5). Finally, Section 6.6 examines how signal restoration problems can be framed as Bayesian inverse problems, where generative models act as priors that guide the restoration process.

6.1 Data Distributions

Generative modeling fundamentally aims to capture and replicate the patterns and structures inherent in observed data. Examples of such data include natural images, audio signals, or text, which exhibit specific, complex regularities that distinguish them from random noise. These patterns can be understood as being governed by an unknown data distribution

$p(x) : \mathcal{X} \to \mathbb{R}$, which assigns a probability to every sample x within the domain of possible signals $\mathcal{X}$. For example, in the context of image data, $\mathcal{X}$ represents the space of all possible images, and $p(x)$ represents the likelihood of each image within this space. Figure 6.1 illustrates the concept of such a data distribution for the MNIST (LeCun, 1998) dataset.

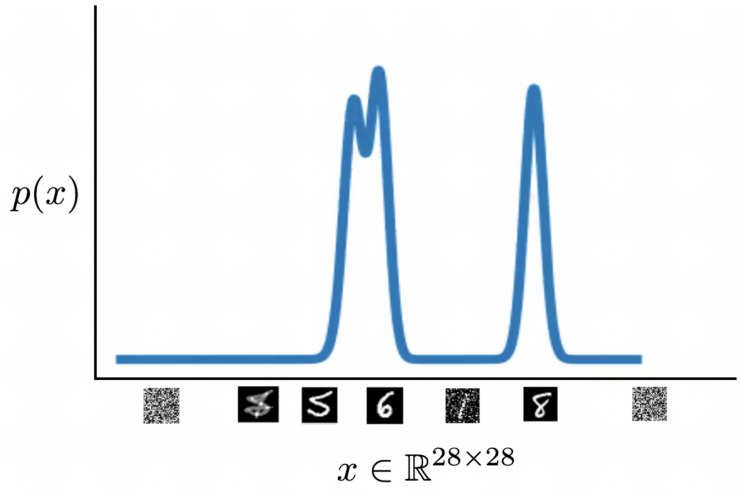


Figure 6.1: A visual illustration of a data distribution p for the MNIST (LeCun, 1998) dataset, simplified as a 1D line in this figure. p is a high-dimensional function that maps the set of all images in $\mathbb{R}^{28\times28} \to \mathbb{R}$ and represents a valid probability distribution. Realistic looking numbers are assigned high probability mass and noise is assigned no mass. The goal of generative modeling is to generate new samples from this distribution.

However, the true distribution p(x) is rarely accessible or explicitly defined in real-world scenarios. Therefore, generative models aim to approximate this underlying data distribution by learning a parameterized approximation $\hat{p}_\theta(x)$. Given a dataset of samples $\{x_1, x_2, \ldots, x_n\} \sim p(x)$, the objective of generative modeling is to produce new samples $x \sim \hat{p}_\theta(x)$ that closely resemble the true data distribution. For example, given many examples of songs, a generative model should be able to produce new songs. The ideal generative model minimizes the divergence between the true distribution $p(x)$ and its approximation

$\hat{p}_\theta(x)$. This can be formalized as:

$$\theta^* = \arg\min_\theta D(p\|\hat{p}_\theta), \tag{6.1}$$

where $D(\cdot\|\cdot)$ represents a divergence measure, such as Kullback-Leibler (KL) divergence. However, since the true distribution $p(x)$ is unknown, we cannot directly compute this divergence. Instead, generative models typically maximize the likelihood of the observed data under the model, a process known as Maximum Likelihood Estimation (MLE). The MLE objective is given by:

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \log \hat{p}_\theta(x_i),$$

which encourages the model to assign high probability to the training data. Naively maximizing the likelihood can lead to overfitting, as demonstrated by the trivial solution:

$$\hat{p}_\theta(x) = \begin{cases} \frac{1}{n}, & \text{if } x \in \{x_1, x_2, \ldots, x_n\}, \\ 0, & \text{otherwise.} \end{cases} \tag{6.2}$$

Such a model merely memorizes the training data and assigns zero probability elsewhere. To prevent this, practical implementations incorporate regularization techniques and underparameterized models, which encourage generalization beyond the training set (Goodfellow et al., 2016). Furthermore, evaluating the models can be achieved by measuring the predicted log-likelihood on a validation set of true data samples. We note that this is the same as measuring the cross entropy of the predicted distribution $\hat{p}_\theta$ relative to p denoted by $H(p, \hat{p}_\theta)$ (Theis et al., 2016b).

$$H(p, \hat{p}_\theta) = \mathop{\mathbb{E}}_{x\sim p} - \log \hat{p}_\theta(x) \approx -\frac{1}{n}\sum_{i=1}^{n} \log \hat{p}_\theta(x_i) \tag{6.3}$$

Implicit Generative Models: One thing to note is that generative models do not need to explicitly parametrize the distribution $\hat{p}_\theta$ in order to generate new samples. Implicit generative models, like Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) define a transformation $g : \mathcal{Z} \to \mathcal{X}$, where z is a sample from a common distribution, e.g.

$z \sim \text{Uniform}[0,1]$. The generator g_θ is then trained to produce samples $g_\theta(z) \sim \hat{p}_\theta(x)$ that mimic the data distribution $p(x)$. Such a function enables the generation of samples from $\hat{p}_\theta$ without providing an explicit function that can evaluate the probability $\hat{p}_\theta$ at any arbitrary point $x \in \mathcal{X}$. Although GANs have been used for signal restoration (Ledig et al., 2017; Yu et al., 2022), these methods are not Bayesian in nature and are not considered in this thesis.

Gradient-Based Sampling: Another way to avoid direct parameterization of $p(x)$ is by modeling the gradient of the data distribution $\nabla_x \log p(x)$, known as the score function. These score-based methods iteratively refine an initial noisy sample x_0 from a known distribution such as $\mathcal{N}(0, I)$ by guiding it toward regions of high probability using the gradient. For example, Langevin dynamics (Welling and Teh, 2011) updates the sample as:

$$x_{t+1} = x_t + \eta \nabla_x \log \hat{p}_\theta(x_t) + \sqrt{2\eta} \cdot \xi_t,$$

where η is a step size and $\xi_t \sim \mathcal{N}(0, I)$ is Gaussian noise. These approaches form the foundation of diffusion and score-based generative models, which are explored in subsequent sections.

In the following sections we explore specific generative modeling architectures in greater depth.

6.2 Flow-Based Generative Models

Flow-based generative models are a class of generative models that explicitly learn a data distribution $p(x)$ through a series of invertible transformations. Their ability to provide exact likelihood computation, coupled with efficient sampling, makes them particularly appealing for signal restoration and inverse problems.

Flow-based generative models transform a simple base distribution $q(z)$, such as a standard Gaussian, into a complex target distribution $p(x)$ using an invertible function g_θ:

$$x = g_\theta(z), \quad z \sim p(z)$$

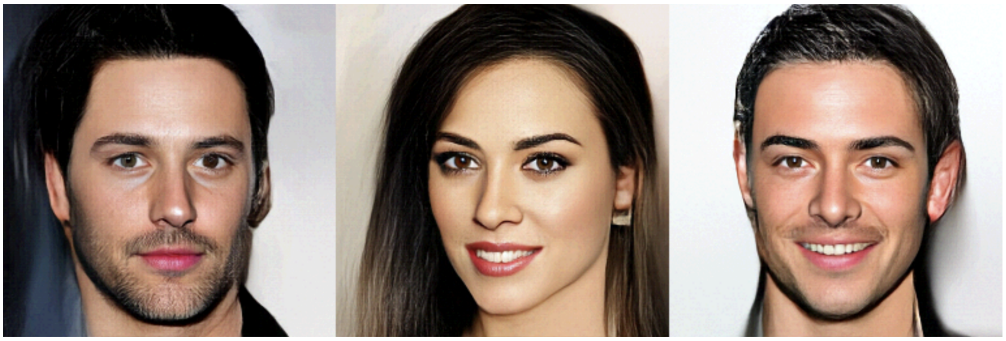


Figure 6.2: Examples of unconditional samples from the Glow (Kingma and Dhariwal, 2018a) model on the Celeb-A HQ (Karras et al., 2018) dataset.

Due to the invertibility of g_θ, the transformation can be reversed:

$$z = g_\theta^{-1}(x).$$

The change of variables formula allows us to compute the exact likelihood of a sample x:

$$p(x) = q(g_\theta^{-1}(x)) \left| \det \frac{\partial g_\theta^{-1}(x)}{\partial x} \right|,$$

where $\left| \det \frac{\partial g_\theta^{-1}(x)}{\partial x} \right|$ is the determinant of the Jacobian of the inverse transformation. Since g_θ is parameterized as a neural network, the primary design challenge lies in ensuring that g_θ is invertible and that the Jacobian determinant is computationally efficient to compute. Popular architectures for flow-based models include:

- Real NVP (Non-Volume Preserving) (Dinh et al., 2016): Introduces coupling layers to ensure efficient computation of the Jacobian determinant.

- Glow (Kingma and Dhariwal, 2018a): Extends Real NVP with more expressive transformations, including 1x1 invertible convolutions. Glow is used as a generative prior for source separation in Chapter 7. We highlight unconditional samples from Glow in Figure 6.2

- Flow++ (Ho et al., 2019): Improves upon Glow by incorporating variational dequantization and softmax-based coupling layers for better image modeling.

Flow models provide exact likelihoods and are also fast to sample from in a single step compared to diffusion or score-based models. However, designing invertible transformations that balance expressiveness and efficiency is non-trivial. Compared to implicit models like GANs, flow models may require more layers to capture highly complex distributions (Ho et al., 2019), leading to longer training and worse performance. Flow models have been explored extensively for audio generation. The Parallel WaveNet (van den Oord et al., 2018b) distills an autoregressive WaveNet (Oord et al., 2016) into flow model using a student-teacher paradigm. More recently, models such as WaveGlow (Prenger et al., 2018), WaveFlow (Ping et al., 2020b), and FlowWaveNet (Kim et al., 2019) have been proposed for audio and music generation through generative flow.

6.3 Autoregressive Generative Models

Autoregressive generative models are designed for modeling sequence data (e.g., discrete sound waveforms, text, pixels), and parameterize the data distribution $p(x)$ as a product of conditional probabilities over each sequence element. As a result, these models can capture the dependencies between elements of a sequence in a highly expressive manner.

We consider a data sample $x \in \mathbb{R}^T$ where there are T elements in the sequence each of dimension d. The likelihood of the overall sequence is computed as:

$$p(x) = \prod_{i=1}^{T} p(x_i | x_{<i}),$$

where $x_{<i}$ denotes all preceding variables in a chosen ordering. This factorization of the likelihood ensures that the joint distribution is modeled in a way that respects the sequential dependencies between each x_i.

Autoregressive models are trained by maximizing the likelihood of observed data samples.

Figure 6.3: Generated samples from the model in Li et al. (2024a), which uses an autoregressive model on a continuous latent space.

The objective for a dataset of sequences $\{x_1, x_2, \ldots, x_n\} \sim p(x)$ is:

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{T} \log p_\theta(x_{j,i} | x_{j,<i}),$$

where $p_\theta(x_i|x_{<i})$ is parameterized by a neural network. During training, the model learns to predict each variable x_i conditioned on the preceding variables $x_{<i}$. When generating a new sample, we sequentially sample from $x_t \sim p_\theta(\cdot|x_{<t})$ for each $0 \leq t < T$. Note that sampling from an autoregressive model is slow, as each new x_t must be generated sequentially with dependence on the previous generations. However, training is fast since the entirety of the training sequence is known, and the MLE can be computed in parallel across all x_i within a given sequence.

Several architectures have been proposed to implement autoregressive models efficiently. In the image domain, PixelCNN (van den Oord et al., 2016c) was an early model for autoregressive generation. It uses convolutional layers with masked filters to ensure that each pixel is generated only from previously generated pixels. PixelCNN++ (Salimans et al., 2017) improved the architecture through skip connections and a different softmax output.

We use PixelCNN++ as a prior for source separation in Chapter 8. In the audio domain, a notable early architecture was WaveNet (Oord et al., 2016). WaveNet uses dilated convolutional layers to efficiently model long-range dependencies in audio signals The architecture of WaveNet is shown in Figure 6.4, and it is used as a prior for source separation in Chapter 8.

Due to the large number of sequence elements to model in images and audio, recent autoregressive methods have opted to generate from a latent representation instead of pixels/audio samples directly. Methods in the audio domain include Jukebox (Dhariwal et al., 2020b), MusicGen (Copet et al., 2024), and AudioLM (Borsos et al., 2023). In the image domain, models such as ImageFolder (Li et al., 2024b), ImageBart (Esser et al., 2021), and the work in Li et al. (2024a) generate images through autoregressive generation in a latent space.

Transformer architectures (Vaswani et al., 2017) have become the dominant approach for autoregressive modeling across domains due to their ability to capture long-range dependencies through self-attention mechanisms. Large Language Models (LLMs) like GPT (Brown et al., 2020a) and multimodal transformers such as DALL-E (Ramesh et al., 2021) exemplify how the transformer architecture, when scaled appropriately, can model complex distributions autoregressively with remarkable fidelity. These models maintain the same autoregressive factorization of the likelihood while replacing convolutional architectures with attention-based ones, offering superior scaling properties for modeling long sequences.

6.4 Score-Based Generative Models

Score-based generative models take a fundamentally different approach to density estimation by learning the score function $\nabla_x \log p(x)$ of the data distribution rather than the density $p(x)$ directly. This gradient-based representation enables powerful sampling procedures through Langevin dynamics while avoiding the architectural constraints of normalizing flows or the sequential generation of autoregressive models.

The key insight of score-based models is that we can learn the score function through score matching (Hyvärinen, 2005), which minimizes the Fisher divergence between the model

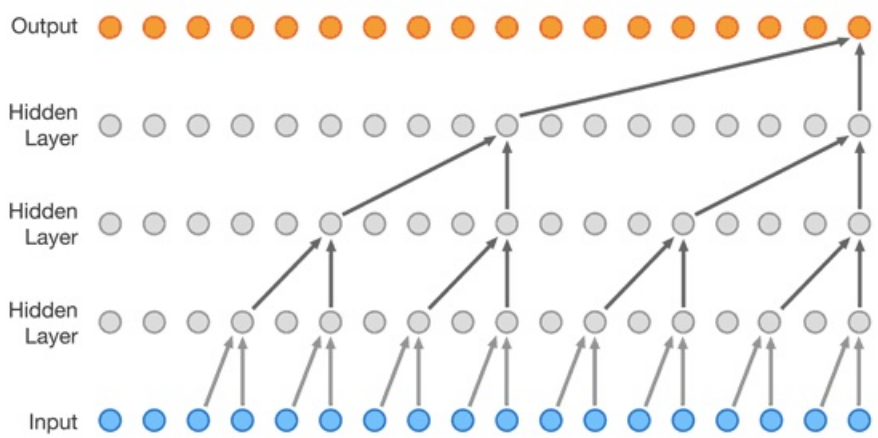


Figure 6.4: The WaveNet architecture (Oord et al., 2016). Dilated convolutions are used to efficiently model long-term dependencies in the audio signal.

distribution $p_\theta(x)$ and data distribution $p(x)$:

$$J(\theta) = \frac{1}{2}\mathbb{E}_{p(x)}\left[\|\nabla_x \log p_\theta(x) - \nabla_x \log p(x)\|_2^2\right] \tag{6.4}$$

However, computing $\nabla_x \log p(x)$ requires access to the true data distribution. Denoising score matching (Vincent, 2011) provides a practical alternative by learning from noisy versions of the data:

$$L(\theta) = \frac{1}{2}\mathbb{E}_{p(x)}\left[\mathbb{E}_{p_\sigma(\tilde{x}|x)}\left\|\nabla_{\tilde{x}} \log p_\theta(\tilde{x}) + \frac{\tilde{x} - x}{\sigma^2}\right\|_2^2\right] \tag{6.5}$$

where $p_\sigma(\tilde{x}|x)$ is a Gaussian perturbation kernel with standard deviation σ.

A major advancement came with noise-conditioned score networks (Song and Ermon, 2019), which learn the score across multiple noise levels simultaneously:

$$L(\theta) = \mathbb{E}_{\sigma\sim p(\sigma)}\mathbb{E}_{p(x)}\mathbb{E}_{p_\sigma(\tilde{x}|x)}\left[\|s_\theta(\tilde{x}, \sigma) + \frac{\tilde{x} - x}{\sigma^2}\|_2^2\right] \tag{6.6}$$

where $s_\theta(x, \sigma)$ is a neural network that estimates the score at noise level σ. This multi-scale approach enables stable sampling by gradually denoising from a high noise level where

the distribution is nearly Gaussian. The sampling process follows an annealed Langevin dynamics:

$$x_{t+1} = x_t + \eta s_\theta(x_t, \sigma_t) + \sqrt{2\eta}\,\xi_t, \quad \xi_t \sim \mathcal{N}(0, I) \tag{6.7}$$

where σ_t is a decreasing sequence of noise levels.

Figure 6.5: Visualization of the score-based sampling process from Song and Ermon (2019) Starting from Gaussian noise (left), the model gradually denoises the sample through annealed Langevin dynamics to produce a clean image (right).

For computational efficiency, sliced score matching (Song et al., 2019) reduces the dimensionality of score estimation by projecting onto random vectors v:

$$L(\theta, v) \equiv \mathbb{E}_{p(x)}\left[\frac{1}{2}\left(\mathbf{v}^T s_\theta(\mathbf{x}) - \mathbf{v}^T \nabla_{\mathbf{x}} \log p(\mathbf{x})\right)^2\right] \tag{6.8}$$

This approach scales better to high dimensions while maintaining the theoretical guarantees of score matching.

Unlike flow-based models and autoregressive models, score-based approaches do not provide explicit density estimation. However, access to $\nabla_x \log p(x)$ is often sufficient for signal restoration as we explore in Section 6.6. The main computational challenge lies in the number

of function evaluations required during the sampling procedure, though recent work on accelerated sampling (Jolicoeur-Martineau et al., 2021) and distillation (Luhman and Luhman, 2021) has significantly improved efficiency.

6.5 Diffusion Models

Closely related to score-based models are diffusion models, including Denoising Diffusion Probabilistic Models (DDPMs)(Ho et al., 2020) and Denoising Diffusion Implicit Models (DDIMs)(Song et al., 2020a). Like score-based models, these provide a principled framework for generative modeling by iteratively transforming noise into data through a series of learned reverse diffusion steps. However, they do not explicitly learn $\nabla_x \log p(x)$ in the same way. Instead, these models can be understood as a probabilistic formulation of score-based generative models, where the score function $\nabla_x \log p(x)$ is implicitly modeled through a noise-corrupted data sequence.

DDPM models a data distribution $q(x_0)$ by constructing a forward diffusion process that progressively adds Gaussian noise to a data sample. The forward process defines a sequence of smoothed distributions:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1-\alpha_t)I), \tag{6.9}$$

where α_t is a monotonically decreasing noise schedule with $\alpha_0 = 1$ (no noise) and $\alpha_T = 0$ (pure Gaussian noise). This sequence ensures that $q(x_T) \approx \mathcal{N}(0, I)$.

The generative process learns a reverse Markov chain $p_\theta(x_{t-1}|x_t)$ to map noisy samples back to the data distribution. The reverse process is defined as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{6.10}$$

where μ_θ and Σ_θ are parameterized by a neural network. In the simplified DDPM formulation, Σ_θ is often fixed to a diagonal covariance matrix, reducing the complexity of the model.

Figure 6.6: Example images from the DPPM algorithm (Ho et al., 2020)

To improve sampling efficiency, DDIMs reformulate the generative process as a deterministic mapping using non-Markovian transitions. The forward process is reinterpreted as:

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{6.11}$$

where the reverse process follows a deterministic update:

$$x_{t-1} = \sqrt{\alpha_{t-1}}\hat{x}_0 + \sqrt{1 - \alpha_{t-1}}\epsilon_t, \quad \hat{x}_0 = \frac{1}{\sqrt{\alpha_t}}(x_t - \sqrt{1 - \alpha_t}\epsilon_\theta(x_t, t)). \tag{6.12}$$

Here, $\epsilon_\theta(x_t, t)$ predicts the noise component added during the forward process, enabling direct control over the quality-speed tradeoff by adjusting the number of reverse steps.

Diffusion models have shown exceptional performance across various domains, including image synthesis (Dhariwal and Nichol, 2021), audio generation (Kong et al., 2021b), and even 3D modeling (Poole et al., 2022). However, the iterative nature of the reverse process makes sampling computationally expensive, necessitating advancements such as fast ODE solvers (Song et al., 2021) and numerical methods(Liu et al., 2022) to reduce the number of reverse steps required. Like autoregressive models, diffusion models have also benefited from generation in a latent space. Most notable is Stable Diffusion (Rombach et al., 2022), which uses a VAE (Kingma and Welling, 2022) and then learns a diffusion model on the lower-dimensional latent representation.

6.6 Signal Restoration as Bayesian Inverse Problems

Generative models that provide access to the probability distribution $p(x)$ or score function $\nabla_x \log p(x)$ have emerged as powerful tools for signal restoration tasks (GM et al., 2020). Signal restoration tasks can be naturally framed within a Bayesian framework, where we seek to infer a clean signal x from its noisy or incomplete observation y. The central principle of Bayesian inference is encapsulated in Bayes' rule, which provides a way to compute the posterior distribution $p(x|y)$:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}, \tag{6.13}$$

where:

- $p(x)$: The prior distribution, which encodes our prior knowledge or assumptions about the clean signal x. This is provided by the generative model and discussed in the previous chapter.

- $p(y|x)$: The likelihood function, which describes how the observed signal y is generated from the clean signal x. This ensures fidelity to our partial observation y.

- $p(y)$: The partition function, a normalization constant ensuring that the posterior is a valid probability distribution.

6.6.1 Modeling the Likelihood Function

The likelihood function $p(y|x)$ defines how well our partial observation y is explained by a candidate x and needs to be carefully balanced with the prior $p(x)$. For the problem formulations explored in this thesis, we assume that we know the underlying observation function $g(x)$ (e.g. downsampling, corruption) that was used to generate y based on the ground truth signal x^*

$$y = g(x^*) \tag{6.14}$$

$g(x)$ need not be linear, but it is non-invertible otherwise the original signal could be trivially computed. The intuition behind the likelihood function is that we would like any generated output $\hat{x}$ to also satisfy the observation: $y = g(\hat{x})$

For tasks where the observation process is deterministic, the likelihood can be modeled as a Dirac delta function:

$$p(y|x) = \delta(y - g(x)), \tag{6.15}$$

This enforces a hard consistency for generated outputs to match the partial observation. However, the gradient of this log-likelihood is not defined, and cannot be used in gradient based sampling algorithms. Furthermore, the posterior likelihood $p(x|y)$ is 0 anywhere without a perfect reconstruction. This also makes it hard to find samples with a good prior probability $p(x)$, since any sampling algorithm will be unable to explore regions without perfect reconstruction. To avoid non-differentiability, it is common to relax the hard constraint to a soft constraint where the likelihood is gaussian cenetered on $g(x)$

$$p(y|x) = \mathcal{N}(y; g(x), \sigma^2 I), \tag{6.16}$$

where σ^2 is a chosen parameter. Note that in the limit, we recover the hard constraint:

$$\lim_{\sigma \to 0} \; p(y|x) = \delta(y - g(x)) \tag{6.17}$$

With this relaxation, the likelihood is differentiable throughout the sampling process. The corresponding log-likelihood gradient is:

$$\nabla_x \log p(y|x) = -\frac{1}{\sigma^2} \nabla_x \|y - g(x)\|^2. \tag{6.18}$$

We also note that a soft guassian likelihood is the correct likelihood in a non-deterministic corruption processes. For example, our output may be corrupted in the presence of random noise:

$$y = g(x) + \mathcal{N}(0, \sigma_y^2) \tag{6.19}$$

In this case, the relaxed likelihood encapsulates the uncertainty in the observation process, and a hard constraint to enforce $y = g(x)$ would cause overfitting to the observational noise. This noisy problem formulation is explored in Chapter 9.

6.6.2 Gradient-Based Sampling for Inverse Problems

The goal in signal restoration is to estimate the clean signal x given the observed y, which can be achieved by sampling from the posterior $p(x|y)$. However, exact sampling is often intractable due to the high dimensionality of x, and the difficulty of computing the partition function $p(y)$. A more practical approach is to sample from the posterior through gradient-based methods. The gradient of the posterior distribution $\nabla_x \log p(x|y)$ provides a principled way to iteratively refine an estimate of x towards desired outputs. Using Bayes' rule, the posterior gradient can be expressed as:

$$\nabla_x \log p(x|y) = \nabla_x \log p(x) + \nabla_x \log p(y|x), \tag{6.20}$$

where:

- $\nabla_x \log p(x)$: Guides the solution towards regions of high prior probability, ensuring that the restored signal is plausible. This can be obtained by backpropagating through an explicit generative model, or is available directly from score-based and diffusion models.

- $\nabla_x \log p(y|x)$: Enforces consistency with the observed signal y, ensuring the reconstruction aligns with the given data. As discussed, this can be a hard or soft constraint.

Notice that the dependency on the partition function $p(y)$ has disappeared as the gradient with respect to x is 0.

Although we can easily calculate $\nabla_x \log p(x|y)$ with the Equation 6.20, generating outputs from the posterior distribution or reasonable outputs at all is still challenging. The first thought is to try to compute the Maximum a Posteriori (MAP) estimate, which seeks the mode of the posterior.

$$\hat{x} = \arg\max_x \ \log p(x|y) \tag{6.21}$$

Naive attempts at gradient ascent fail catastrophically due to the extreme non-convex and non-smooth nature of the loss landscape (see Figure 6.7). For example, suppose we randomly

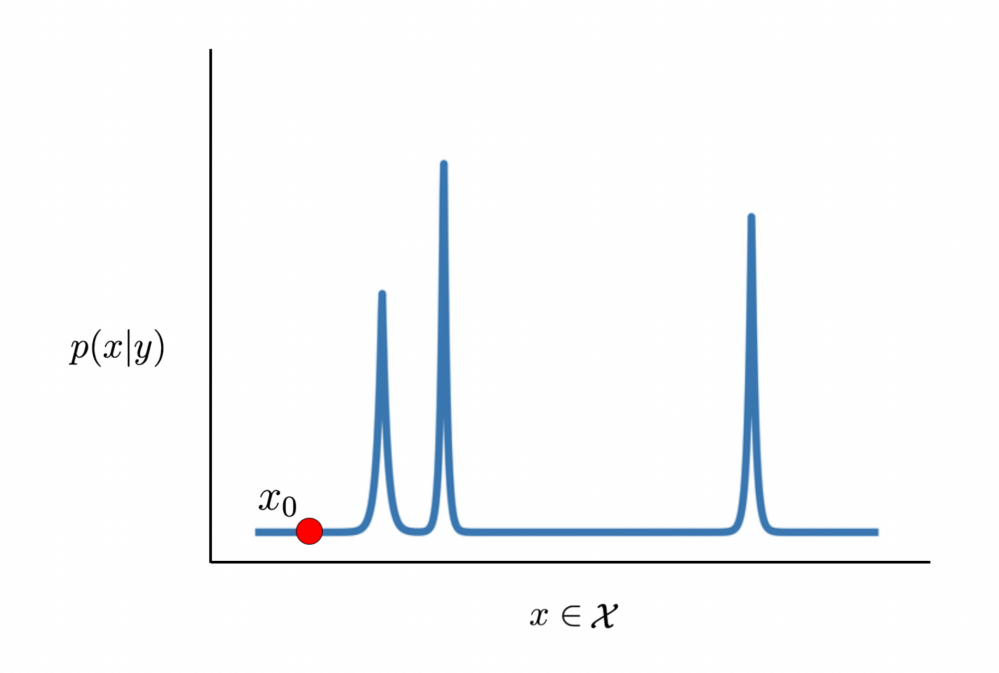


Figure 6.7: An illustration of the difficulty of sampling from $p(x|y)$ even when we can compute the gradient $\nabla_x \log p(x|y)$. A random initialization x_0 will almost certainly start in a spot with near zero likelihood and no meaningful gradient. Furthermore, naive gradient ascent will get stuck at local maxima, which may produce suboptimal or non-representative samples.

initialize our guess x_0 at $\mathcal{N}(0, I)$. We note that $p(x|y) = 0$ and in a local neighborhood $x + \epsilon$, we also find that $p(x + \epsilon|y) = 0$. Therefore, we obtain no reasonable guidance through $\nabla_x \log p(x|y)$. Naive optimization also has a tendency to get stuck at local minima, leading to sub-par outputs (Jayaram and Thickstun, 2020).

Another option is stochastic sampling methods, such as Langevin dynamics (Welling and Teh, 2011). These use the posterior gradient to iteratively refine a sample:

$$x_{t+1} = x_t + \eta \nabla_x \log p(x|y) + \sqrt{2\eta} \cdot \xi_t, \tag{6.22}$$

where η is the step size and $\xi_t \sim \mathcal{N}(0, I)$ is Gaussian noise. Although Langevin dynamics can generate true samples from the posterior distribution in the limit, i.e. $\lim_{t \to \infty} x_t \sim p(x|y)$,

in practice the convergence is extremely slow with highly non-convex posterior distributions (Song and Ermon, 2019). This challenge of sampling from the posterior in an efficient way while balancing the prior and likelihood gradients is a central focus of the following chapters.

Chapter 7

SOURCE SEPARATION WITH DEEP GENERATIVE PRIORS

The first inverse problem we tackle with generative models is source separation. The single-channel source separation problem (Davies and James, 2007) asks us to decompose a mixed signal $\mathbf{m} \in \mathcal{X}$ into a linear combination of k components $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathcal{X}$ with scalar mixing coefficients $\alpha_i \in \mathbb{R}$:

$$\mathbf{m} = g(\mathbf{x}) \equiv \sum_{i=1}^{k} \alpha_i \mathbf{x}_i. \tag{7.1}$$

This is motivated by, for example, the "cocktail party problem" of isolating the utterances of individual speakers $\mathbf{x}_i$ from an audio mixture $\mathbf{m}$ captured at a busy party, where multiple speakers are talking simultaneously.

With no further constraints or regularization, solving Equation Equation (7.1) for $\mathbf{x}$ is highly underdetermined. Classical "blind" approaches to single-channel source separation resolve this ambiguity by privileging solutions to Equation (7.1) that satisfiy mathematical constraints on the components $\mathbf{x}$, such as statistical independence (Davies and James, 2007) sparsity (Lee et al., 1999) or non-negativity (Lee and Seung, 1999). These constraints can be be viewed as weak priors on the structure of sources, but the approaches are blind in the sense that they do not require adaptation to a particular dataset.

Recently, most works have taken a data-driven approach. To separate a mixture of sources, it is natural to suppose that we have access to samples $\mathbf{x}$ of individual sources, which can be used as a reference for what the source components of a mixture are supposed to look like. This data can be used to regularize solutions of Equation Equation (7.1) towards structurally plausible solutions. The prevailing way to do this is to construct a supervised regression model that maps an input mixture $\mathbf{m}$ to components $\mathbf{x}_i$ (Huang et al., 2014; Halperin et al., 2018). Paired training data $(\mathbf{m}, \mathbf{x})$ can be constructed by summing

randomly chosen samples from the component distributions $\mathbf{x}_i$ and labeling these mixtures with the ground truth components.

Instead of regressing against components $\mathbf{x}$, we use samples to train a generative prior $p(\mathbf{x})$; we separate a mixed signal $\mathbf{m}$ by sampling from the posterior distribution $p(\mathbf{x}|\mathbf{m})$. For some mixtures this posterior is quite peaked, and sampling from $p(\mathbf{x}|\mathbf{m})$ recovers the only plausible separation of $\mathbf{m}$ into likely components. But in many cases, mixtures are highly ambiguous: see, for example, the orange-highlighted MNIST images in Figure 7.1. This motivates our interest in sampling, which explores the space of plausible separations. In Section 7.2 we introduce a procedure for sampling from the posterior, an extension of the noise-annealed Langevin dynamics introduced in Song and Ermon (2019), which we call Bayesian Annealed SIgnal Source separation: "BASIS" separation.

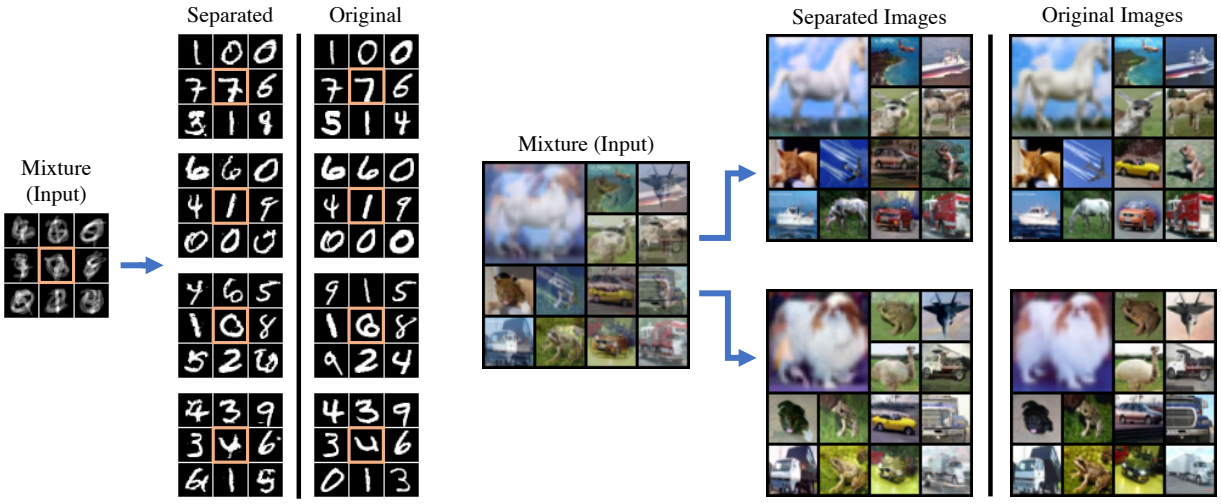


Figure 7.1: Separation results for mixtures of four images from the MNIST dataset (Left) and two images from the CIFAR-10 dataset (Right), using BASIS with the NCSN (Song and Ermon, 2019) generative model as a prior over images. We draw attention to the central panel of the MNIST results (highlighted in orange), which shows how a mixture can be separated in multiple ways.

Ambiguous mixtures pose a challenge for traditional source separation metrics, which presume that the original mixture components are identifiable and compare the separated components to ground truth. For ambiguous mixtures of rich data, we argue that recovery of the original mixture components is not a well-posed problem. Instead, the problem we aim to solve is finding components of a mixture that are consistent with a particular data distribution. Motivated by this perspective, we discuss evaluation metrics in Section 7.3.

Formulating the source separation problem in a Bayesian framework decouples the problem of source generation from source separation. This allows us to leverage pre-trained, state-of-the-art, likelihood-based generative models as prior distributions, without requiring architectural modifications to adapt these models for source separation. Examples of source separation using noise-conditioned score networks (NCSN) (Song and Ermon, 2019) as a prior are presented in Figure 7.1. Further separation results using NCSN and Glow (Kingma and Dhariwal, 2018b) are presented in Section 7.4.

7.1 Related Work

Blind separation. Work on blind source separation is data-agnostic, relying on generic mathematical properties to privilege particular solutions to Equation (7.1) Comon (1994); Bell and Sejnowski (1995); Davies and James (2007); Huang et al. (2012). Because blind methods have no access to sample components, they face the challenging task of modeling the distribution over unobserved components while simultaneously decomposing mixtures into likely components. It is difficult to fit a rich model to latent components, so blind methods often rely on simple models such as dictionaries to capture the structure of these components.

One promising recent work in the blind setting is Double-DIP Gandelsman et al. (2019). This work leverages the unsupervised Deep Image Prior Ulyanov et al. (2018) as a prior over signal components, similar to our use of a trained generative model. But the authors of this work document fundamental obstructions to applying their method to single-channel source separation; they propose using multiple image frames from a video, or multiple mixtures of

the same components with different mixing coefficients α. This multiple-mixture approach is common to much of the work on blind separation. In contrast, our approach is able to separate components from a single mixture.

Supervised regression. Regression models for source separation learn to predict components for a mixture using a dataset of mixed signals labeled with ground truth components. This approach has been extensively studied for separation of images Halperin et al. (2018), audio spectrograms Huang et al. (2014, 2015); Nugraha et al. (2016); Jansson et al. (2017), and raw audio Lluis et al. (2019); Stoller et al. (2018b); Défossez et al. (2019b), as well as more exotic data domains, e.g. medical imaging Nishida et al. (1999). By learning to predict components (or equivalently, masks on a mixture) this approach implicitly builds a generative model of the signal components. This connection is made more explicit in recent work that uses GAN's to force components emitted by a regression model to match the distribution of a given dataset Zhang et al. (2018b); Stoller et al. (2018a).

The supervised approach takes advantage of expressive deep models to capture a strong prior over signal components. But it requires specialized model architectures trained specifically for the source separation task. In contrast, our approach leverages standard, pre-trained generative models for source separation. Furthermore, our approach can directly exploit ongoing advances in likelihood-based generative modeling to improve separation results.

Signal Dictionaries. Much work on source separation is based on the concept of a signal dictionary, most notably the line of work based on non-negative matrix factorization (NMF) Lee and Seung (2001). These approaches model signals as combinations of elements in a latent dictionary. Decomposing a mixture into dictionary elements can be used for source separation by (1) clustering the elements of the dictionary and (2) reconstituting a source using elements of the decomposition associated with a particular cluster.

Dictionaries are typically learned from data of each source type and combined into a joint dictionary, clustered by source type Schmidt and Olsson (2006); Virtanen (2007). The blind setting has also been explored, where the clustering is obtained without labels by e.g. k-means Spiertz and Gnann (2009). Recent work explores more expressive decomposition

models, replacing the linear decompositions used in NMF with expressive neural autoencoders Smaragdis and Venkataramani (2017); Venkataramani et al. (2017).

When the dictionary is learned with supervision from labeled sources, dictionary clusters can be interpreted as implicit priors on the distributions over components. Our approach makes these prior explicit, and works with generic priors that are not tied to the dictionary model. Furthermore, our method can separate mixed sources of the same type, whereas mixtures of sources with similar structure present a conceptual difficulty for dictionary-based methods.

Generative adversarial separation. Recent work by Subakan and Smaragdis (2018) and Kong et al. (2019) explores the intriguing possibility of optimizing $\mathbf{x}$ given a mixture $\mathbf{m}$ to satisfy Equation (7.1), where components $\mathbf{x}_i$ are constrained to the manifold learned by a GAN. The GAN is pre-trained to model a distribution over components. Like our method, this approach leverages modern deep generative models in a way that decouples generation from source separation. We view this work as a natural analog to our likelihood-based approach in the GAN setting.

Likelihood-based approaches. Our approach is similar in spirit to older ideas based on maximum a posteriori estimation Geman and Geman (1984) likelihood maximization Pearlmutter and Parra (1997); Roweis (2001) and Bayesian source separation Benaroya et al. (2005). We build upon their insights, with the advantage of increased computational resources and modern expressive generative models.

7.2 BASIS Separation

We consider the following generative model of a mixed signal $\mathbf{m}$, relaxing the mixture constraint $g(\mathbf{x}) = \mathbf{m}$ to a soft Gaussian approximation:

$$\mathbf{x} \sim p, \tag{7.2}$$

$$\mathbf{m} \sim \mathcal{N}\left(g(\mathbf{x}), \gamma^2 I\right). \tag{7.3}$$

This defines a joint distribution $p_\gamma(\mathbf{x}, \mathbf{m}) = p(\mathbf{x})p_\gamma(\mathbf{m}|\mathbf{x})$ over signal components $\mathbf{x}$ and mixtures $\mathbf{m}$, and a corresponding posterior distribution

$$p_\gamma(\mathbf{x}|\mathbf{m}) = p(\mathbf{x})p_\gamma(\mathbf{m}|\mathbf{x})/p_\gamma(\mathbf{m}). \tag{7.4}$$

In the limit as $\gamma^2 \to 0$, we recover the hard constraint on the mixture $\mathbf{m}$ given by Equation Equation (7.1).

BASIS separation (Algorithm 3) presents an approach to sampling from Equation (7.4) based on the discussion in Sections 7.2.1 and 7.2.2. In Section 9.3.4 we discuss the behavior of the gradients $\nabla_\mathbf{x} \log p(\mathbf{x})$, which motivates some of the hyper-parameter choices in Section 7.2.4. We describe a procedure to construct the noisy models p_{σ_i} required for BASIS in Section 7.2.5.

7.2.1 Langevin dynamics

Sampling from the posterior distribution $p_\gamma(\mathbf{x}|\mathbf{m})$ looks formidable; just computing Equation Equation (7.4) requires evaluation of the partition function $p_\gamma(\mathbf{m})$. But using Langevin dynamics (Neal et al., 2011; Welling and Teh, 2011) we can sample $\mathbf{x} \sim p_\gamma(\cdot|\mathbf{m})$ while avoiding explicit computation of $p_\gamma(\mathbf{x}|\mathbf{m})$. Let $\mathbf{x}_0 \sim \text{Uniform}(\mathcal{X})$, $\varepsilon_t \sim \mathcal{N}(0, I)$, and define a sequence

$$\begin{aligned} \mathbf{x}^{(t+1)} &\equiv \mathbf{x}^{(t)} + \eta \nabla_\mathbf{x} \log p_\gamma(\mathbf{x}^{(t)}|\mathbf{m}) + \sqrt{2\eta}\varepsilon_t \\ &= \mathbf{x}^{(t)} + \eta \nabla_\mathbf{x} \left(\log p(\mathbf{x}^{(t)}) + \tfrac{1}{2\gamma^2} \|\mathbf{m} - g(\mathbf{x}^{(t)})\|^2 \right) + \sqrt{2\eta}\varepsilon_t. \end{aligned} \tag{7.5}$$

Observe that $\nabla_\mathbf{x} \log p_\gamma(\mathbf{m}) = 0$, so this term is not required to compute Equation (8.2). By standard analysis of Langevin dynamics, as the step size $\eta \to 0$, $\lim_{t\to\infty} D_{\text{KL}}\left(\mathbf{x}_t \,\|\, \mathbf{x}|\mathbf{m}\right) = 0$, under regularity conditions on the distribution $p_\gamma(\mathbf{x}|\mathbf{m})$.

If the prior $p(\mathbf{x})$ is parameterized by a neural model, then gradients $\nabla_\mathbf{x} \log p(\mathbf{x})$ can be computed by automatic differentiation with respect to the inputs of the generator network. This family of likelihood-based models includes autoregressive models (Salimans et al., 2017; Parmar et al., 2018), the variational autoencoder (Kingma and Welling, 2014b; van den

Oord et al., 2017b), or flow-based models (Dinh et al., 2016; Kingma and Dhariwal, 2018b). Alternatively, if gradients of the distribution are modeled (Song and Ermon, 2019), then $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ can be used directly.

7.2.2 Accelerated mixing

To accelerate mixing of Equation (8.2) we adopt a simulated annealing schedule over noisy approximations to the model $p(\mathbf{x})$, extending the unconditional sampling algorithm proposed in Song and Ermon (2019) to accelerate sampling from the posterior distribution $p_\gamma(\mathbf{x}|\mathbf{m})$. Let $p_\sigma(\mathbf{x})$ denote the distribution of $\mathbf{x}+\epsilon_\sigma$ for $\mathbf{x} \sim p$ and $\epsilon_\sigma \sim \mathcal{N}(0, \sigma^2 I)$. We define the noisy joint likelihood $p_{\sigma,\gamma}(\mathbf{x}, \mathbf{m}) \equiv p_\sigma(\mathbf{x}) p_\gamma(\mathbf{m}|\mathbf{x})$, which induces a noisy posterior approximation $p_{\sigma,\gamma}(\mathbf{x}|\mathbf{m})$. At high noise levels σ, $p_\sigma(\mathbf{x})$ is approximately Gaussian and irreducible, so the Langevin dynamics Equation (8.2) will mix quickly. And as $\sigma \to 0$, $D_{\mathrm{KL}}\left(p_\sigma \,\|\, p\right) \to 0$. This motivates defining the modified Langevin dynamics

$$\mathbf{x}^{(t+1)} \equiv \mathbf{x}^{(t)} + \eta \nabla_{\mathbf{x}} \log p_{\sigma,\gamma}(\mathbf{x}^{(t)}|\mathbf{m}) + \sqrt{2\eta}\varepsilon_t. \tag{7.6}$$

The dynamics Equation (7.6) approximate samples from $p(\mathbf{x}|g(\mathbf{x}) = \mathbf{m})$ as $\eta \to 0$, $\gamma^2 \to 0$, $\sigma^2 \to 0$, and $t \to \infty$. An implementation of these dynamics, annealing η, γ^2, and σ^2 as $t \to \infty$ according to the hyper-parameter settings presented in Section 7.2.4, is presented in Algorithm 3.

We anneal η, γ^2, and σ^2 using a heuristic introduced in Song and Ermon (2019): the idea is to maintain a constant signal-to-noise ratio (SNR) between the expected size of the posterior log-likelihood gradient term $\eta \nabla_{\mathbf{x}} \log p_{\sigma,\gamma}(\mathbf{x}|\mathbf{m})$ and the expected size of the Langevin noise $\sqrt{2\eta}\varepsilon$:

$$\begin{aligned} &\mathop{\mathbb{E}}_{\mathbf{x}\sim p_\sigma}\left[\left\| \frac{\eta \nabla_{\mathbf{x}} \log p_{\sigma,\gamma}(\mathbf{x}|\mathbf{m})}{\sqrt{2\eta}} \right\|^2\right] \\ &\quad = \frac{\eta}{4} \mathop{\mathbb{E}}_{\mathbf{x}\sim p_\sigma}\left[\|\nabla_{\mathbf{x}} \log p_\gamma(\mathbf{m}|\mathbf{x}) + \nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})\|^2\right]. \end{aligned} \tag{7.7}$$

Assuming that gradients w.r.t. to the likelihood and the prior are uncorrelated, the SNR is

Algorithm 3 BASIS Separation

Input: $\mathbf{m} \in \mathcal{X}$, $\{\sigma_i\}_{i=1}^{L}$, δ, T
Sample $\mathbf{x}_1, \ldots, \mathbf{x}_k \sim \text{Uniform}(\mathcal{X})$
for $i \leftarrow 1$ to L **do**
    $\eta_i \leftarrow \delta \cdot \sigma_i^2 / \sigma_L^2$
    for $t = 1$ to T **do**
        Sample $\varepsilon_t \sim \mathcal{N}(0, I)$
        $\mathbf{u}^{(t)} \leftarrow \mathbf{x}^{(t)} + \eta_i \nabla_{\mathbf{x}} \log p_{\sigma_i}(\mathbf{x}^{(t)}) + \sqrt{2\eta}\varepsilon_t$
        $\mathbf{x}^{(t+1)} \leftarrow \mathbf{u}^{(t)} - \frac{\eta_i}{\sigma_i^2}(\alpha) \left(\mathbf{m} - g(\mathbf{x}^{(t)})\right)$
    end for
end for

approximately

$$\frac{\eta}{4} \mathop{\mathbb{E}}_{\mathbf{x} \sim p_\sigma} \left[\|\nabla_{\mathbf{x}} \log p_\gamma(\mathbf{m}|\mathbf{x})\|^2 \right] + \frac{\eta}{4} \mathop{\mathbb{E}}_{\mathbf{x} \sim p_\sigma} \left[\|\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})\|^2 \right]. \tag{7.8}$$

Observe that $\log p_\gamma(\mathbf{m}|\mathbf{x})$ is a concave quadratic with smoothness proportional to $1/\gamma^2$; it follows analytically that $\mathbb{E}\left[\|\nabla_{\mathbf{x}} \log p_\gamma(\mathbf{m}|\mathbf{x})\|^2 \right] \propto 1/\gamma^2$. Song and Ermon (2019) found empirically that $\mathbb{E}\|\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})\|^2 \propto 1/\sigma^2$ for the NCSN model; we observe similar behavior for the flow-based Glow model (Kingma and Dhariwal, 2018b) and in Section 9.3.4 we propose a possible explanation for this behavior. Therefore, to maintain a constant SNR, it suffices to set both γ^2 and σ^2 proportional to η.

7.2.3 The gradients of the noisy prior

We remark that the empirical finding $\mathbb{E}\|\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})\|^2 \propto 1/\sigma^2$ discussed in Section 7.2.2, and the consistency of this observation across models and datasets, could be surprising. Gradients of the noisy densities p_σ can be described by convolution of p with a Gaussian kernel:

$$\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x}) = \nabla_{\mathbf{x}} \log \mathop{\mathbb{E}}_{\epsilon \sim \mathcal{N}(0,I)} \left[p(\mathbf{x} - \sigma\epsilon) \right]. \tag{7.9}$$

From this expression, assuming p is continuous, we clearly see that the gradients are asymptotically independent of σ:

$$\lim_{\sigma \to 0} \nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}). \tag{7.10}$$

Maintaining proportionality $\mathbb{E}\|\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})\|^2 \propto 1/\sigma^2$ requires the gradients to grow unbounded as $\sigma \to 0$, but the gradients of the noiseless distribution $\log p(\mathbf{x})$ are finite. Therefore, proportionality must break down asymptotically and we conclude that–even though we turn the noise σ^2 down to visually imperceptible levels–we have not reached the asymptotic regime.

We conjecture that the proportionality between the gradients and the noise is a consequence of severe non-smoothness in the noiseless model $p(\mathbf{x})$. The probability mass of this distribution is peaked around plausible images $\mathbf{x}$, and decays rapidly away from these points in most directions. Consider the extreme case where the prior has a Dirac delta point mass. The convolution of a Dirac delta with a Gaussian is itself Gaussian so, near the point mass, the noisy distribution p_σ will be proportional to a Gaussian density with variance σ^2. If p_σ were exactly Gaussian then analytically

$$\mathop{\mathbb{E}}_{\mathbf{x} \sim p_\sigma} \left[\|\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})\|^2 \right] = \frac{1}{\sigma^4} \mathop{\mathbb{E}}_{\mathbf{x} \sim p_\sigma} \left[\mathbf{x}^2 \right] = \frac{1}{\sigma^2}. \tag{7.11}$$

Because the distribution $p(\mathbf{x})$ does not contain actual delta spikes–only approximations thereof–we would expect this proportionality to eventually break down as $\sigma \to 0$. Indeed, Figure 7.2 shows that both for NCSN and Glow models of CIFAR-10, after maintaining a very consistent proportionality $\mathbb{E}\left[\|\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})\|^2\right] \propto 1/\sigma^2$ at the higher noise levels, the decay of σ^2 to zero eventually outpaces the growth of the gradients.

7.2.4 Hyper-parameter settings

We adopt the hyper-parameters proposed by Song and Ermon (2019) for annealing σ^2, the proportionality constant δ, and the iteration count T. The noise σ is geometrically annealed from $\sigma_1 = 1.0$ to $\sigma_L = 0.01$ with $L = 10$. We set $\delta = 2 \times 10^{-5}$, and $T = 100$. We find that

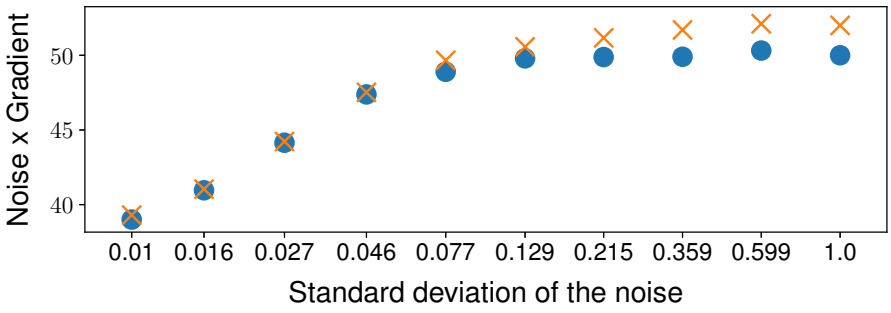


Figure 7.2: The behavior of $\sigma \times \|\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})\|$ in expectation for the NCSN (orange) and Glow (blue) models trained on CIFAR-10 at each of 10 noise levels as σ decays geometrically from 1.0 to 0.01. For large σ, $\|\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})\| \approx 50/\sigma$. This proportional relationship breaks down for smaller σ. Because the expected gradient of the noiseless density $\log p(\mathbf{x})$ is finite, its product with σ must asymptotically approach zero as $\sigma \to 0$.

the same proportionality constant between σ^2 and η also works well for γ^2 and η, allowing us to set $\gamma^2 = \sigma^2$. We use these hyper-parameters for both the NCSN and Glow models, applied to each of the three datasets MNIST, CIFAR-10, and LSUN.

7.2.5 Constructing noise-conditioned models

For noise-conditioned score networks, we can directly compute $\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})$ by evaluating the score network at the desired noise level. For generative flow models like Glow, these noisy distributions are not directly accessible. We could estimate the distributions $p_\sigma(\mathbf{x})$ by training Glow from scratch on datasets perturbed by each of the required noise levels σ^2. But this not practical; Glow is expensive to train, requiring thousands of epochs to converge and consuming hundreds of gpu-hours to obtain good models even for small low-resolution datasets.

Instead of training models $p_\sigma(\mathbf{x})$ from scratch, we apply the concept of fine-tuning from transfer learning (Yosinski et al., 2014). Using pre-trained models of $p(\mathbf{x})$ published by the

Glow authors, we fine-tune these models on noise-perturbed data $\mathbf{x}+\epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. Empirically, this procedure quickly converges to an estimate of $p_\sigma(\mathbf{x})$, within about 10 epochs.

7.2.6 The importance of stochasticity

We remark that adding Gaussian noise to the gradients in the BASIS algorithm is essential. If we set aside the Bayesian perspective, it is tempting to simply run gradient ascent on the pixels of the components to maximize the likelihood of these components under the prior, with a Lagrangian term to enforce the mixture constraint $g(\mathbf{x}) = \mathbf{m}$:

$$\mathbf{x} \leftarrow \mathbf{x} + \eta \nabla_{\mathbf{x}} \left[\log p(\mathbf{x}) - \lambda \| g(\mathbf{x}) - \mathbf{m} \|^2 \right]. \tag{7.12}$$

But this does not work. As demonstrated in Figure 7.3, there are many local optima in the loss surface of $p(\mathbf{x})$ and a greedy ascent procedure simply gets stuck. Pragmatically, the noise term in Langevin dynamics can be seen as a way to knock the greedy optimization Equation (7.12) out of local maxima.

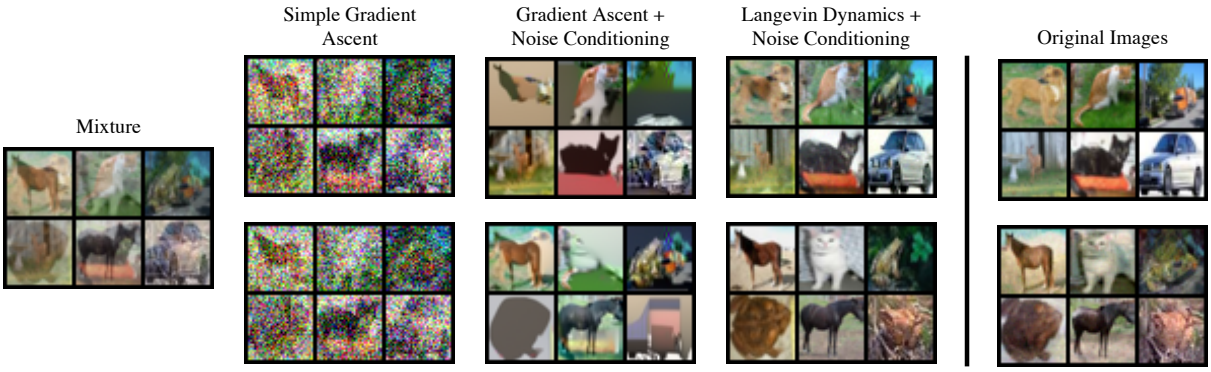


Figure 7.3: Non-stochastic gradient ascent produces sub-par results. Annealing over smoothed-out distributions (Noise Conditioning) guides the optimization towards likely regions of pixel space, but gets stuck at sub-optimal solutions. Adding Gaussian noise to the gradients (Langevin dynamics) shakes the optimization trajectory out of bad local optima.

In the recent literature, pixel-space optimizations by following gradients $\nabla_{\mathbf{x}}$ of some objective are perhaps associated more with adversarial examples than with desirable results (Goodfellow et al., 2015; Nguyen et al., 2015). We note that there have been some successes of pixel-wise optimization in texture synthesis (Gatys et al., 2015) and style transfer (Gatys et al., 2016). But broadly speaking, pixel-space optimization procedures often seem to go wrong. We speculate that noisy optimizations Equation (7.6) on smoothed-out objectives like p_σ could be a widely applicable method for making pixel-space optimizations more robust.

7.3 Evaluation Methodology

Many previous works on source separation evaluate their results using peak signal-to noise ratio (PSNR) or structural similarity index (SSIM) (Wang et al., 2004). These metrics assume that the original sources are identifiable; in probabilistic terms, the true posterior distribution $p(\mathbf{x}|\mathbf{m})$ is presumed to have a unique global maximum achieved by the ground truth sources (up to permutation of the sources). Under the identifiability assumption, it is reasonable to measure the quality of a separation algorithm by comparing separated sources to ground truth mixture components. PSNR, for example, evaluates separations by computing the mean-squared distance between pixel values of the ground truth and separated sources on a logarithmic scale.

For CIFAR-10 source separation, the ground truth source components of a mixture are not identifiable. As evidence for this claim, we call the reader's attention to Figure 7.4. For each mixture depicted in Figure 7.4, we present separation results that sum to the mixture and (to our eyes) look plausibly like CIFAR-10 images. However, in each case the separated images exhibit high deviation from the ground truth. This phenomenon is not unusual; Figure 7.5 shows an un-curated collection of samples from $p(\mathbf{x}|\mathbf{m})$ using BASIS, illustrating a variety of plausible separation results for each given mixture. We will later see evidence again of non-identifiability in Figure 7.7. If we accept that the separations presented in Figures 7.4, 7.5, and 7.7 are reasonable, then source separation on this dataset is fundamentally underdetermined; we cannot measure success using metrics like PSNR that

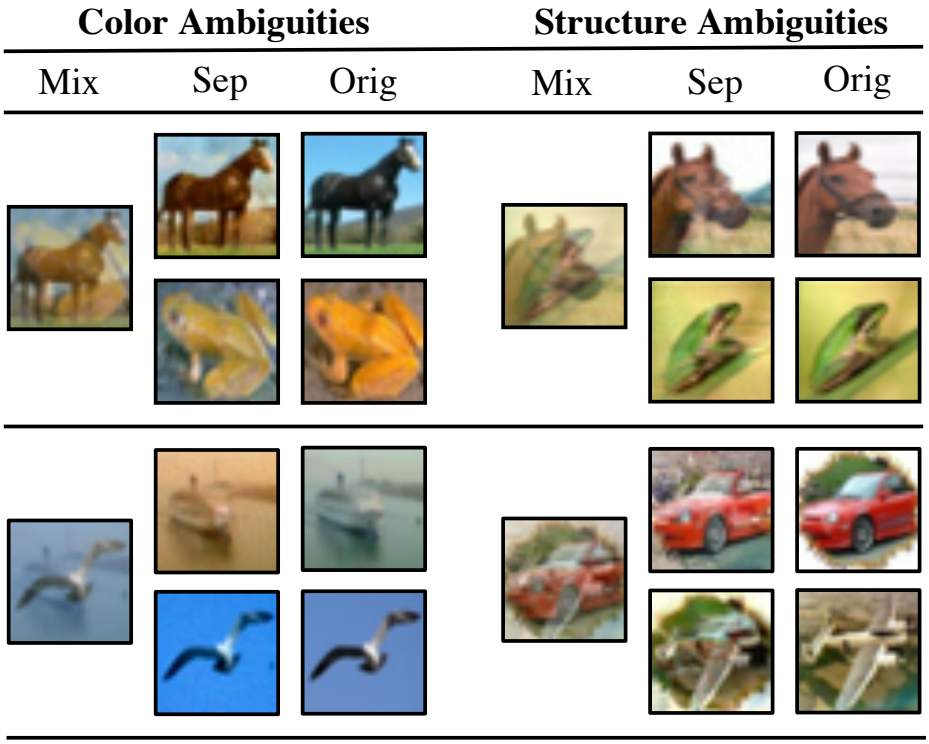


Figure 7.4: A curated collection of examples demonstrating color and structural ambiguities in CIFAR-10 mixtures. In each case, the original components differ substantially from the components separated by BASIS using NCSN as a prior. But in each case, the separation results also look like plausible CIFAR-10 images and exactly sum to the mixture.

compare separation results to ground truth.

Instead of comparing separations to ground truth, we propose instead to quantify the extent to which the results of a source separation algorithm look like samples from the data distribution. If a pair of images sum to the given mixture and look like samples from the data distribution, we deem the separation to be a success. This shift in perspective from identifiability of the latent components to the quality of the separated components is analogous to the classical distinction in the statistical literature between estimation and prediction (Shmueli et al., 2010; Bellec et al., 2018). To this end, we borrow the Inception

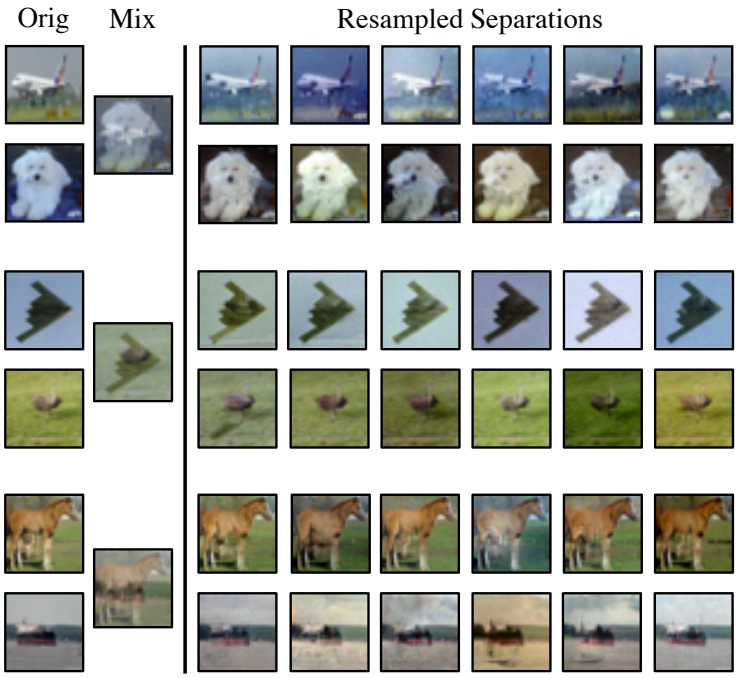


Figure 7.5: Repeated sampling using BASIS with NCSN as a prior for several mixtures of CIFAR-10 images. While most separations look reasonable, variation in color and lighting makes comparative metrics like PSNR unreliable. This challenges the notion that the ground truth components are identifiable.

Score (IS) (Salimans et al., 2016) and Frechet Inception Distance (FID) (Heusel et al., 2017) metrics from the generative modeling literature to evaluate CIFAR-10 separation results. These metrics attempt to quantify the similarity between two distributions given samples. We use them to compare the distribution of components produced by a separation algorithm to the distribution of ground truth images.

In contrast to CIFAR-10, the posterior distribution $p(\mathbf{x}|\mathbf{m})$ for an MNIST model is demonstrably peaked. Moreover, BASIS is able to consistently identify these peaks. This constitutes a constructive proof that components of MNIST mixtures are identifiable, and therefore comparisons to the ground-truth components make sense. We report PSNR results for MNIST, which allows us to compare the results of BASIS to other recent work on MNIST

image separation (Halperin et al., 2018; Kong et al., 2019).

7.4 Experiments

We evaluate results of BASIS on 3 datasets: MNIST (LeCun et al., 1998) CIFAR-10 (Krizhevsky, 2009a) and LSUN (Yu et al., 2015). For MNIST and CIFAR-10, we consider both NCSN (Song and Ermon, 2019) and Glow (Kingma and Dhariwal, 2018b) models as priors, using pre-trained weights published by the authors of these models. For LSUN there is no pre-trained NCSN model, so we consider results only with Glow. For Glow, we fine-tune the weights of the pre-trained models to construct noisy models p_σ using the procedure described in Section 7.2.5. Code and instructions for reproducing these experiments is available online.[1]

Baselines. On MNIST we compare to results reported for the GAN-based "S-D" method (Kong et al., 2019) and the fully supervised version of Neural Egg separation "NES" (Halperin et al., 2018). Results for MNIST are presented in Section 7.4.1. To the best of our knowledge there are no previously reported quantitative metrics for CIFAR-10 separation, so as a baseline we ran Neural Egg separation on CIFAR-10 using the authors' published code. CIFAR-10 results are presented in Section 7.4.2. We present additional qualitative results for 64×64 LSUN in Section 7.4.3, which demonstrate that BASIS scales to larger images.

We also consider results for a simple baseline, "Average," that separates a mixture $\mathbf{m}$ into two 50% masks $\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{m}/2$. This is a surprisingly competitive baseline. Observe that if we had no prior information about the distribution of components, and we measure separation quality by PSNR, then by a symmetry argument setting $\mathbf{x}_1 = \mathbf{x}_2$ is the optimal separation strategy in expectation. In principle we would expect Average to perform very poorly under IS/FID, because these metrics purport to measure similarity of distributions and mixtures should have little or no support under the data distribution. But we find that IS and FID both assign reasonably good scores to Average, presumably because mixtures exhibit many features that are well supported by the data distribution. This speaks to well-

[1] https://github.com/jthickstun/basis-separation

known difficulties in evaluating generative models Theis et al. (2016a) and could explain the strength of "Average" as a baseline.

We remark that we cannot compare our algorithm to the separation-like task reported for CapsuleNets (Sabour et al., 2017). The segmentation task discussed in that work is similar to source separation, but the mixtures used for the segmentation task are constructed using the non-linear threshold function $h(\mathbf{x}) = \max(\mathbf{x}_1 + \mathbf{x}_2, 1)$, in contrast to our linear function g. While extending the techniques of this paper to non-linear relationships between $\mathbf{x}$ and $\mathbf{m}$ is intriguing, we leave this to future this work.

Class conditional separation. The Neural Egg separation algorithm is designed with the assumption that the components $\mathbf{x}_i$ are drawn from different distributions. For quantitative results on MNIST and CIFAR-10, we therefore consider two slightly different tasks. The first is class-agnostic, where we construct mixtures by summing randomly selected images from the test set. The second is class-conditional, where we partition the test set into two groupings: digits $0-4$ and $5-9$ for MNIST, animals and machines for CIFAR-10. The former task allows us compare to S-D results on MNIST, and the latter task allows us to compare to Neural Egg separation on MNIST and CIFAR-10.

There are two different ways to apply a prior for class-conditional separation. First observe that, because $\mathbf{x}_1$ and $\mathbf{x}_2$ are chosen independently,

$$p(\mathbf{x}) = p(\mathbf{x}_1, \mathbf{x}_2) = p_1(\mathbf{x}_1)p_2(\mathbf{x}_2). \tag{7.13}$$

In the class agnostic setting, $\mathbf{x}_1$ and $\mathbf{x}_2$ are drawn from the same distribution (the empirical distribution of the test set) so it makes sense to use a single prior $p = p_1 = p_2$. In the class conditional setting, we could potentially use separate priors over components $\mathbf{x}_1$ and $\mathbf{x}_2$. For the MNIST and CIFAR-10 experiments in this paper, we use pre-trained models trained on unconditional distribution of the training data for both the class agnostic and class conditional setting. It is possible that better results could be achieved in the class conditional setting by re-training the models on class conditional training data. For LSUN, the authors of Glow provide separate pre-trained models for the Church and Bedroom categories, so we

are able to demonstrate class-conditional LSUN separations using distinct priors in Section 7.4.3.

Sample Likelihoods. Although we do not directly model the posterior likelihood $p(\mathbf{x}|\mathbf{m})$, we can compute the log-likelihood of the output samples $\mathbf{x}$. The log-likelihood is a function of the artificial variance hyper-parameter γ, so it is more informative to look at the unweighted square error $\|\mathbf{m} - g(\mathbf{x})\|^2$; this quantity can be interpreted as a reconstruction error, and measures how well we approximate the hard mixture constraint. Because we geometrically anneal the variance γ, by the end of optimization the mixture constraint is rigorously enforced; per-pixel reconstruction error is smaller than the quantization level of 8-bit color, resulting in pixel-perfect visual reconstructions.

For Glow, we can also compute the log-probability of samples under the prior. How do the probabilities of sources $\mathbf{x}_{\text{BASIS}}$ constructed by BASIS separation compare to the probabilities of data $\mathbf{x}_{\text{test}}$ taken directly from a dataset's test set? Because we anneal the noise to a fixed level $\sigma_L > 0$, we find it most informative to ask this question using the minimal-noise, fine-tuned prior $p_{\sigma_L}(\mathbf{x})$. As seen in Table 7.1, the outputs of BASIS separation are generally comparable in log-likelihood to test set images; BASIS separation recovers sources deemed typical by the prior.

Table 7.1: The mean log-likelihood under the minimal-noise Glow prior $p_{\sigma_L}(\mathbf{x})$ for the test set $\mathbf{x}_{\text{test}}$, and for samples of 100 BASIS separations $\mathbf{x}_{\text{BASIS}}$. The log-likelihood of each test set under the noiseless prior $p(\mathbf{x}_{\text{test}})$ is reported for reference.

Dataset	$p(\mathbf{x}_{\text{test}})$	$p_{\sigma_L}(\mathbf{x}_{\text{test}})$	$p_{\sigma_L}(\mathbf{x}_{\text{BASIS}})$
MNIST	0.5	3.6	3.6
CIFAR-10	3.4	4.5	4.7
LSUN (bed)	2.4	4.2	4.4
LSUN (crh)	2.7	4.4	4.4

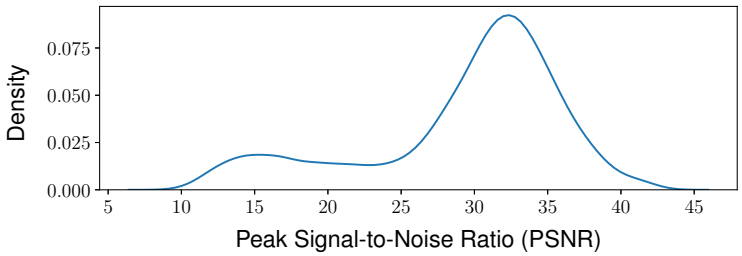


Figure 7.6: The empirical distribution of PSNR for 5,000 class agnostic MNIST digit separations using BASIS with the NCSN prior (see Table 7.2 for comparison of the central tendencies of this and other separation methods).

7.4.1 MNIST separation

Quantitative results for MNIST image separation are reported in Table 7.2, and a panel of visual separation results are presented in Figure 7.1. For quantitative results, we report mean PSNR over separations of $12,000$ separated components. The distribution of PSNR for class agnostic MNIST separation is visualized in Figure 7.6. We observe that approximately $2/3$ of results exceed the mean PSNR of 29.5, which to our eyes is visually indistinguishable from ground truth.

A natural approach to improve separation performance is to sample multiple $\mathbf{x} \sim p(\cdot|\mathbf{m})$ for a given mixture $\mathbf{m}$. A major advantage of models like Glow, that explicitly parameterize the prior $p(\mathbf{x})$, is that we can approximate the maximum of the posterior distribution with the maximum over multiple samples. By construction, samples from BASIS approximately satisfy $g(\mathbf{x}) = \mathbf{m}$, so for the noiseless model we simply declare $p(\mathbf{m}|\mathbf{x}) = 1$ and therefore $p(\mathbf{x}|\mathbf{m}) \propto p(\mathbf{x})$. We demonstrate the effectiveness of resampling in Table 7.2 (Glow, 10x) by comparing the expected PSNR of $\mathbf{x} \sim p(\cdot|\mathbf{m})$ to the expected PSNR of $\arg\max_i p(\mathbf{x}_i)$ over 10 samples $\mathbf{x}_1, \ldots, \mathbf{x}_{10} \sim p(\cdot|\mathbf{m})$. Even moderate resampling dramatically improves separation performance. Unfortunately this approach cannot be applied to the otherwise superior NCSN model, which does not model explicit likelihoods $p(\mathbf{x})$.

Table 7.2: PSNR results for separating 6,000 pairs of equally mixed MNIST images. For class split results, one image comes from label $0-4$ and the other comes from $5-9$. We compare to S-D Kong et al. (2019), NES Halperin et al. (2018), convolutional NMF (class split) Halperin et al. (2018) and standard NMF (class agnostic) Kong et al. (2019).

Algorithm	Class Split	Class Agnostic
Average	14.8	14.9
NMF	16.0	9.4
S-D	-	18.5
BASIS (Glow)	22.9	22.7
NES	24.3	-
BASIS (Glow, 10x)	27.7	27.1
BASIS (NCSN)	**29.5**	**29.3**

Without any modification, we can apply BASIS to separate mixtures of $k > 2$ images. We contrast this with regression-based methods, which require re-training to target varying numbers of components. Figure 7.1 shows the results of BASIS using the NCSN prior applied to mixtures of four randomly selected images. For more mixture components, we observe that identifiability of ground truth sources begins to break down. This is illustrated by looking at the central item in each panel of Figure 7.1 (highlighted in orange).

7.4.2 CIFAR-10

Quantitative results for CIFAR-10 image separation measured are presented in Table 8.2, and visual separation results are presented in Figure 7.1.

We can also view image colorization Levin et al. (2004); Zhang et al. (2016) as a source separation problem by interpreting a grayscale image as a mixture of the three color channels

Table 7.3: Inception Score / FID Score of 25,000 separations (50,000 separated images) of two overlapping CIFAR-10 images using NCSN as a prior. In Class Split one image comes from the category of animals and other from the category of vehicles. NES results using published code from Halperin et al. (2018).

Algorithm	Inception Score	FID
Class Split		
NES	5.29 ± 0.08	51.39
BASIS (Glow)	5.74 ± 0.05	40.21
Average	6.14 ± 0.11	39.49
BASIS (NCSN)	**7.83 ± 0.15**	**29.92**
Class Agnostic		
BASIS (Glow)	6.10 ± 0.07	37.09
Average	7.18 ± 0.08	28.02
BASIS (NCSN)	**8.29 ± 0.16**	**22.12**

of an image $\mathbf{x} = (\mathbf{x}_r, \mathbf{x}_g, \mathbf{x}_b)$ with

$$g(\mathbf{x}) = (\mathbf{x}_r + \mathbf{x}_g + \mathbf{x}_b)/3. \tag{7.14}$$

Unlike our previous separation problems, the channels of an image are clearly not independent, and the factorization of p given by Equation 7.13 is unwarranted. But conveniently, a generative model trained on color CIFAR-10 images itself models the joint distribution $p(\mathbf{x}) = p(\mathbf{x}_r, \mathbf{x}_g, \mathbf{x}_b)$. Therefore, the same pre-trained generative model that we use to separate images can also be used to color them.

Qualitative colorization results are visualized in Figure 7.7. The non-identifiability of ground truth is profound for this task (see Section 7.3 for discussion of identifiability). We draw attention to the two cars in the middle of the panel: the white car that is colored

yellow by the algorithm, and the blue car that is colored red. The colors of these specific cars cannot be inferred from a greyscale image; the best an algorithm can do is to choose a reasonable color, based on prior information about the colors of cars.

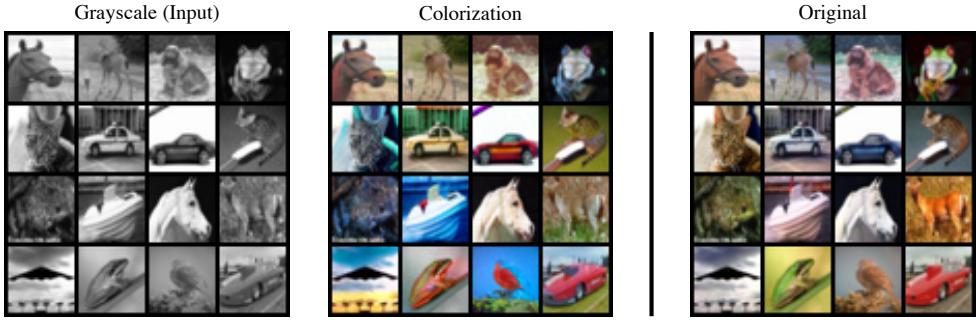


Figure 7.7: Colorizing CIFAR-10 images. Left: original CIFAR-10 images. Middle: greyscale conversions of the images on the left. Right: imputed colors for the greyscale images, found by BASIS using NCSN as a prior.

Table 7.4: Inception Score / FID Score of 50,000 colorized CIFAR-10 images. As measured by IS/FID, the quality of NCSN colorizations nearly matches CIFAR-10 itself.

Data Distribution	Inception Score	FID Score
Input Grayscale	8.01 ± 0.10	68.52
BASIS (Glow)	8.69 ± 0.15	28.70
BASIS (NCSN)	**10.53 ± 0.17**	**11.58**
CIFAR-10 Original	11.24 ± 0.12	0.00

Quantitative coloring results for CIFAR-10 are presented in Table 7.4. We remark that the IS and FID scores for coloring are substantially better than the IS and FID scores of 8.87 and 25.32 respectively reported for unconditional samples from the NCSN model; conditioning on a greyscale image is enormously informative. Indeed, the Inception Score of NCSN-colorized CIFAR-10 is close to the Inception Score of the CIFAR-10 dataset itself.

7.4.3 LSUN separation

Qualitative results for LSUN separations are visualized in Figure 7.8. While the separation results in Figure 7.8 are imperfect, Table 7.1 shows that the mean log-likelihood of the separated components is comparable to the mean log-likelihood that the model assigns to images in the test set. This suggests that the model is incapable of distinguishing these separations from better results, and the imperfections are attributable to the quality of the model rather than to the separation algorithm. This is encouraging, because it suggests that the artifacts are due to the Glow model rather than the BASIS separation algorithm, and that better separation results will be achievable with improved generative models.

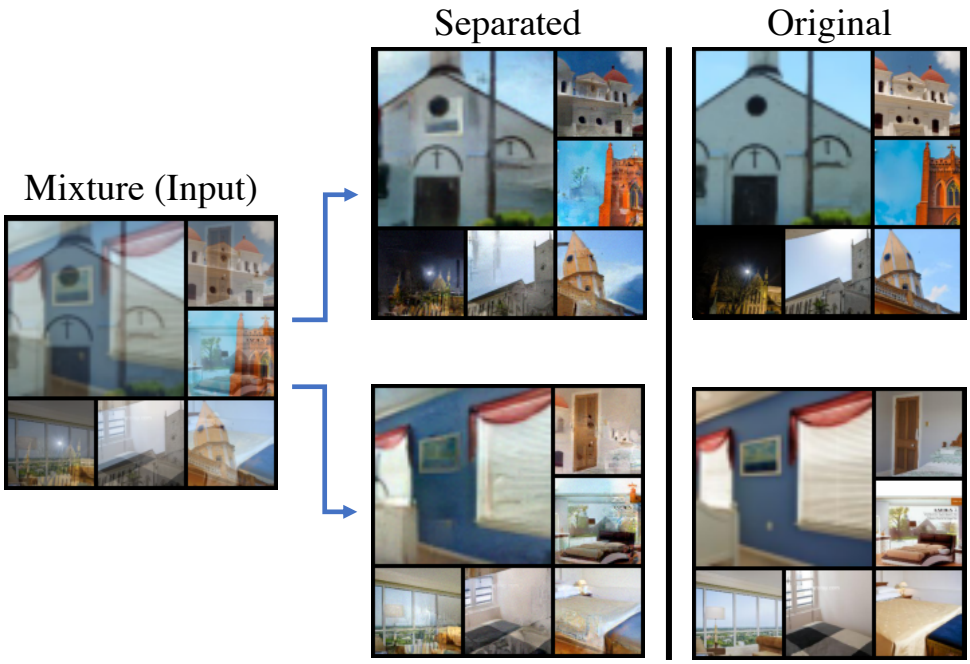


Figure 7.8: 64×64 LSUN separation results using Glow as a prior. One mixture component is sampled from the LSUN churches category, and the other component is sampled from LSUN bedrooms.

7.5 Conclusion

In this chapter, we introduced a new approach to source separation that makes use of a likelihood-based generative model as a prior. We demonstrated the ability to swap in different generative models for this purpose, presenting results of our algorithm using both NCSN and Glow. We proposed new methodology for evaluating source separation on richer datasets, demonstrating strong performance on MNIST and CIFAR-10. Finally, we presented qualitative results on LSUN that point the way towards scaling this method to practical tasks such as speech separation, using generative audio models like WaveNets Oord et al. (2016).

Chapter 8

PARALLEL AND FLEXIBLE SAMPLING FROM AUTOREGRESSIVE MODELS VIA LANGEVIN DYNAMICS

In this chapter, we extend the BASIS algorithm from Chapter 7 to solve linear inverse problems with autoregressive models. Furthermore, we show that we can use the BASIS method of sampling to speed up unconditional generation autoregressive models by parallelizing inference. Neural autoregressive models (Larochelle and Murray, 2011) are a popular family of generative models, with wide-ranging applications in a variety of domains including audio (Oord et al., 2016; Dhariwal et al., 2020a), images (van den Oord et al., 2016b; Salimans et al., 2017; Parmar et al., 2018; Razavi et al., 2019), and text (Radford et al., 2019; Brown et al., 2020b).

These models parameterize the conditional distribution over a token in an ordered sequence, given previous tokens in the sequence. The standard approach to sampling from an autoregressive model iteratively generates tokens, according to a conditional distribution over tokens defined by the model, conditioned on the partial sequence of previously generated tokens. We will refer to this approach to sampling as the ancestral sampler.

There are two major drawbacks to ancestral sampling that limit the usefulness of autoregressive models in practical settings. First, ancestral sampling has time complexity that scales linearly in the length of the generated sequence. For data such as high-resolution images or audio, ancestral sampling from an autoregressive model (where the tokens are pixels or sound pressure readings respectively) can be impractically slow. Second, ancestral sampling is frustratingly inflexible. It is easy to sample the second half of a sequence conditioned on the first, but filling in the first half a sequence conditioned on the second naively requires training a new model that reverses the ordering of tokens in the autoregressive factorization.

Conditioning on arbitrary subsets of tokens for tasks such as inpainting or super-resolution seems beyond reach of autoregressive modeling.

This paper introduces an alternative, parallel and flexible (PnF) sampler for autoregressive models that can be parallelized and steered using conditioning information or constraints. Instead of sampling tokens sequentially, the PnF sampler initializes a complete sequence (with random tokens) and proceeds to increase the log-likelihood of this sequence by following a Markov chain defined by Langevin dynamics (Neal et al., 2011) on a smoothed log-likelihood. The smoothing temperature is cooled over time according to an annealing schedule informed by Song and Ermon (2019, 2020). Convergence time of this annealed Langevin dynamics is empirically independent of the sequence length and, generalizing the method in Chapter 7, the PnF sampler can be applied to posterior log-likelihoods to incorporate conditional information into the sampling process. We highly encourage the reader to view the audio examples at `https://grail.cs.washington.edu/projects/pnf-sampling/`.

The primary technical contribution of this paper is the development of the PnF sampler for discretized autoregressive models (Section 8.2.1). Our interest in these models is motivated by their success as unconditional models of audio waves (Oord et al., 2016; Mehri et al., 2017; Dhariwal et al., 2020a). Defined over a discrete lattice within a continuous space, these models occupy a middle ground between continuous and discrete models. For continuous models such as RNADE (Uria et al., 2013), PnF sampling can be directly applied as in Song and Ermon (2019); Jayaram and Thickstun (2020). We defer the development of the PnF sampler for fully discrete models to future work.

In Section 8.2.2, we present a stochastic variant of the PnF sampler based on stochastic gradient Langevin dynamics (Welling and Teh, 2011). This is an embarrassingly parallel, asynchronous distributed algorithm for autoregressive sampling. Using a WaveNet model, we show in Section 8.3.3 that stochastic PnF sampling approximates the quality of ancestral sampling to arbitrary accuracy, with compute time that is inverse proportional to the number of computing devices. This allows PnF sampling to take full advantage of modern, massively parallel computing infrastructure.

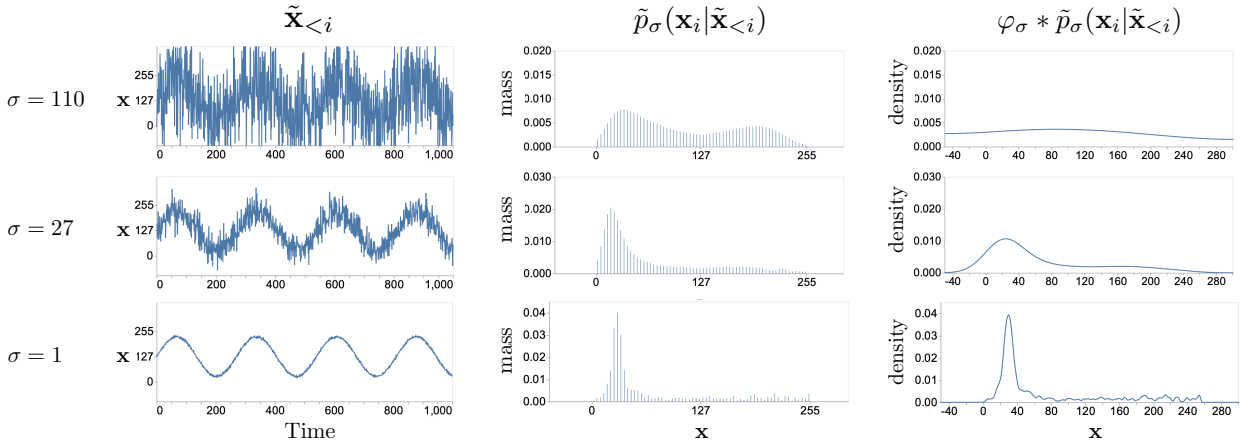


Figure 8.1: A visual summary of discretized autoregressive smoothing. Given a noisy history $\tilde{\mathbf{x}}_{<i} = \mathbf{x}_{<i} + \boldsymbol{\varepsilon}_{<i}$ (left column) where $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 I)$, we train a model to predict the un-noised distribution over $\mathbf{x}_i \in \mathbb{R}$ (middle column). This distribution is discrete and non-differentiable in $\tilde{\mathbf{x}}$; we convolve with a Gaussian $\varphi_\sigma(t) = \mathcal{N}(t; 0, \sigma^2)$ to produce a continuous estimate of $\tilde{\mathbf{x}}_i$ (right column). We can run Langevin dynamics on the continuous distribution, and gradually anneal the smoothing to approximate the target distribution.

We will see in Section 8.2.3 how the PnF sampler can find solutions to general posterior sampling problems, using an unconditional generative model as a prior. In Section 8.3 we present applications of the PnF sampler to a variety of Bayesian image and audio inverse problems. We focus on linear inverse problems, using PixelCNN++ (Salimans et al., 2017) and WaveNet (Oord et al., 2016) models as priors. Sections 8.3.4, 8.3.5, and 8.3.6 demonstrate PnF conditional sampling for source separation, super-resolution, inpainting respectively. PnF sampling results correlate strongly with the strength of the generative model used as a prior; as better autoregressive models are developed, they can be used with PnF sampling to improve performance on conditional generation tasks. We refer the reader to the project website for demonstrations of audio PnF sampling: https://grail.cs.washington.edu/projects/pnf-sampling/.

8.1 Related Work

The PnF autoregressive sampler is based on the annealed Langevin dynamics introduced in Song and Ermon (2019), which accelerates standard Langevin dynamics (Neal et al., 2011; Du and Mordatch, 2019) using a smoothing procedure in the spirit of simulated annealing (Kirkpatrick et al., 1983) and graduated optimization (Blake and Zisserman, 1987). The extension of annealed Langevin dynamics to conditional sampling problems was discussed in Jayaram and Thickstun (2020) for source separation and image coloring problems, and developed further in Song et al. (2021) for general posterior sampling problems. The present work extends these methods to discretized autoregressive models, for which the smoothing procedures described in previous work are not directly applicable (Frank and Ilse, 2020). Markov-chain Monte Carlo posterior samplers based on Gibbs sampling rather than Langevin dynamics are proposed in Theis and Bethge (2015) and Hadjeres et al. (2017) as solutions for inpainting tasks.

The slow speed of ancestral sampling is a persistent obstacle to the adoption and deployment of autoregressive models. This has inspired algorithms that seek to parallelize the sampling process. Parallel WaveNet (van den Oord et al., 2018a) and ClariNet (Ping et al., 2019) train generative flow models to mimic the behavior of an autoregressive model. Sampling a flow model requires only one pass through a feed-forward network and can be distributed across multiple devices. Wiggers and Hoogeboom (2020) and Song et al. (2020c) propose fixed-point algorithms that, like PnF sampling, iteratively refine an initial sample from a simple distribution into a sample from the target distribution. But none of these methods are easily adaptable to source separation (Section 8.3.4) or more general conditional sampling tasks.

Like anytime sampling (Xu et al., 2021), PnF sampling offers a tradeoff between sample quality and computational budget. The algorithm's iterates gradually mix to the target distribution and, by stopping early, we can approximate samples from this distribution using less computation. We explore the empirical tradeoff between sample quality and computation

for PnF sampling from autoregressive models in Section 8.3.2. The anytime sampler proposed in Xu et al. (2021) requires a specific model architecture based on the VQ-VAE (van den Oord et al., 2017b; Razavi et al., 2019). In contrast, the PnF sampler can be used with any likelihood-based model. Unlike an anytime sampler, the computational budget for PnF sampling must be specified in advance: halting prior to completing the annealing schedule will result in noisy samples.

Bayesian inverse problems are explored extensively in theoretical settings, where the prior is given by a simple analytical distribution (Tropp and Wright, 2010b; Knapik et al., 2011; Wang et al., 2017). These problems have also been studied using learned priors given by GAN's, with a focus on linear inverse problems (Rick Chang et al., 2017; Bora et al., 2017; Raj et al., 2019). These GAN-based approaches are tailored to the latent variable architecture of the model, performing latent space optimizations to find codes that correspond to desired outputs. There is no obvious extension of these latent variable approaches to autoregressive models.

While we focus on autoregressive models, due to their strong empirical performance as unconditional models of audio, PnF sampling could be applied more generally with other likelihood-based models. In the audio space, this includes recent diffusion models (Kong et al., 2021a; Chen et al., 2021). Note however that audio vocoder models (Prenger et al., 2019; Kim et al., 2018; Ping et al., 2020a), which rely on spectrogram conditioning, cannot be adapted as priors for the source separation, super-resolution, and inpainting experiments presented in Section 8.3. In addition, GAN based models (Donahue et al., 2019; Kumar et al., 2019), which are not likelihood based, cannot be sampled using PnF.

8.2 Parallel and Flexible Sampling

We want to sample from an autoregressive generative model over some indexed sequence of values $\mathbf{x} \in \mathcal{X}^n$ where

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(\mathbf{x}_i | \mathbf{x}_{<i}). \tag{8.1}$$

We are particularly interested in developing a sampler for discretized autoregressive models, where $\mathcal{X} = \mathbb{R}$ and each conditional $p(\mathbf{x}_i|\mathbf{x}_{<i})$ has support on a finite set of scalar values $\mathcal{D} = \{e_1, \ldots, e_d\} \subset \mathbb{R}$. The set $\mathcal{D}^n$ could represent, for example, an 8-bit encoding of an image or audio wave.

We propose to sample from $p(\mathbf{x})$ via Langevin dynamics. Let $\mathbf{x}_0 \sim \text{Uniform}(\mathbb{R}^n)$, $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, I_n)$, and define a Markov chain

$$\mathbf{x}^{(t+1)} \equiv \mathbf{x}^{(t)} + \eta \nabla_{\mathbf{x}} \log p(\mathbf{x}^{(t)}) + \sqrt{2\eta}\boldsymbol{\varepsilon}_t \tag{8.2}$$

If $p(\mathbf{x})$ were a smooth density then, for sufficiently small η, the Markov chain mixes and $\mathbf{x}^{(t)}$ converges in distribution to p as $t \to \infty$. But a discretized probability distribution defined over $\mathcal{D}^n$ is not smooth; the gradient $\nabla_{\mathbf{x}} \log p(\mathbf{x}^{(t)})$ is not even well-defined. In Section 8.2.1 we propose a smoothing of the discrete model $p(\mathbf{x})$, creating a differentiable density on which the Markov chain Equation (8.2) can mix.

To support conditional generation, we then turn our attention to sampling from the posterior of a joint distribution $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$, where $p(\mathbf{x})$ is an autoregressive model over $\mathbf{x} \in \mathbb{R}^n$ and $p(\mathbf{y}|\mathbf{x})$ is a conditional likelihood. Langevin dynamics for sampling from the posterior $p(\mathbf{x}|\mathbf{y})$ are given by

$$\begin{aligned} \mathbf{x}^{(t+1)} &\equiv \mathbf{x}^{(t)} + \eta \nabla_{\mathbf{x}} \log p(\mathbf{x}^{(t)}|\mathbf{y}) + \sqrt{2\eta}\boldsymbol{\varepsilon}_t \\ &= \mathbf{x}^{(t)} + \eta \nabla_{\mathbf{x}} \left(\log p(\mathbf{x}^{(t)}) + \log p(\mathbf{y}|\mathbf{x}^{(t)}) \right) + \sqrt{2\eta}\boldsymbol{\varepsilon}_t. \end{aligned} \tag{8.3}$$

This is a convenient Markov chain for posterior sampling, because the partition function $p(\mathbf{y})$ vanishes when we take the gradient. However, like $p(\mathbf{x})$, the posterior $p(\mathbf{x}|\mathbf{y})$ is not smooth. In Section 8.2.3 we propose a smoothing of the joint distribution $p(\mathbf{x}, \mathbf{y})$, for which the posterior $p(\mathbf{x}|\mathbf{y})$ is differentiable and the Markov chain Equation (8.3) can mix.

Given a smoothing procedure parameterized by a temperature parameter, we appeal to the simulated annealing heuristic developed in Song and Ermon (2019) to turn down the temperature as the Markov chain Equation (8.2) or Equation (8.3) mixes. In contrast to classical Markov chain sampling, for which samples $\mathbf{x}^{(t)}$ converge in distribution to p, annealed

Algorithm 4 Parallel and Flexible Sampling

Input: $\mathbf{y}$, $\{\sigma_i\}_{i=1}^{L}$, δ, T
Sample $\mathbf{x}^{(0)} \sim \mathcal{N}(0, \sigma_1^2 I_n)$
for $i \leftarrow 1$ to L **do**
    $\eta_i \leftarrow \delta \cdot \sigma_i^2 / \sigma_L^2$
    for $t = 1$ to T **do**
        Sample $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, I_n)$
        $\mathbf{g}^{(t)} \leftarrow \nabla_{\mathbf{x}} \log p_{\sigma_i}(\mathbf{x}^{(t)}) + \nabla_{\mathbf{x}} \log p_{\sigma_i}(\mathbf{y}|\mathbf{x}^{(t)})$
        $\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} + \eta_i \mathbf{g}^{(t)} + \sqrt{2\eta}\boldsymbol{\varepsilon}_t$
    end for
end for

Langevin dynamics converges asymptotically to a single point distributed approximately according to p. Algorithm 4 describes these annealed Langevin dynamics given a smoothed prior $p_\sigma(\mathbf{x})$ and smoothed likelihood $p_\sigma(\mathbf{y}|\mathbf{x})$. The structure of this algorithm is the same as the annealed Langevin dynamics presented in Song and Ermon (2019) (the unconditional case where $\nabla_{\mathbf{x}} p_\sigma(\mathbf{y}|\mathbf{x}) = 0$) and in Chapter 7 (the conditional case); the novel contribution of this paper is the smoothing algorithm for evaluating $\nabla_{\mathbf{x}} p_\sigma(\mathbf{x})$ given $p(\mathbf{x})$ (Section 8.2.1) and $\nabla_{\mathbf{x}} p_\sigma(\mathbf{y}|\mathbf{x})$ given $p(\mathbf{y}|\mathbf{x})$ (Section 8.2.3).

Each step of the Langevin dynamics described in Equations Equation (8.2) or Equation (8.3) requires inference of $\log p(\mathbf{x})$, an $O(n)$ operation. Unlike sequential sampling, calculating $\log p(\mathbf{x})$ for a given sequence $\mathbf{x} \in \mathbb{R}^n$ is embarrassingly parallel; for moderate sequence lengths n, the cost of computing $\log p(\mathbf{x})$ is essentially constant using a modern parallel computing device. When n is very large, e.g. for WaveNet models where just a minute of audio has $n > 10^6$ samples, it can be convenient to distribute sampling across multiple computing devices. In Section 8.2.2 we describe a stochastic variant of PnF sampling based on stochastic gradient Langevin dynamics that is easily distributed across a cluster of devices.

8.2.1 Autoregressive Smoothing

We consider models that parameterize the conditional distribution $p(\mathbf{x}_i|\mathbf{x}_{<i})$ with a categorical softmax distribution over $d = |\mathcal{D}|$ discrete values; given functions $f_i : \mathcal{X}^i \to \mathbb{R}^d$, we define

$$p(\mathbf{x}_i = e_k|\mathbf{x}_{<i}) = \frac{\exp(f_{i,k}(\mathbf{x}_{<i}))}{\sum_{\ell=1}^{d} \exp(f_{i,\ell}(\mathbf{x}_{<i}))}. \tag{8.4}$$

The functions f_i are typically given by a neural network, with shared weights across the sequential indices i. Collectively, these conditional models define the joint distribution $p(\mathbf{x})$ according to Equation Equation (8.1).

We cannot directly compute gradients of the distribution $p(\mathbf{x})$ defined by discrete conditionals $p(\mathbf{x}_i|\mathbf{x}_{<i})$. Instead, we smooth $p(\mathbf{x})$ by convolving it with a spherical Gaussian $\mathcal{N}(0, \sigma^2 I_n)$. This smoothing relies on the fact that $e_k \in \mathcal{D}$ represent scalar values on the real line, and therefore the discrete distribution $p(\mathbf{x}_i|\mathbf{x}_{<i})$ can be viewed as a linear combination of weighted Dirac spikes on $\mathcal{D} \subset \mathbb{R}$. If $\phi_\sigma(\mathbf{x})$ denotes the density of a spherical Gaussian $\mathcal{N}(0, \sigma^2 I_n)$ on $\mathbb{R}^n$ then we define a density $p_\sigma(\tilde{\mathbf{x}})$ on $\mathbb{R}^n$ given by

$$p_\sigma(\tilde{\mathbf{x}}) = (\phi_\sigma * p)(\tilde{\mathbf{x}}) = \int \phi_\sigma(\tilde{\mathbf{x}} - \mathbf{x}) p(\mathbf{x})\, d\mathbf{x}. \tag{8.5}$$

This distribution has well-defined gradients $\nabla_{\tilde{\mathbf{x}}} \log p_\sigma(\tilde{\mathbf{x}})$ and $p_\sigma(\tilde{\mathbf{x}}) \to p(\mathbf{x})$ in total variation as $\sigma^2 \to 0$.

When $p(\mathbf{x})$ is a deep autoregressive model, the convolution Equation (8.5) is difficult to calculate directly. In the previous chapter, we proposed training a smoothed model $p_\sigma(\tilde{\mathbf{x}})$ by fine-tuning a model $p(\mathbf{x})$ on noisy data $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\varepsilon}_\sigma$ where $\mathbf{x} \sim p$ and $\boldsymbol{\varepsilon}_\sigma \sim \mathcal{N}(0, \sigma^2 I_n)$. This approach cannot be directly applied to discrete autoregressive models as defined by Equation Equation (8.4). The obstruction is that noisy samples $\tilde{\mathbf{x}}_i \in \mathbb{R}$ are not supported by the discretization $\mathcal{D}$.

One way to address the problem is to replace the discrete model $p(\mathbf{x})$ with a continuous autoregressive model of $p_\sigma(\tilde{\mathbf{x}})$, e.g. RNADE (Uria et al., 2013). We avoid this approach because fine-tuning $p(\mathbf{x})$ to $p_\sigma(\tilde{\mathbf{x}})$ becomes complicated when these models have different architectures.

Instead of directly fine-tuning $p(\mathbf{x})$ to a model $p_\sigma(\tilde{\mathbf{x}})$, we combine an analytic calculation with an auxiliary model learned via fine-tuning. Let $\tilde{p}_\sigma(\mathbf{x}_i|\tilde{\mathbf{x}}_{<i})$ denote a (discrete) conditional model trained to predict $\mathbf{x}_i$ given noisy covariates $\tilde{\mathbf{x}}_{<i} = \mathbf{x}_{<i} + \boldsymbol{\varepsilon}_{\sigma,<i}$. If φ_σ denotes the density of $\mathcal{N}(0, \sigma^2)$ then we can re-write the factored density $p_\sigma(\tilde{\mathbf{x}})$ as

$$p_\sigma(\tilde{\mathbf{x}}) = \prod_{i=1}^{n} p_\sigma(\tilde{\mathbf{x}}_i|\tilde{\mathbf{x}}_{<i}) = \prod_{i=1}^{n} \big(\varphi_\sigma * \tilde{p}_\sigma(\,\cdot\,|\tilde{\mathbf{x}}_{<i})\big)(\tilde{\mathbf{x}}_i). \tag{8.6}$$

On the right-hand side, we decompose the smoothed conditional densities $p_\sigma(\tilde{\mathbf{x}}_i|\tilde{\mathbf{x}}_{<i})$ into Gaussian convolutions of discrete conditionals $\tilde{p}_\sigma(\,\cdot\,|\tilde{\mathbf{x}}_{<i})$ evaluated at $\tilde{\mathbf{x}}_i$. This suggests the following approach to evaluating $p_\sigma(\tilde{\mathbf{x}}_i|\tilde{\mathbf{x}}_{<i})$:

- Learn a (discrete) model $\tilde{p}_\sigma(\mathbf{x}_i|\tilde{\mathbf{x}}_{<i})$, trained to predict the un-noised value $\mathbf{x}_i$ given noisy history $\tilde{\mathbf{x}}_{<i}$. This model can be learned efficiently by finetuning a pre-trained model $p(\mathbf{x}_i|\mathbf{x}_{<i})$ on noisy covariates $\tilde{\mathbf{x}}_{<i}$. This is visualized in the middle column of Figure 8.1.

- Evaluate the Gaussian convolution $\varphi_\sigma * \tilde{p}_\sigma(\,\cdot\,|\tilde{\mathbf{x}}_{<i})$ at $\tilde{\mathbf{x}}_i$ to compute $p_\sigma(\tilde{\mathbf{x}}_i|\tilde{\mathbf{x}}_{<i})$. This convolution can be calculated in closed form given $\tilde{p}_\sigma(\mathbf{x}_i|\tilde{\mathbf{x}}_{<i})$. This is visualized in the right column of Figure 8.1.

The convolution $p_\sigma(\tilde{\mathbf{x}}_i|\tilde{\mathbf{x}}_{<i}) = \big(\varphi_\sigma * \tilde{p}_\sigma(\,\cdot\,|\tilde{\mathbf{x}}_{<i})\big)(\tilde{\mathbf{x}}_i)$ has a simple closed form given by a Gaussian mixture model

$$p_\sigma(\tilde{\mathbf{x}}_i|\tilde{\mathbf{x}}_{<i}) = \sum_{k=1}^{d} \tilde{p}_\sigma(e_k|\tilde{\mathbf{x}}_{<i})\varphi_\sigma(\tilde{\mathbf{x}}_i - e_k). \tag{8.7}$$

Using the softmax parameterization of $\tilde{p}(\mathbf{x}_i|\mathbf{x}_{<i})$ given by Equation Equation (8.4), with fine-tuned logits $f_{\sigma,i} : \mathcal{X}^{\otimes i} \to \mathbb{R}^d$, the log-density of this smoothed conditional density can be written in a numerically stable form:

$$\begin{aligned} \log\, p_\sigma(\mathbf{x}_i|\mathbf{x}_{<i}) &= -\log \sum_{\ell=1}^{d} \exp(f_{\sigma,i,\ell}(\mathbf{x}_{<i})) \\ &+ \log \sum_{k=1}^{d} \exp\left(f_{\sigma,i,k}(\mathbf{x}_{<i}) - \frac{1}{2\sigma^2}(\mathbf{x}_i - e_k)^2\right) + C. \end{aligned} \tag{8.8}$$

This smoothing procedure requires us to store finetuned models $\tilde{p}_\sigma(\mathbf{x}_i|\tilde{\mathbf{x}}_{<i})$ for each of the L noise levels $\sigma \in \{\sigma_1, \ldots, \sigma_L\}$. This could be avoided by training a single noise-conditioned generative model. The advantage of finetuning is that we can directly use standard models, without any adjustment to the network architecture or subsequent hyper-parameter tuning of the modified architecture; this cleanly decouples our approach to conditional sampling from neural architecture design questions. Note that while we store L copies of the model, there is no additional memory overhead: these models are loaded and unloaded serially during optimization as we anneal the noise levels, so only one model is resident in memory at a time. While memory is a scarce resource, disk space is generally abundant.

8.2.2 Stochastic Gradient Langevin Dynamics

Calculating the Langevin updates described in Equations Equation (8.2) and Equation (8.3) requires $O(n)$ operations to compute $\log p_\sigma(\mathbf{x})$, given a sequence $\mathbf{x} \in \mathbb{R}^n$. This calculation decomposes into n calculations of $\log p_\sigma(\mathbf{x}_i|\mathbf{x}_{<i})$ $(i = 1, \ldots, n)$ which can each be computed in parallel (this is what allows autoregressive models to be efficiently trained using the maximum likelihood objective). But for very large values of n, even modern parallel computing devices cannot fully parallelize all n calculations. In this section, we develop a stochastic variant of PnF sampling (Algorithm 5) that is easily distributed across multiple devices.

Instead of making batch updates on a full sequence $\mathbf{x} \in \mathbb{R}^n$, consider updating a single coordinate $j \in \{1, \ldots, n\}$:

$$\mathbf{x}_j^{(t+1)} = \mathbf{x}_j^{(t)} + \eta \nabla_{\mathbf{x}_j} \log p_\sigma(\mathbf{x}^{(t)}) + \sqrt{2\eta}\boldsymbol{\varepsilon}_j^{(t)}. \tag{8.9}$$

This coordinate-wise derivative is only dependent on the tail of the sequence $\mathbf{x}_{\geq j}$:

$$\nabla_{\mathbf{x}_j} \log p_\sigma(\mathbf{x}) = \sum_{i=j}^{n} \nabla_{\mathbf{x}_j} \log p_\sigma(\mathbf{x}_i|\mathbf{x}_{<i}). \tag{8.10}$$

This doesn't look promising; calculating an update on a single coordinate $\mathbf{x}_j$ required $n - j$ inference calculations $p_\sigma(\mathbf{x}_i|\mathbf{x}_{<i})$. But models over very long sequences, including WaveNets,

usually make a Markov assumption $p(\mathbf{x}_i|\mathbf{x}_{<i}) = p(\mathbf{x}_i|\mathbf{x}_{i-w}, \ldots, \mathbf{x}_{i-1})$ for some limited contextual window of length w. In this case, the coordinate-wise derivative requires only w calls:

$$\nabla_{\mathbf{x}_j} \log p_\sigma(\mathbf{x}) = \sum_{i=j}^{j+w} \nabla_{\mathbf{x}_j} \log p_\sigma(\mathbf{x}_i|\mathbf{x}_{<i}). \tag{8.11}$$

Calculating a gradient on a contiguous block of c coordinates leads to a more efficient update

$$\nabla_{\mathbf{x}_{j:j+c}} \log p_\sigma(\mathbf{x}) = \sum_{i=j}^{j+c+w} \nabla_{\mathbf{x}_{j:j+c}} \log p_\sigma(\mathbf{x}_i|\mathbf{x}_{<i}). \tag{8.12}$$

Calculating Equation Equation (8.12) requires transmission of a block $\{\mathbf{x}_{j-w}, \ldots, \mathbf{x}_{j+c+w}\}$ of length $c+2w$ to the computing device, and $c+w$ calculations $p_\sigma(\mathbf{x}_i|\mathbf{x}_{<i})$ in order to compute the gradient of a block of length c. If we partition a sequence of length n into n/c blocks of length c, then we can distribute computation of $\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})$ with an overhead factor of $1 + w/c$. This motivates choosing c as large as possible, under the constraint that $c + w$ calculations $p_\sigma(\mathbf{x}_i|\mathbf{x}_{<i})$ can still be parallelized on a single device.

We can calculate $\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})$ by aggregating n/c blocks of gradients according to Equation Equation (8.12), requiring synchronous communication between n/c machines for every update Equation Equation (8.2); this is a MapReduce algorithm (Dean and Ghemawat, 2004). We propose a bolder approach in Algorithm 5 based on block-stochastic Langevin dynamics (Welling and Teh, 2011). If $j \in \{1, \ldots, n\}$ is chosen uniformly at random then Equation Equation (8.12) is an unbiased estimate of $\nabla_{\mathbf{x}} \log p_\sigma(\mathbf{x})$. This motivates block-stochastic updates on patches, which multiple devices can perform asynchronously, a Langevin analog to Hogwild! (Niu et al., 2011).

8.2.3 Smoothing a Joint Distribution

We now consider joint distributions $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$ over sources $\mathbf{x} \in \mathcal{X}^n$ and measurements $\mathbf{y} \in \mathcal{Y}^n$. For example, $\mathbf{y}$ could be a low resolution version of the signal, or an observed mixture of two signals. We are particularly interested in measurement models of the form

Algorithm 5 Stochastic Parallel and Flexible Sampling

Input: $\mathbf{y}$, $\{\sigma_i\}_{i=1}^L$, δ, T
Sample $\mathbf{x} \sim \mathcal{N}(0, \sigma_1^2 I_n)$
for $i \leftarrow 1$ to L **do**
    $\eta_i \leftarrow \delta \cdot \sigma_i^2 / \sigma_L^2$
    Fork()
    for $t = 1$ to T **do**
        Sample $j \sim \text{Uniform}\{1, \ldots, n\}$
        Sample $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, I_c)$
        Read $\mathbf{x}_{j-w:j+c+w}^{(t)} \leftarrow \mathbf{x}_{j-w:j+c+w}$
        $\mathbf{g}_j^{(t)} \leftarrow \nabla_{\mathbf{x}_{j:j+c}} \log p_{\sigma_i}(\mathbf{x}_{j-w:j+c+w}^{(t)})$
            $+ \nabla_{\mathbf{x}_{j:j+c}} \log p_{\sigma_i}(\mathbf{y}_{j:j+c}^{(t)} | \mathbf{x}_{j-w:j+c+w}^{(t)})$
        Write $\mathbf{x}_{j:j+c} \leftarrow \mathbf{x}_{j:j+c}^{(t)} + \eta_i \mathbf{g}_j^{(t)} + \sqrt{2\eta}\boldsymbol{\varepsilon}_t$
    end for
    Synchronize()
end for

$\mathbf{y} = g(\mathbf{x})$, for some linear function $g : \mathcal{X}^n \to \mathcal{Y}^n$. We can view these measurements $\mathbf{y}$ as degenerate likelihoods $p(\mathbf{y}|\mathbf{x})$ of the form

$$p(\mathbf{y}|\mathbf{x}) = \delta(\mathbf{y} - g(\mathbf{x})), \tag{8.13}$$

where δ denotes the Dirac delta function. This family of linear measurement models describes the source separation, in-painting, and super-resolution tasks featured in Section 8.3, as well as other linear inverse problems including sparse recovery and image colorization.

Extending our analysis in Section 8.2.1, we can smooth the joint density $p(\mathbf{x}, \mathbf{y})$ by convolving $\mathbf{x}$ with a spherical Gaussian $\mathcal{N}(0, \sigma^2 I_n)$. Let $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\varepsilon}_\sigma$ where $\boldsymbol{\varepsilon}_\sigma \sim \mathcal{N}(0, \sigma^2 I_n)$. Note that $\tilde{\mathbf{x}}$ is conditionally independent of $\mathbf{y}$ given $\mathbf{x}$ and therefore the joint distribution

over $\mathbf{x}$, $\tilde{\mathbf{x}}$, and $\mathbf{y}$ can be factored as

$$p_\sigma(\mathbf{x}, \mathbf{y}, \tilde{\mathbf{x}}) = p_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p(\mathbf{y}|\mathbf{x})p(\mathbf{x}). \tag{8.14}$$

We will work with the smoothed marginal $p_\sigma(\tilde{\mathbf{x}}, \mathbf{y})$ of the joint distribution $p_\sigma(\mathbf{x}, \mathbf{y}, \tilde{\mathbf{x}})$. If $\phi_\sigma(\mathbf{x})$ denotes the density of a spherical Gaussian $\mathcal{N}(0, \sigma^2 I_n)$ on $\mathbb{R}^n$ then the marginal $p_\sigma(\tilde{\mathbf{x}}, \mathbf{y})$ of Equation Equation (8.14) can be expressed by

$$p_\sigma(\tilde{\mathbf{x}}, \mathbf{y}) = \int \phi_\sigma(\tilde{\mathbf{x}} - \mathbf{x})p(\mathbf{x}, \mathbf{y})\, d\mathbf{x} \tag{8.15}$$

This density approximates the original distribution $p(\mathbf{x}, \mathbf{y})$ in the sense that $p_\sigma(\tilde{\mathbf{x}}, \mathbf{y}) \to p(\mathbf{x}, \mathbf{y})$ in total variation as $\sigma^2 \to 0$.

The smoothed density $p_\sigma(\tilde{\mathbf{x}}, \mathbf{y})$ can be factored as $p_\sigma(\mathbf{y}|\tilde{\mathbf{x}})p_\sigma(\tilde{\mathbf{x}})$. The density $p_\sigma(\tilde{\mathbf{x}})$ is simply the smoothed Gaussian convolution of $p(\mathbf{x})$ given by Equation Equation (8.5).

For general likelihoods $p(\mathbf{y}|\mathbf{x})$, because $\mathbf{y}$ is conditionally independent of $\tilde{\mathbf{x}}$ given $\mathbf{x}$, we can write the density $p_\sigma(\mathbf{y}|\tilde{\mathbf{x}})$ by marginalizing over $\mathbf{x}$ as

$$p_\sigma(\mathbf{y}|\tilde{\mathbf{x}}) = \int p_\sigma(\mathbf{y}|\tilde{\mathbf{x}}, \mathbf{x})p_\sigma(\mathbf{x}|\tilde{\mathbf{x}})\, d\mathbf{x} \tag{8.16}$$

$$= \int p(\mathbf{y}|\mathbf{x})\phi_\sigma(\mathbf{x} - \tilde{\mathbf{x}})\, d\mathbf{x}. \tag{8.17}$$

This integral is difficult to calculate directly. One way to evaluate $p_\sigma(\mathbf{y}|\tilde{\mathbf{x}})$ is to take the same approach described in Section 8.2.1 to evaluate $p_\sigma(\tilde{\mathbf{x}})$: fine-tune the model $p(\mathbf{y}|\mathbf{x})$ to a model $\tilde{p}(\mathbf{y}|\tilde{\mathbf{x}})$ that predicts $\mathbf{y}$ given noisy covariates $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\varepsilon}_\sigma$. A similar procedure (using a noise-conditioned architecture rather than finetuning) is described in Section 5 of Song et al. (2021).

For linear measurement models with the form given by Equation Equation (8.13), the smoothed density $p_\sigma(\mathbf{y}|\tilde{\mathbf{x}})$ can be calculated in closed form. Writing the linear function $g : \mathcal{X}^n \to \mathcal{Y}^n$ as a matrix $g(\mathbf{x}) = A\mathbf{x}$, we have

$$\mathbf{y} = g(\mathbf{x}) = g(\tilde{\mathbf{x}}) + g(-\boldsymbol{\varepsilon}_\sigma) \sim \mathcal{N}\left(g(\tilde{\mathbf{x}}), \sigma^2 AA^T\right). \tag{8.18}$$

This smoothing given by Equation Equation (8.18) generalizes the smoothing proposed in Chapter 7 for source separation. That work proposed separately smoothing the prior and

likelihood, resulting in a smoothed likelihood $p(\tilde{\mathbf{y}}|\mathbf{x})$ over $\tilde{\mathbf{y}} = \mathbf{y} + \boldsymbol{\varepsilon}_{\mathbf{y}}$, where $\boldsymbol{\varepsilon}_{\mathbf{y}} \sim \mathcal{N}(0, \sigma^2 I_n)$. This is equivalent to Equation Equation (8.18) in the case of source separation, for which $g(\mathbf{x}) = \frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2$ and therefore $g(\mathbf{x}) \sim \mathcal{N}\left(g(\tilde{\mathbf{x}}), \sigma^2 I\right)$.

For general likelihoods $p(\mathbf{y}|\mathbf{x})$ (e.g. a classifier) the conditioning values $\mathbf{y}$ may depend on the whole sequence $\mathbf{x}$. In this case, stochastic PnF must read the entire sequence $\mathbf{x}$ in order to calculate the posterior

$$\nabla_{\mathbf{x}_{j:j+c}} \log p(\mathbf{x}|\mathbf{y}) = \nabla_{\mathbf{x}_{j:j+c}} \big(\log p(\mathbf{x}) + \log p(\mathbf{y}|\mathbf{x}) \big). \tag{8.19}$$

But for long sequences $\mathbf{x}$ such as audio, the conditioning information $\mathbf{y}$ is often a local function of the sequence $\mathbf{x}$. In this case, $\mathbf{x} \in \mathcal{X}^n$, $\mathbf{y} \in \mathcal{Y}^n$, $\mathbf{y}_i = g(\mathbf{x}_{N(i)})$ where $N(i)$ is a local neighborhood of indices near i, and the likelihood decomposes via conditional independence into

$$\log p(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{n} \log p(\mathbf{y}_i|\mathbf{x}_{N(i)}). \tag{8.20}$$

All experiments presented in Section 8.3 feature this conditioning pattern. For spectrogram conditioning, $N(i)$ is the set of indices (centered at i) required to compute a short-time Fourier transform. For source separation, super-resolution, and in-painting, $N(i) = i$. This allows us to compute block gradients of the conditional likelihood (Algorithm 5).

8.3 Experiments

We present qualitative and quantitive results of PnF sampling for WaveNet models of audio (Oord et al., 2016) and a PixelCNN++ model of images (Salimans et al., 2017). In Section 8.3.2 we show that PnF sampling can produce samples of comparable quality to ancestral sampling. In Section 8.3.3 we show that stochastic PnF sampling is faster than ancestral sampling, when parallelized across a modest number of devices. We go on to demonstrate how PnF sampling can be applied to a variety of image and audio restoration tasks: source separation (Section 8.3.4), super-resolution (Section 8.3.5), and inpainting (Section 8.3.6). We encourage the reader to browse the supplementary material for qualitative examples of PnF audio sampling.

8.3.1 Datasets

For audio experiments we use the VCTK dataset (Veaux et al., 2016a) consisting of 44 hours of speech, as well as the Supra Piano dataset (Shi et al., 2019) consisting of 52 hours of piano recordings. We use a random 80-20 train-test split of VCTK speakers and piano recordings for evaluation. Audio sequences are sampled at a 22kHz, with 8-bit μ-law encoding (CCITT, 1988), except for source separation where 8-bit linear encoding is used. Sequences used for quantitative evaluation are 50k sample excerpts, approximately 2.3 seconds of audio, chosen randomly from the longer test set recordings. For image experiments we use the CIFAR-10 dataset (Krizhevsky, 2009b) with the standard train-test split. Additional training and hyperparameter details can be found in the appendix.

8.3.2 Quality of Generated Samples

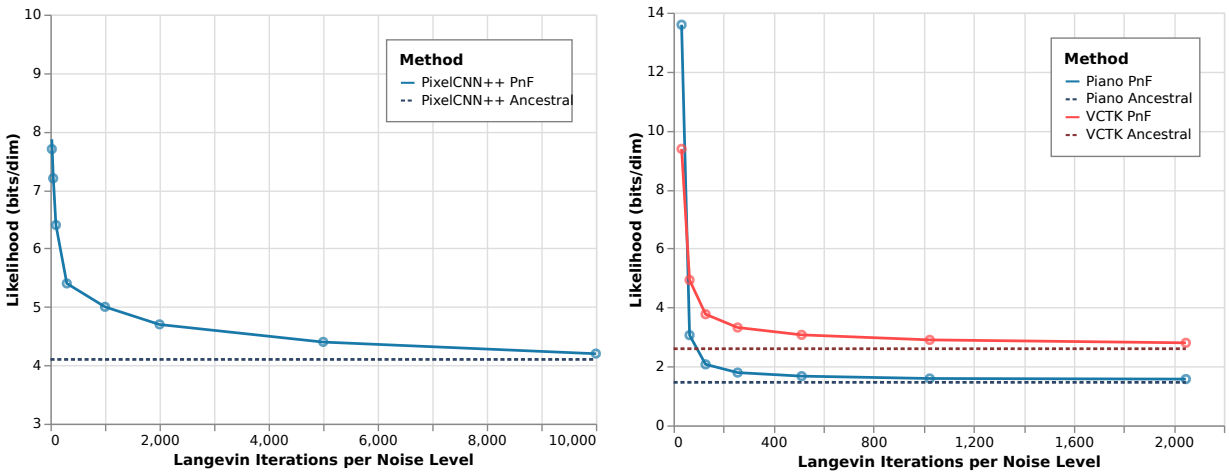


Figure 8.2: As the number of Langevin iterations T increases, the log-likelihood of sequences generated by PnF sampling approaches the log-likelihood of test set sequences. Left: sampling from a PixelCNN++ model trained on CIFAR-10. Right: sampling from WaveNet models trained on the Supra Piano and VCTK speech datasets.

To evaluate the quality of samples generated by PnF sampling, we follow a similar pro-

cedure to Holtzman et al. (2020). We compare log-likelihoods, calculated using the noiseless model $p(\mathbf{x})$, of sequences generated by PnF sampling to sequences generated by ancestral sampling from the lowest-noise model $p_{\sigma_L}(\mathbf{x})$. Because PnF-sampled sequences are continuous, we quantize these samples to 8-bit values when evaluating their likelihood under the noiseless model. We consider PnF sampling to be successful if it generates sequences with comparable log-likelihoods to ancestral generations.

In Figure 8.2 we present quantitative results for PnF sampling using an unconditional PixelCNN++ model of CIFAR-10, and a spectrogram-conditioned WaveNet model of both voice and piano datasets. We evaluate $1,000$ generations (length $n = 50,000$ sequences for the WaveNet models) using various numbers of Langevin iterations, and report the median log likelihood of quantizations of these sequences under the noiseless model. Asymptotically, as the iterations T of Langevin dynamics increase, the likelihood of PnF samples approaches the likelihood of ancestral samples. Audio PnF samples for various T are presented on the project website.

8.3.3 Speed and Parallelism

Ancestral sampling has $O(n)$ serial runtime in the length n of the generated sequence. Using the stochastic PnF sampler described in Section 8.2.2, the serial runtime is $O(T)$, where T is the number of Langevin iterations at each level of smoothing. We find empirically that we can set T independent of n, so in principle the serial runtime of stochastic PnF is constant as a function of sequence length. In practice, we do not have an infinite supply of parallel devices, so the serial runtime of stochastic PnF grows inversely proportional to the number of devices. This behavior is demonstrated in Figure 8.3 for spectrogram-conditioned WaveNet stochastic PnF sampling using a cluster of 8 Nvidia Titan Xp GPU's and $T = 256$. Each GPU can calculate Equation Equation (8.12) for a block of $c = 50,000$ samples (2.3 seconds of audio). For PixelCNN++, we find that $T > n$ and therefore the PnF sampler does not improve sampling speed for this model.

Stochastic PnF sampling depends upon asynchronous writes being sparse so that memory

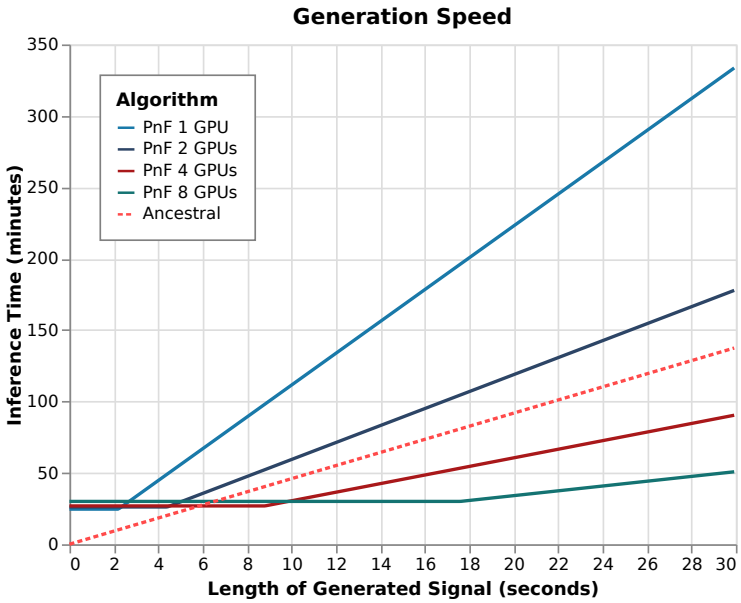


Figure 8.3: PnF sampling can be parallelized across multiple devices, resulting in faster inference time than ancestral sampling. Beyond a threshold level of computation, PnF sampling time is inversely proportional to the number of devices.

overwrites, when two workers update overlapping blocks, are rare. This situation is analogous to the sparse update condition required for Hogwild! If blocks are length c and the number of devices is substantially less than n/c, then updates are sufficiently sparse. But if the number of devices is larger than n/c, memory overwrites become common, and stochastic PnF sampling fails to converge. This imposes a floor on generation time determined by c, exhibited in Figure 8.3. We cannot substantially reduce this floor by decreasing c because of the tradeoff between c and the model's Markov window w described in Section 8.2.2.

In general, PnF sampling becomes faster than ancestral sampling for long sequences n, in which case the stochastic variant of PnF sampling becomes necessary in order to distribute the calculation of n conditional likelihoods. For shorter sequences, accurate unconditional samples can be produced more quickly with the ancestral sampler. Unconditional 32×32

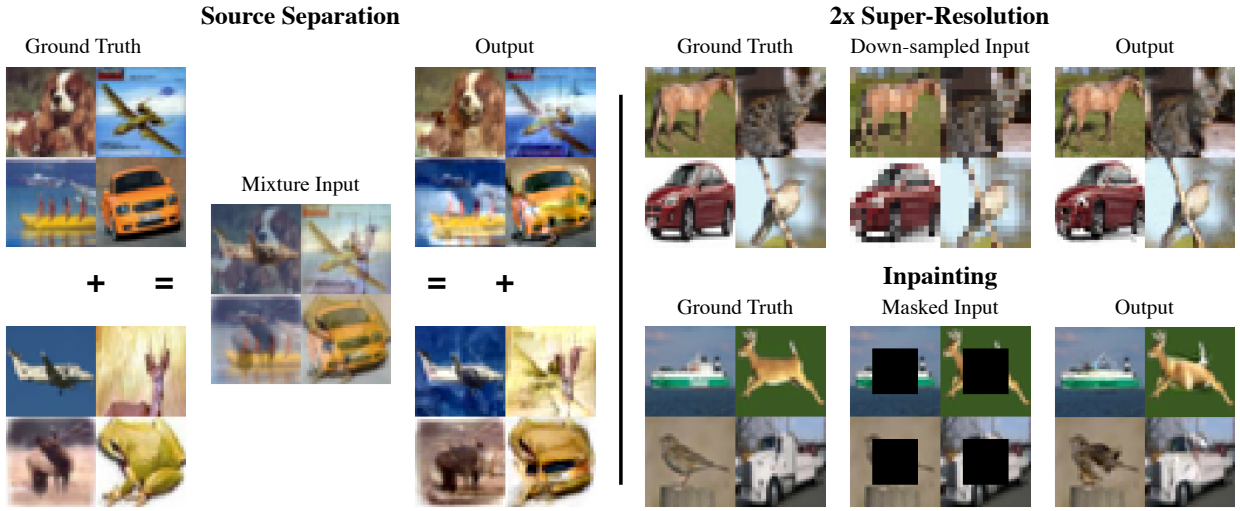


Figure 8.4: PnF sampling applied to visual source separation (Section 8.3.4) super-resolution (Section 8.3.5) and inpainting (Section 8.3.6) using a PixelCNN++ prior over images trained on CIFAR-10. Ground-truth images in this figure are taken from the CIFAR-10 test set.

CIFAR-10 generation using the PixelCNN++ mode requires $T = 10,000$ Langevin iterations per noise level (Figure 8.2) for accurate samples; annealing through $L = 20$ levels requires a total of $L \times T = 200,000$ serial queries to the PixelCNN++ model, far more than $n = 3 \times 32 \times 32$ serial queries to PixelCNN++ for ancestral sampling. Note also that PixelCNN++ conditions on a full image ($w = n$) so the stochastic variant of Pnf sampling is not applicable to this model.

8.3.4 Source Separation

The single-channel source separation problem (Davies and James, 2007) asks us to recover unobserved sources $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}^n = \mathbb{R}^{2 \times n}$ given an observed mixture $\mathbf{y} = g(\mathbf{x}) = \mathbf{x}_1 + \mathbf{x}_2$. Like in Chapter 7, we view source separation as a linear Bayesian inverse problem: recover $(\mathbf{x}_1, \mathbf{x}_2)$ given $\mathbf{y}$ and a prior $p(\mathbf{x}_1, \mathbf{x}_2) = p(\mathbf{x}_1)p(\mathbf{x}_2)$. We consider three variants of this task: (1) audio separation of mixtures of voice (VCTK test set) and piano (Supra Piano test set), (2) visual

separation of mixtures of CIFAR-10 test set "animal" images and "machine" images, and (3) class-agnostic visual separation of mixtures of CIFAR-10 test set images.

Table 8.1: Quantitative results for audio source separation of mixtures of Supra piano and VCTK voice samples. Results are measured using SI-SDR (higher is better).

Algorithm	Test SI-SDR (dB)		
	All	Piano	Voice
PnF (WaveNet)	17.07	13.92	20.25
Conv-Tasnet	17.48	20.02	15.50
Demucs	14.18	16.67	12.75

We compare PnF audio separation to results using the Demucs (Défossez et al., 2019b) and Conv-Tasnet (Luo and Mesgarani, 2019) source separation models. Both Demucs and Conv-Tasnet are supervised models, trained specifically for the source separation task, that learn to output source components given an input mixture. An advantage of PnF sampling is that it does not rely on pairs of source signals and mixes like these supervised methods. We train the supervised models on 10K mixtures of VCTK and Supra Piano samples and measure results on 1K test set mixtures using the standard Scale Invariant Signal-to-Distortion Ratio (SI-SDR) metric for audio source separation (Le Roux et al., 2019). Results in Table 8.1 show that PnF sampling is competitive with these specialized source separation models. Qualitative comparisons are provided in the supplement. We do not compare results on the popular MusDB dataset (Rafii et al., 2017) because this dataset has insufficient single-channel audio to train WaveNet generative models.

For CIFAR-10, we follow the experimental methodology described in Chapter 7. Table 8.2 shows that PixelCNN++ performs comparably to Glow as a prior, but underperforms NCSN. This makes sense, given the relative strength of NCSN as a prior over CIFAR-10 images in comparison to PixelCNN++ and Glow. Given the strong correlation between

Table 8.2: Quantitative results for visual sources separation on CIFAR-10. Results are measured using Inception Score / FID Score of 25,000 separations (50,000 separated images) of two overlapping CIFAR-10 images. In Class Split one image comes from the category of animals and other from the category of machines. NES results (Halperin et al., 2019) and BASIS results are as reported in Chapter 7.

Algorithm	Inception Score	FID
Class Split		
NES	5.29 ± 0.08	51.39
BASIS (Glow)	5.74 ± 0.05	40.21
PnF (PixelCNN++)	5.86 ± 0.07	40.66
Average	6.14 ± 0.11	39.49
BASIS (NCSN)	7.83 ± 0.15	29.92
Class Agnostic		
BASIS (Glow)	6.10 ± 0.07	37.09
PnF (PixelCNN++)	6.14 ± 0.15	37.89
Average	7.18 ± 0.08	28.02
BASIS (NCSN)	8.29 ± 0.16	22.12

the quality of a generative model and the quality of separations using that model as a prior, we anticipate that more recent innovations in autoregressive image models based on transformers (Parmar et al., 2018; Child et al., 2019) will lead to stronger separation results once implementations of these models that match the results reported in these papers become public. Select qualitative image separation results are presented in Figure 8.4.

8.3.5 Super-Resolution

The super-resolution problem asks us to recover unobserved data $\mathbf{x}$ given a down-sampled observation $\mathbf{y} = g(\mathbf{x})$. For 1-dimensional (audio) super-resolution, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^{n/r}$, and $\mathbf{y}_i = \mathbf{x}_{ri}$ (Kuleshov et al., 2017; Eskimez et al., 2019). For 2-dimensional (image) super-resolution $\mathbf{x} \in \mathbb{R}^n = \mathbb{R}^{k \times k \times 3}$, $\mathbf{y} \in \mathbb{R}^{k/r \times k/r \times 3}$, and $\mathbf{y}_{i,j} = \mathbf{x}_{ri,rj}$ (Dahl et al., 2017; Zhang et al., 2018c). Like source separation, super-resolution can be viewed as a Bayesian linear inverse problem, and we can recover solutions to this problem via PnF sampling. In the audio domain, the down-sampling operation $g(\mathbf{x})$ can be interpreted as a low-pass filter.

Table 8.3: Quantitative results for audio super-resolution at three different scales on the Supra piano and VCTK voice datasets. Results are measured using PSNR (higher is better). KEE refers to the method described in Kuleshov et al. (2017)

	Piano			Voice		
Ratio	Spline	KEE	PnF	Spline	KEE	PnF
4x	23.07	22.25	29.78	15.8	16.04	15.47
8x	13.58	15.79	23.49	10.7	11.15	10.03
16x	7.09	6.76	14.23	6.4	7.11	5.32

We measure audio super-resolution performance using peak signal-to-noise ratio (PSNR) and compare against a deep learning baseline (Kuleshov et al., 2017) as well as a simple cubic B-spline. Quantitative audio results are presented in Table 8.3, which show that we outperform these baselines on piano data and produce similar quality reconstructions on voice data. Qualitative audio samples are available in the supplement, where we also show examples of 32x super resolution—beyond the reported ability of existing methods. Select qualitative visual results are presented in Figure 8.4.

8.3.6 Inpainting

Inpainting problems involve the recovery of unobserved data $\mathbf{x}$ given a masked observation $g(\mathbf{x}) = \mathbf{m} \odot \mathbf{x}$, where $\mathbf{m} \in \{0, 1\}^n$ (Adler et al., 2011; Pathak et al., 2016). This family of problems includes completion tasks (finishing a sequence given a prime) pre-completion tasks (generating a prefix to a sequence) and outpainting tasks. Ancestral sampling can only be applied to completion tasks, whereas PnF sampling can be used to fill in any pattern of masked occlusions. Qualitative results for audio inpainting are available in the supplement. Select qualitative results for image inpainting are presented in Figure 8.4.

8.3.7 Audio Qualitative Examples

We showcase qualitative results on various audio tasks on our webste: https://grail.cs.washington.edu/projects/pnf-sampling/. Some of these tasks are shown visually in Figure 8.5.

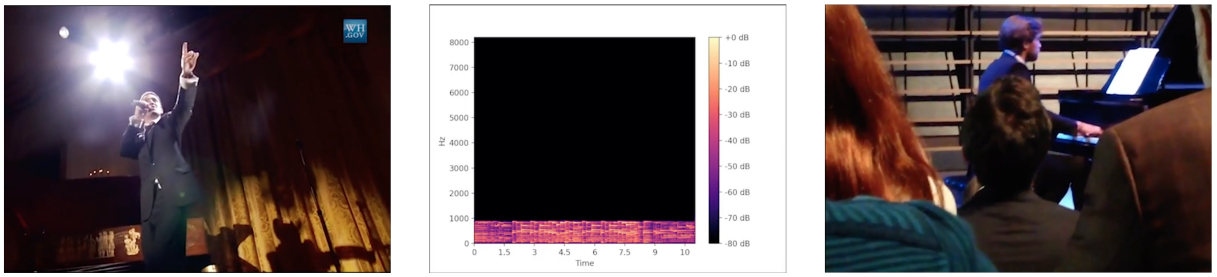


Figure 8.5: We present several in-the-wild audio experiments on our project website. These include musical instrument separation on the Hamilton soundtrack, audio super-resolution, and background sound removal during a piano concert.

8.4 Conclusion

In this chapter we introduced PnF sampling, a parallelizable approach to sampling from autoregressive models that can be flexibly adapted to conditional sampling tasks. The flexibility of PnF sampling decouples the (unconditional) generative modeling problem from the details of specific conditional sampling tasks. Using WaveNet models, we demonstrated a reduction in wall-clock sampling time using PnF sampling in comparison to ancestral sampling, as well as PnF's ability to solve a variety of practical audio processing problems: source separation, super-resolution, and inpainting. We anticipate that PnF conditional sampling results will improve as developments in generative modeling empower us to incorporate stronger models as priors. More broadly, we are inspired by ongoing research in autoregressive generative modeling that, coupled with PnF sampling, will continue to drive performance improvements for practical conditional image and audio restoration tasks.

Chapter 9

CONSTRAINED DIFFUSION IMPLICIT MODELS FOR NOISY INVERSE PROBLEMS

We now turn our attention to diffusion models (Ho et al., 2020), and present a novel method for solving noisy linear inverse problems with constrained diffusion sampling. Recovering an unknown signal from noisy linear measurements is a fundamental challenge encompassing tasks like super-resolution, inpainting, and denoising. For these problems, we assume that we have a partial observation $\mathbf{y}$ that was generated through a linear measurement operator $\mathbf{A}$ applied to some original signal $\mathbf{x}$. In the generalized noisy case, the observed $\mathbf{y}$ is corrupted with measurement noise sampled i.i.d. from a residual distribution r: $\boldsymbol{\sigma} \sim r^{\otimes d}$.

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\sigma} \tag{9.1}$$

While diffusion models have emerged as powerful tools for modeling complex data distributions, adapting them to inverse problems has posed several key challenges. First, existing methods introduce significant computational overhead during inference in order to guide a diffusion method towards a linear constraint. Second, existing methods do not provide guarantees on exact signal recovery while preserving accelerated sampling techniques.

We propose constrained diffusion implicit models (CDIM) to address these challenges. CDIM makes an algorithmic modification to DDIM (Song et al., 2020a) in order to constrain the generated image with guarantees about measurement consistency. The key insight is to to project DDIM updates to satisfy measurement constraints either via KL divergence minimization or L^2 optimization with early stopping.

Our approach differs fundamentally from previous canonical work like Diffusion Posterior Sampling (DPS) (Chung et al., 2022b) and other works in that family like DSG (Yang

et al., 2024). DPS alternates between unconstrained updates and measurement projection, using a soft optimization to guide outputs towards constraints. In contrast, CDIM uses a hard optimization for both noisy and noiseless inverse problems, resulting in faster sampling and exact recovery. We present comparisons to show that simply using DPS with DDIM accelerated sampling does not yield satisfactory results.

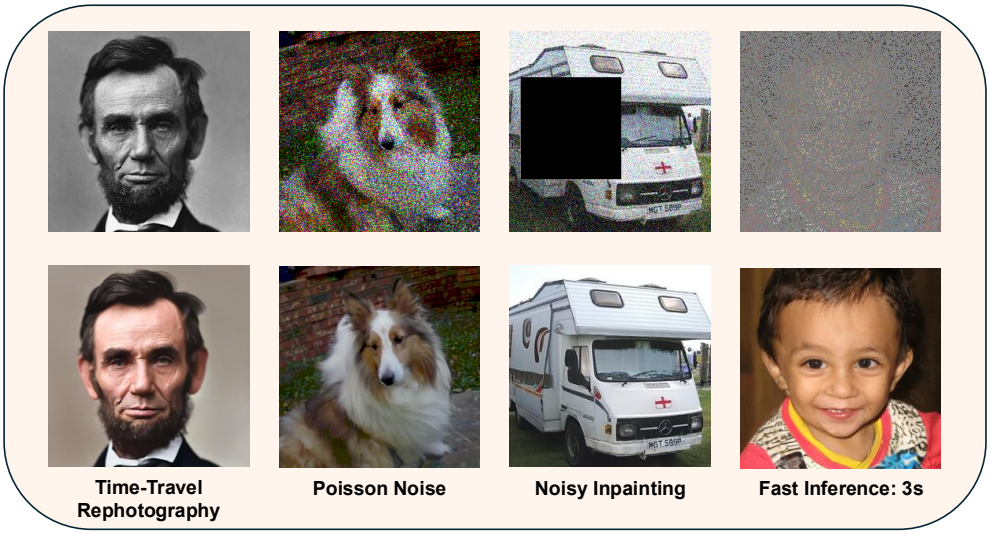


Figure 9.1: We show several applications of CDIM including image colorization, denoising, inpainting, and sparse recovery. We highlight the fact that we can handle general noise distributions, such as Poisson noise, and that our method runs in as little as 3 seconds.

In the noiseless case, CDIM exactly recovers noiseless measurements, even for out-of-distribution inputs (impossible with DPS), while requiring 10-50x fewer network evaluations. For noisy measurements, we generalize CDIM by optimizing the Kullback-Leibler divergence between empirical and expected residual distributions. By formulating noise handling through distributional matching rather than soft constraints, CDIM can accelerate sampling even with measurement noise. We further show that in the presence of Gaussian measurement noise, KL divergence optimization is equivalent to L^2 optimization with early stopping, the latter which converges faster while still matching the residual distribution to

the first two moments. However, the KL divergence method has the advantage of extending beyond Gaussian measurement noise to handle arbitrary noise distributions.

Our contributions are as follows:

- Accelerated inference under linear constraints: we accelerate inference, reducing the number of model evaluations and wall-clock time by an order of magnitude—10 to 50 times faster than previous posterior diffusion methods—while maintaining comparable quality.
- Exact recovery of noiseless observations: we can find solutions $\hat{\mathbf{x}}$ are consistent with the noiseless observation, i.e. $\mathbf{A}\hat{\mathbf{x}} = \mathbf{y}$ to arbitrary precision.
- General noise models: we extend the CDIM framework to accommodate arbitrary observational noise distributions through distributional divergence minimization, demonstrating effectiveness given non-Gaussian noise, such as Poisson noise and bimodal noise.

9.1 Related Work

Diffusion models (Ho et al., 2020) have emerged as powerful generative models, building upon early work in nonequilibrium thermodynamics (Sohl-Dickstein et al., 2015) and implicit models (Mohamed and Lakshminarayanan, 2017). Denoising Diffusion Implicit Models (DDIM) (Song et al., 2020a) was a notable work that improved the efficiency of diffusion sampling through non-markovian sampling. This was further advanced through stochastic differential equations (Song et al., 2021) and numerical ODE solvers like PNDM (Liu et al., 2021).

Applying diffusion models to inverse problems has been an active research area. DPS (Chung et al., 2022b) was a notable method that uses alternating projection steps to guide the diffusion process. DDNM (Wang et al., 2022), DDRM (Kawar et al., 2022), SNIPS (Kawar et al., 2021), and PiGDM (Song et al., 2023a) use linear algebraic approaches and singular value decompositions. Techniques such as DMPS (Meng and Kabashima, 2022),

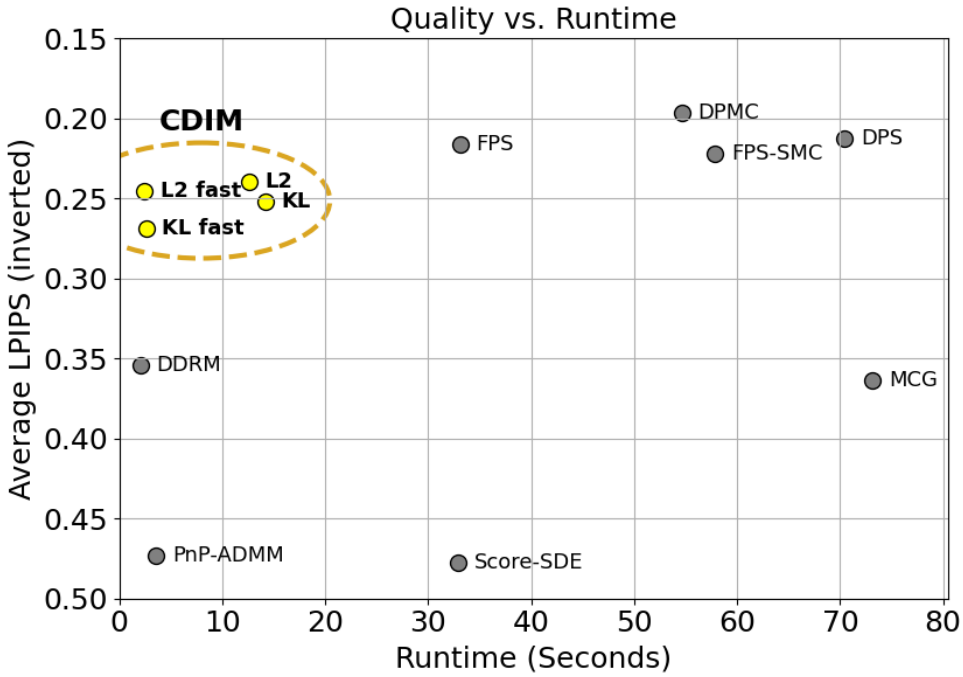


Figure 9.2: The inference speed and average LPIPS image quality score (inverted) averaged across multiple inverse tasks on the FFHQ dataset. The family of CDIM methods (top left corner) simultaneously achieves strong generation strong quality and fast inference compared to other inverse solvers.

FPS (Dou and Song, 2023), LGD (Song et al., 2023b), DPMC (Zhu et al., 2024), and MCG (Cardoso et al., 2023) focus on likelihood approximation for improved sampling. Guidance mechanisms have been incorporated through classifier gradients (Dhariwal and Nichol, 2021), data consistency enforcement (Chung et al., 2022c), and low-frequency feature matching Choi et al. (2021).

Other approaches use projection (Boys et al., 2023; Chung et al., 2024) or optimization (Chan et al., 2016; Wahlberg et al., 2012). DMPlug Wang et al. (2024) backpropagates through the entire diffusion process, leading to extremely slow inference. DSG (Yang et al., 2024) uses a similar optimization update to us for enforcing consistency with the partial

observation; however, it does not guarantee matching a constraint exactly, instead using a soft constraint, like DPS, to handle observational noise. Finally, works such as Blind DPS (Chung et al., 2022a) and FastEM (Laroche et al., 2023) solve inverse problems when the forward operator is unknown, a more difficult problem than the setting studied in this work.

9.2 Background

We work in the context of DDPM (Ho et al., 2020), which models a data distribution $q(\mathbf{x}_0)$ by modeling a sequence $t = 1, \ldots, T$ of smoothed distributions defined by

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1-\alpha_t)\mathbf{I}). \tag{9.2}$$

The degree of smoothing is controlled by a monotone decreasing noise schedule α_t with $\alpha_0 = 1$ (no noise) and $\alpha_T = 0$ (pure Gaussian noise).[1] The idea is to model a *reverse process* $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ that that incrementally removes the noise in $\mathbf{x}_t$ such that $p_\theta(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; 0, \mathbf{I})$ and $p(\mathbf{x}_0)$ approximates the data distribution, where $p(\mathbf{x}_0)$ is the marginal distribution of outputs from the reverse process:

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\, d\mathbf{x}_{1:T}. \tag{9.3}$$

Given noisy samples $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}$, where $\mathbf{x}_0$ is a sample from the data distribution and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, a diffusion model $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ is trained to predict $\boldsymbol{\epsilon}$:

$$\min_\theta \mathop{\mathbb{E}}_{\mathbf{x}_t, \boldsymbol{\epsilon}} \left[\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta (\mathbf{x}_t, t) \|^2 \right]. \tag{9.4}$$

To parameterize the reverse process $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, DDIM (Song et al., 2020a) exploits the Tweedie formula (Efron, 2011) for the posterior mean of a noisy observation:

$$\mathbb{E}\left[\mathbf{x}_0|\mathbf{x}_t\right] = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \sqrt{1-\alpha_t} \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) \right). \tag{9.5}$$

[1] We define α_t using the DDIM convention (Song et al., 2020a); our α_t corresponds to $\bar{\alpha}_t$ in Ho et al. (2020).

Using the denoising model $\boldsymbol{\epsilon}(\mathbf{x}_t, t)$ as a plug-in estimate of the score function $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$ (Vincent, 2011) we define the Tweedie estimate of the posterior mean:

$$\hat{\mathbf{x}}_0 \equiv \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \sqrt{1-\alpha_t} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) \approx \mathbb{E}\left[\mathbf{x}_0 | \mathbf{x}_t \right]. \tag{9.6}$$

And we use this estimator to define a DDIM forward process $\mathbf{x}_{t-1} = f_\theta(\mathbf{x}_t)$ defined by

$$x_{t-1} = f_\theta(\mathbf{x}_t) = \sqrt{\alpha_{t-1}} \hat{\mathbf{x}}_0 + \sqrt{1-\alpha_{t-1}} \left(\frac{\mathbf{x}_t - \sqrt{\alpha_t} \hat{\mathbf{x}}_0}{\sqrt{1-\alpha_t}} \right). \tag{9.7}$$

Unlike DDPM, the forward process defined by Equation (9.7) is deterministic; the value $p_\theta(\mathbf{x}_0)$ is entirely determined by $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ thus making DDIM an implicit model.

With a slight modification of the DDIM update, we are able to take larger denoising steps and accelerate inference. Given $\delta \geq 1$, we define an accelerated denoising process

$$x_{t-\delta} = f_\theta^\delta(\mathbf{x}_t) = \sqrt{\alpha_{t-\delta}} \hat{\mathbf{x}}_0 + \sqrt{1-\alpha_{t-\delta}} \left(\frac{\mathbf{x}_t - \sqrt{\alpha_t} \hat{\mathbf{x}}_0}{\sqrt{1-\alpha_t}} \right). \tag{9.8}$$

Using this process, inference is completed in just $T' \equiv T/\delta$ steps, albeit with degraded quality of the resulting sample $\mathbf{x}_0$ as δ becomes large.

9.3 Methods

We are interested in solving linear inverse problems of the form $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{y} \in \mathbb{R}^d$ is a linear measurement of $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{d \times n}$ describes the form of our measurements. For example, if $\mathbf{A} \in \{0,1\}^{n \times n}$ is a binary mask (which is the case for, e.g., in-painting or sparse recovery problems) then $\mathbf{y}$ describes a partial observation of $\mathbf{x}$. We seek an estimate $\hat{\mathbf{x}}$ that is consistent with our observations: in the noiseless case, $\mathbf{A}\hat{\mathbf{x}} = \mathbf{y}$. More generally, we seek to recover a robust estimate of $\hat{\mathbf{x}}$ when the observations $\mathbf{y}$ have been corrupted by noise: $\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\sigma}$, where $\boldsymbol{\sigma} \sim r^{\otimes d}$. Given a noise distribution r, we seek to minimize $D_{\mathrm{KL}}\left(\hat{r} \,\|\, r\right)$, where $\hat{r}$ is the empirical distribution of d residuals, e.g., $\mathbf{y} - \mathbf{A}\hat{\mathbf{x}} \in \mathbb{R}^d$, between noisy observations $\mathbf{y}$ and our estimates $\mathbf{A}\hat{\mathbf{x}}$.

We rely on a diffusion model $p_\theta(\mathbf{x})$ to identify an estimate $\hat{\mathbf{x}}$ that is both consistent with the observed measurements $\mathbf{y}$ and likely according to the model. In Section 9.3.1, we propose

a modification of the DDIM inference procedure to efficiently optimize the Tweedie estimates of $\hat{\mathbf{x}}_0$ to satisfy $\mathbf{A}\hat{\mathbf{x}}_0 = \mathbf{y}$ during the diffusion process, resulting in a consistent and likely final result $\mathbf{x}_0$. In Section 9.3.2 we extend this optimization-based inference procedure to account for noise in the observations $\mathbf{y}$. In Section 9.3.3 we describe an early-stopping heuristic to avoid overfitting to noisy observations, which further reduces the cost of inference. Finally, in Section 9.3.4 we describe heuristics for setting the step sizes of these optimization-based methods.

9.3.1 Optimizing $\hat{\mathbf{x}}_0$ to match the observations

For linear measurements $\mathbf{A}$, the Tweedie formula for $\hat{\mathbf{x}}_0$ (and the corresponding plugin-estimate Equation (9.6)) extends to a formula for the expected observations:

$$\mathbb{E}\left[\mathbf{y}|\mathbf{x}_t\right] = \mathbf{A}\mathbb{E}\left[\mathbf{x}_0|\mathbf{x}_t\right] \approx \mathbf{A}\hat{\mathbf{x}}_0. \tag{9.9}$$

For noiseless observations $\mathbf{y}$, we propose a modification of the DDIM updates Equation (9.7) to find $\mathbf{x}_{t-1}$ that satisfies the constraint $\mathbf{A}\hat{\mathbf{x}}_0 = \mathbf{y}$. I.e., at each time step t, we force the Tweedie estimate of the posterior mean of $q(\mathbf{y}|\mathbf{x}_t)$ to match the observed measurements $\mathbf{y}$:

$$\begin{aligned} \underset{\mathbf{x}_{t-1}}{\arg\min} \quad & \|\mathbf{x}_{t-1} - f_\theta(\mathbf{x}_t)\|^2 \\ \text{subject to} \quad & \mathbf{A}\hat{\mathbf{x}}_0 = \mathbf{y}. \end{aligned} \tag{9.10}$$

We can interpret Equation (9.10) as a projection of the DDIM update $f_\theta(\mathbf{x}_t)$ onto the set of values $\mathbf{x}_{t-1}$ that satisfy the constraint $\mathbf{A}\hat{\mathbf{x}}_0 = \mathbf{y}$. The full inference procedure is analogous to projected gradient descent, whereby we alternately take a step $f_\theta(\mathbf{x}_t)$ determined by the diffusion model, and then project back onto the constraint $\mathbf{A}\hat{\mathbf{x}}_0 = \mathbf{y}$. We implement the projection step itself via gradient descent, initialized from $\mathbf{x}_{t-1}^{(0)} = f_\theta(\mathbf{x}_t)$ and computing

$$\mathbf{x}_{t-1}^{(k)} = \mathbf{x}_{t-1}^{(k-1)} + \eta\nabla_{\mathbf{x}_{t-1}}\|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0\|^2. \tag{9.11}$$

As t approaches 0, $\hat{\mathbf{x}}_0$ converges to $\mathbf{x}_0$ and $\|\mathbf{y}-\mathbf{A}\hat{\mathbf{x}}_0\|^2$ becomes a simple convex quadratic, which can be minimized to arbitrary accuracy by taking sufficiently many gradient steps.

This allows us to guarantee exact recovery of the observations $\mathbf{y} = \mathbf{A}\mathbf{x}_0$ in the recovered inverse $\mathbf{x}_0$.

We face two conceptual challenges in optimizing Equation (9.10). First, for large t, no value $\mathbf{x}_t$ will satisfy $\mathbf{A}\hat{\mathbf{x}}_0 = \mathbf{y}$ and therefore the optimization is infeasible. Second, the estimate of the score function $\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t)$ from the diffusion model, $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$, may be inaccurate particularly at large t; we risk overfitting to a bad plug-in estimate $\hat{\mathbf{x}}_0$. [2]

In light of these observations, we replace Equation (9.10) with a Lagrangian

$$\arg\min_{\mathbf{x}_{t-1}} \|\mathbf{x}_{t-1} - f_\theta(\mathbf{x}_t)\|^2 + \lambda \|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0\|^2. \tag{9.12}$$

We can interpret Equation (9.12) as a relaxation of Equation (9.10); the regularization by $\lambda\|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0\|^2$ is achieved implicitly by stopping after $k = K$ steps of gradient descent. In contrast to a hard constraint at each timestep, this Lagrangian objective is robust to both (1) the possible infeasibility of $\mathbf{A}\hat{\mathbf{x}}_0 = \mathbf{y}$ and (2) overfitting the measurements based on an inaccurate Tweedie plug-in estimator. This regularization is needed at higher noise levels, so as $t \to 0$, we set $\lambda_t \to 0$ to recover the hard constraint. This can be achieved by running more optimization steps K at lower noise levels. See Figure 9.3 (b) for an example of satisfying a hard constraint that may be out of distribution.

9.3.2 Optimizing the KL Divergence of Residuals

For noisy inverse problems, imposing a hard constraint $\mathbf{A}\hat{\mathbf{x}}_0 = \mathbf{y}$ will overfit to noise $\boldsymbol{\sigma}$ in the measurements, as illustrated by Figure 9.3.

Previous work, such as DPS (Chung et al., 2022b) and DSG (Yang et al., 2024), account for noise by incompletely optimizing the objective $\mathbf{A}\hat{\mathbf{x}}_0 = \mathbf{y}$. In contrast, we propose to exactly optimize the Kullback-Leibler (KL) divergence between the empirical distribution of

[2] We illustrate both these claims by considering the Tweedie estimator Equation (9.6) in the case $t = T$. In this case, $\mathbf{x}_t \sim \mathcal{N}(0, I)$ is independent of $\mathbf{x}_0$ and therefore $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = \mathbb{E}[\mathbf{x}_0]$, the mean of the data distribution $q(\mathbf{x}_0)$. Unless $\mathbf{A}\mathbb{E}[\mathbf{x}_0] = \mathbf{y}$, the optimization is infeasible when $t = T$. Furthermore, we observe that when $t = T$, the plug-in estimator $\hat{\mathbf{x}}_0$ is not independent of $\mathbf{x}_t$ and $\hat{\mathbf{x}}_0 \neq \mathbb{E}[\mathbf{x}_0]$. This is indicative of error in the diffusion model, especially at high noise levels.

(a) (b) (c)

Figure 9.3: Results on a 50% noisy inpainting task. (a) is the noisy partial observation. (b) is generated by Algorithm 7 with a hard constraint, showing that we can exactly match the observation even when it's out of distribution. This is impossible with DPS, as it would simply generate the denoised image through soft optimization. (c) is generated by Algorithm 7 with early stopping.

residuals $R(\mathbf{A}\hat{\mathbf{x}}_0, \mathbf{y})$ and a known, i.i.d. noise distribution r:

$$\begin{aligned} \arg\min_{\mathbf{x}} \quad & \|\mathbf{x} - \mathbf{x}_t\|^2 \\ \text{subject to} \quad & D_{\mathrm{KL}}\left(R(\mathbf{A}\hat{\mathbf{x}}_0, \mathbf{y}) \,\|\, r\right) = 0. \end{aligned} \tag{9.13}$$

In Algorithm 6, we show how to optimize a constraint on categorical KL divergences to match arbitrary distributions of discretized residuals. We also provide a convenient objective for optimizing the empirical distribution of continuous residuals to match common noise patterns, including Gaussian and Poisson noise.

Additive Noise. The general additive noise model is defined by $\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\sigma} \in \mathbb{R}^d$, where $\boldsymbol{\sigma} \sim r^{\otimes d}$. By discretizing the distribution of residuals into B buckets, we can compute a categorical KL divergence between observed residuals and a discrete approximation of r_B

Algorithm 6 Constrained Diffusion Implicit Models with KL Constraints

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, T-\delta, \ldots, 1$ **do**
  $\mathbf{x}_{t-\delta} \leftarrow \sqrt{\bar{\alpha}_{t-\delta}} \left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t-\delta}}\epsilon_\theta(\mathbf{x}_t, t)$ ▷ Unconditional Generation
  for $k = 0, \ldots, K$ **do**
    $\hat{\mathbf{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_{t-\delta}}} \left(\mathbf{x}_{t-\delta} - \sqrt{1-\bar{\alpha}_{t-\delta}} \cdot \epsilon_\theta(\mathbf{x}_{t-\delta}, t-\delta) \right)$
    $\mathbf{x}_{t-\delta} \leftarrow \mathbf{x}_{t-\delta} + \eta \cdot \nabla_{\mathbf{x}_{t-\delta}} D_{\mathrm{KL}} \left(R(\mathbf{A}\hat{\mathbf{x}}_0, \mathbf{y}) \,\|\, r \right)$ ▷ Projection
  end for
end for
return $\hat{\mathbf{x}}_0$

of r:

$$D_{\mathrm{KL}} \left(R(\mathbf{A}\hat{\mathbf{x}}_0, \mathbf{y}) \,\|\, r_B \right) = \sum_{b=1}^{B} r_B(b) \log \left(\frac{r_B(b)}{\lfloor R(\mathbf{A}\hat{\mathbf{x}}_0, \mathbf{y}) \rfloor_B} \right). \tag{9.14}$$

In Figure 9.4 we show results on the box inpainting task when the observation has been corrupted with highly non-Gausian bimodal noise: $p(\boldsymbol{\sigma}_i = -0.75) = p(\boldsymbol{\sigma}_i = 0.75) = 0.5$ for $i = 1, \ldots, n$, where image pixels are normalized values $\mathbf{x}_i \in [-1, 1]$. We show that KL optimization (Algorithm 6) is far superior to the L^2 optimization explored later in Algorithm 7. It also greatly outperforms DPS which simply performs a soft optimization based on the noisy observation.

Gaussian Noise. Additive Gaussian noise is defined by $\boldsymbol{\sigma} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, in which case the residuals $R(\mathbf{Ax}, \mathbf{y}) \equiv \mathbf{y} - \mathbf{Ax} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ are i.i.d. with distribution $r \sim \mathcal{N}(0, \sigma^2)$. The empirical mean and variance of the residuals are

$$\hat{\mu} = \frac{1}{d} \sum_{i=1}^{d} R(\mathbf{A}\hat{\mathbf{x}}, \mathbf{y})_i, \quad \hat{\sigma}^2 = \frac{1}{d} \sum_{i=1}^{k} \left(R(\mathbf{A}\hat{\mathbf{x}}, \mathbf{y})_i - \hat{\mu} \right)^2. \tag{9.15}$$

Using the analytical formula for KL divergence between two Gaussians (Kingma and Welling, 2014a), we can match the empirical mean and variance of the residuals to r by

Input: Box inpainting with bimodal noise | Ground Truth | DPS | Ours - L^2 | Ours - Discrete KL

Figure 9.4: Results on the box inpainting task with a highly non-Gaussian bimodal noise distribution. Optimizing the discrete KL divergence to exactly match the known residuals (Alg 6) provides far better results than L^2 optimization (Alg 7) on this highly non-Gaussian noise. DPS also produces results that don't match the ovservation as it is performing a soft optimization.

enforcing

$$D_{\mathrm{KL}}\left(R(\mathbf{A}\hat{\mathbf{x}}_0, \mathbf{y}) \,\|\, r\right) = \log\left(\frac{\sigma^2}{\hat{\sigma}^2}\right) + \frac{\hat{\sigma}^2 + \hat{\mu}^2}{2\sigma^2} - \frac{1}{2} = 0. \tag{9.16}$$

Note that, although the discrete KL can be configured to enforce arbitrary precision on the distribution, the above analytical formula only enforces the KL divergence on the first two moments of the distribution $\hat{\mu}^2$ and $\hat{\sigma}^2$ under Gaussian residual noise.

Poisson Noise. Possion noise is non-additive noise defined by $s\mathbf{y} \sim \text{Poisson}(s\mathbf{Ax})$, where $\mathbf{y}$ is interpreted as discrete integer pixel values. The scaling factor $s \leq 1$ controls the degree of Poisson noise. Poisson noise is not identically distributed across $\mathbf{y}$; the variance increases with the scale of each observation. To remedy this, we consider the Pearson residuals (Pregibon, 1981):

$$R(\mathbf{A}\hat{\mathbf{x}}_0, \mathbf{y}) = \frac{\lambda(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0)}{\sqrt{\lambda\hat{\mathbf{x}}_0}}. \tag{9.17}$$

These residuals are identically distributed; moreover, they are approximately normal $r \sim \mathcal{N}(0, 1)$ (Pierce and Schafer, 1986). We can therefore optimize the KL divergence

Algorithm 7 Constrained Diffusion Implicit Models with L^2 Constraints and Early Stopping

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, T-\delta, \ldots, 1$ **do**
  $\mathbf{x}_{t-\delta} \leftarrow \sqrt{\bar{\alpha}_{t-\delta}}\left(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-\delta}}\epsilon_\theta(\mathbf{x}_t, t)$ ▷ Unconditional Generation
  for $k = 0, \ldots, K$ **do**
    $\hat{\mathbf{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_{t-\delta}}}\left(\mathbf{x}_{t-\delta} - \sqrt{1-\bar{\alpha}_{t-\delta}}\epsilon_\theta(\mathbf{x}_{t-\delta}, t-\delta)\right)$
    ▷ Early Stopping
    if $\frac{1}{d}||R(\mathbf{A}\hat{\mathbf{x}}_0, \mathbf{y})||_2^2 < Var(r)$ **then**
      break
    end if
    $\mathbf{x}_{t-\delta} \leftarrow \mathbf{x}_{t-\delta} + \eta \cdot \nabla_{\mathbf{x}_{t-\delta}} \frac{1}{d}\|R(\mathbf{A}\hat{\mathbf{x}}, \mathbf{y})\|_2^2$ ▷ Projection
  end for
end for

between Pearson residuals and a standard normal using Equation (9.16) to solve inverse problems with Poisson noise. Although the Pearson residuals closely follow the standard normal distribution for positive values of $\hat{\mathbf{x}}_0$, this breaks down for values of $\hat{\mathbf{x}}_0$ close to zero, and extreme noise levels s. In practice we find the Gaussian assumption to be valid for natural images corrupted by as much noise as $s \approx 0.025$. In Figure 9.1 we show an example of denoising an image corrupted by Poisson noise with $s = 0.05$.

9.3.3 L^2 Optimization with Early Stopping

We now establish an important connection between KL and L^2 optimization. For linear operators under Gaussian measurement noise, optimizing the L^2 loss of the residuals with early stopping is the same as KL divergence minimization. To show this, consider Equation 9.16, which describes the KL divergence between two Gaussians. First, observe that the L^2 loss of the residuals, $||\mathbf{A}\hat{\mathbf{x}}_0 - \mathbf{y}||_2^2$, equals $d(\hat{\sigma}^2 + \hat{\mu}^2)$, where d is the dimension of the measurement space. For many common linear inverse problems - such as inpainting, super-

resolution, and deblurring - the measurement operator $\mathbf{A}$ does not introduce systematic bias in the output, $\mathbb{E}_{\mathbf{x}\sim q}[\|\mathbf{A}\mathbf{x}\|^2] \approx \mathbb{E}_{\mathbf{x}\sim q}[\|\mathbf{x}\|^2]$. As a result, the residual mean is approximately 0: $\hat{\mu} = \frac{1}{d}\sum_i(\mathbf{A}\hat{\mathbf{x}}_0 - \mathbf{y}) \approx 0$, which we also observe empirically.[3] If we run L^2 optimization until the normalized L^2 loss equals our target residual variance σ^2 (presented in Algorithm 7), we have:

$$\frac{1}{d}||\mathbf{A}\hat{\mathbf{x}}_0 - \mathbf{y}||_2^2 = \hat{\sigma}^2 + \hat{\mu}^2 \approx \hat{\sigma}^2 = \sigma^2 \tag{9.18}$$

Substituting this result into Equation 9.16 yields:

$$D_{\mathrm{KL}}\left(R(\mathbf{A}\hat{\mathbf{x}}_0, \mathbf{y}) \,\|\, r\right) = \log\left(\frac{\sigma^2}{\sigma^2}\right) + \frac{\sigma^2}{2\sigma^2} - \frac{1}{2} = 0 \tag{9.19}$$

This equivalence demonstrates that optimizing the L^2 loss of the residuals until $\frac{1}{d}||R(\mathbf{A}\hat{\mathbf{x}}_0, \mathbf{y})||_2^2 = \sigma^2$ achieves KL divergence minimization of Gaussian noise up to the first two moments. However, it's important to note the limitation that L^2 optimization implicity tries to fit the residuals to a Gaussian distribution. Therefore, Algorithm 7 fails for highly non-Gaussian noise distributions, as demonstrated in Figure 9.4.

In practice, L^2 optimization often converges faster than KL optimization for Gaussian measurement noise for two reasons: 1) we can stop running projection steps immediately once $\frac{1}{d}||R(\mathbf{A}\hat{\mathbf{x}}_0, \mathbf{y})||_2^2$ drops below σ^2, rather than continuing to run KL optimization steps for the entire inference process, and 2) during the early phases of KL optimization, the gradient through the partition function $\log(\frac{\sigma^2}{\hat{\sigma}^2})$ dominates, leading to less efficient updates on individual pixel values.

Algorithm 7 can also handle unknown additive noise distributions r, provided we can bound the variance $Var(r)$ to use as a stopping criterion (see Fig 9.9b).

9.3.4 Choice of Step Size η

An important hyperparameter is the step size η. DPS sets η proportional to $1/\|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0\|$. We find that this fails to converge for KL optimization, and also produces unstable results

[3] For tasks that introduce systematic bias, the stopping criteria could depend on both $\hat{\sigma}^2$ and $\hat{\mu}^2$ to ensure fidelity to the KL divergence.

Input | $\eta \propto 1/\left\|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0\right\|$ | $\eta \propto 1/\left\|\nabla_{\mathbf{x}_{t-\delta}}\right\|$ | $\eta \propto 1/\mathbb{E}\left\|\nabla_{\mathbf{x}_{t-\delta}}\right\|$

Figure 9.5: Comparison of different step size schedules on a 50% inpainting task. We choose a challenging task with $T' = 10$, $K = 10$, $\sigma_y^2 = 0.15$ and use Algorithm 7. $\eta \propto 1/\mathbb{E}\left\|\nabla_{\mathbf{x}_{t-\delta}}\right\|$ is the most stable and converges the fastest.

for L^2 optimization when T' is small. This is because $\|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}_0\| \to 0$ towards the end of the optimization, leading to extremely large steps. One option is to set η inversely proportional to the magnitude of the gradient $\|\nabla \mathbf{x}_{t-\delta}\|$ at every single optimization step. Although this is the easiest solution, it can also result in unstable oscillations and slower convergence. Instead, we propose to set η inversely proportional to $\mathbb{E}_{\mathbf{x}\sim\mathcal{X}_{\text{train}}}\left\|\nabla_{\mathbf{x}_{t-\delta}}\right\|$, a common optimization heuristic (Amari, 1998; Pascanu and Bengio, 2014). In Appendix B.1 we describe how to compute this expectation. In Figure 9.5 we show qualitatively what happens with different η schedules.

We find that for a specific optimization objective and task, the magnitude of the gradient $\|\nabla \mathbf{x}_{t-\delta}\|$ is highly similar across data points, datasets, and model architectures. We empirically demonstrate this observation in Appendix B.1. This suggests that a learned step size based on $\mathbb{E}_{\mathbf{x}\sim\mathcal{X}_{\text{train}}}\left\|\nabla_{\mathbf{x}_{t-\delta}}\right\|$ generalizes as a good learning rate for unseen data. For all experiments, we estimate these magnitudes from FFHQ training data.

9.4 Results and Experiments

We conduct experiments to demonstrate the efficiency and quality of CDIM across various tasks and datasets. In Section 9.4.1, we present quantitative comparisons to state-of-the-

art approaches, followed by ablation studies in Section 9.4.3 examining inference speed and hyperparameters. In Section 9.4.4 we explore two novel applications of diffusion models for inverse problems.

Table 9.1: Quantitative results (FID, LPIPS) of our model and existing models on various linear inverse problems on FFHQ 256 $\times$ 256-1k validation dataset. (Lower is better). The best result is in **bold** and the second best is underlined. We note that our L^2 method is better than KL on this Gaussian additive noise task, which is expected based on the discussion in Section 9.3.3.

FFHQ	**Super Res**		**Inpainting (box)**		**Gaussian Deblur**		**Inpainting (random)**		**Runtime (seconds)**
Methods	FID	LPIPS	FID	LPIPS	FID	LPIPS	FID	LPIPS	
Ours - KL fast	36.76	0.283	35.15	0.2239	37.44	0.308	35.73	0.259	2.57
Ours - L^2 fast	33.87	0.276	27.51	0.1872	34.18	0.276	29.67	0.243	2.4
Ours - KL	34.71	0.269	30.88	0.1934	35.93	0.296	31.09	0.249	10.2
Ours - L^2	31.54	0.269	**26.09**	0.196	**29.68**	**0.252**	28.52	0.240	9.0
FPS-SMC	**26.62**	**0.210**	26.51	**0.150**	29.97	0.253	33.10	0.275	116.90
DPS	39.35	0.214	33.12	0.168	44.05	0.257	**21.19**	**0.212**	70.42
DDRM	62.15	0.294	42.93	0.204	74.92	0.332	69.71	0.587	2.0
MCG	87.64	0.520	40.11	0.309	101.2	0.340	29.26	0.286	73.2
PnP-ADMM	66.52	0.353	151.9	0.406	90.42	0.441	123.6	0.692	3.595
Score-SDE	96.72	0.563	60.06	0.331	109.0	0.403	76.54	0.612	32.39
ADMM-TV	110.6	0.428	68.94	0.322	186.7	0.507	181.5	0.463	-

9.4.1 Numerical Results on FFHQ and ImageNet

We evaluate CDIM on the FFHQ-1k (Karras et al., 2019) and ImageNet-1k (Russakovsky et al., 2015) validation sets. Each dataset contains 256 $\times$ 256 RGB images scaled to the range [0, 1]. The tasks include 4x super-resolution, box inpainting, Gaussian deblur, and random inpainting. Details of each task are included in the appendix. For all tasks, we apply zero-centered Gaussian observational noise with $\sigma = 0.05$. To ensure fair comparisons, use identical pre-trained diffusion models used in the baseline methods: for FFHQ we use the network from Chung et al. (2022b) and for ImageNet we use the network from Dhariwal and Nichol (2021). We report Frechet Inception Distance (FID) (Heusel et al., 2018) and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018a) with peak signal-to-noise ratio (PSNR) results in Appendix B.2.6. All experiments are carried out on a single Nvidia A100 GPU.

In Table 9.1 we compare CDIM with several other inverse solvers using the FID and LPIPS metrics on the FFHQ dataset. We present results using both our KL divergence optimization method (Algorithm 6) and our L^2 optimization method (Algorithm 7) with early stopping. For these experiments, we present results with $T' = 50$ and $K = 3$ as well as $T' = 25$ and $K = 1$ labeled as "fast". For ImageNet results please see Appendix B.2.5.

9.4.2 Comparison with DPS using DDIM

We show a qualitative comparison against DPS Chung et al. (2022b) when we combine it DDIM and fewer steps (see Figure 9.6). We use the core DPS sampling algorithm, but with DDIM as the denoising algorithm instead of DDPM. The number of denoising steps is set to 50 and the step size of DPS is scaled to achieve the best convergence possible.

9.4.3 Ablation Studies

Number of Inference Steps CDIM offers the flexibility to trade off quality for faster inference time on demand. We investigate how generation quality changes as we vary the

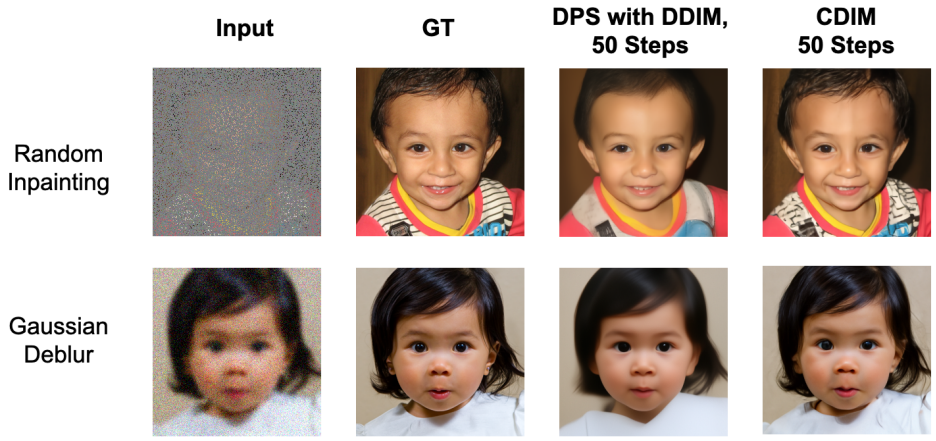


Figure 9.6: We show a comparison against DPS Chung et al. (2022b) combined with DDIM. The step size γ of DPS was tuned to acheive the best results without diverging. If you just run DPS with DDIM and fewer steps, the output does not accurately match the observation; it is blurry and does not match the constraint. Both algorithms use 50 total steps.

total computational budget during inference. Recall that the total number of network passes during inference is $T'(K+1)$, where T' is the number of denoising steps and K is the number of optimization steps per denoising step. We use the random inpainting task on the FFHQ dataset with the setup described in the previous section. For this experiment we use KL optimization (Algorithm 6). The total network forward passes are varied from 200 to 20, and we show qualitative results. Notably, CDIM yields high quality samples with as few as 50 total inference steps, with quality degradations after that.

T′ vs K Trade-Off We consider the optimal balance between T' and K when the total number of inference steps $T'(K+1)$ is fixed. Using the random inpainting task on the FFHQ dataset with the previously described setup, we set $T'(K+1) = 200$ and analyze how PSNR, FID, and LPIPS change based on the chosen T' and K values. Results are plotted in Figure B.3. FID results consistently favor the maximum number of denoising steps T'

Input	10 Steps	20 Steps	50 Steps	200 Steps
Random Inpainting	$T' = 5$ $K = 1$	$T' = 10$ $K = 1$	$T' = 25$ $K = 1$	$T' = 50$ $K = 3$

Figure 9.7: We reduce the total number of inference steps $T'(K+1)$ and visualize the results. There is almost no visible degradation until fewer than 50 total steps.

with minimal optimization steps K. This is because FID evaluates overall distribution similarity rather than per-sample fidelity, and thus is not penalized by lower reconstruction-observation fidelity. In contrast, PSNR and LPIPS achieve optimal results with a balanced mix of denoising and optimization steps.

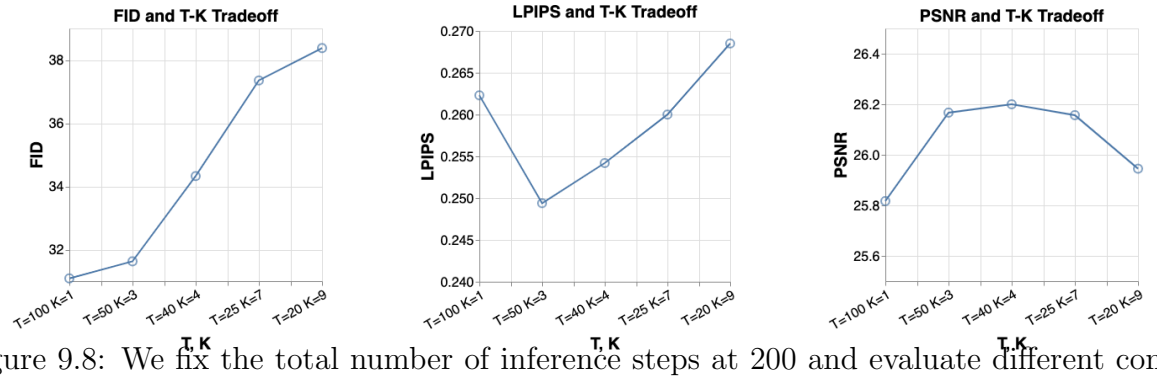


Figure 9.8: We fix the total number of inference steps at 200 and evaluate different combinations of T' and K. FID always prefers more denoising steps T', while LPIPS and PSNR are best at a mix of T' and K steps.

9.4.4 Additional Applications

Time-Travel Rephotography In Figure 9.1 we showcase an application of time-travel rephotography Luo et al. (2021). Antique cameras lack red light sensitivity, exaggerating wrinkles by filtering out skin subsurface scatter which occurs mostly in the red channel. To address this, we input the observed image into the blue color channel and use the pretrained FFHQ model with Algorithm 7 to project the face into the space of modern images. We further emphasize the power of our approach; Luo et al. (2021) trained a specialized model for this task while we are able to use a pretrained model without modification.

Sparse Point Cloud Reprojection For this task, 20 different images from a scene in The Grand Budapest Hotel scene were entered into Colmap (Schönberger and Frahm, 2016) to generate a sparse 3D point cloud. Projections of this sparse point cloud have roughly 90% of the pixels missing. Furthermore, the observations often contain significant amounts of non-Gaussian noise due to false correspondences. We can formulate this as noisy inpainting problem and use Algorithm 7 along with a variance threshold that adequately captures the imprecise nature of the point cloud. We showcase the results in Figure 9.9.

9.5 Conclusion

In this chapter we introduced CDIM, a new approach for solving noisy linear inverse problems with pretrained diffusion models. By projecting the DDIM updates, such that Tweedie estimates of the denoised image $\hat{\mathbf{x}}_0$ match the linear constraints, we can enforce constraints without making out-of-distribution edits to the noised iterates $\mathbf{x}_t$. Note that our method cannot handle non-linear constraints, including latent diffusion, because for a non-linear function h, $\mathbb{E}\left[h(\mathbf{x}_0)\right] \neq h(\mathbb{E}\left[\mathbf{x}_0\right])$. Therefore, we cannot extend Tweedie's estimate of the posterior mean $\mathbf{x}_0$ to an estimate of the posterior mean of non-linear observations $h(\mathbf{x}_0)$. However, for linear constraints, our method generates high quality images with faster inference than previous methods, creating a new point on the Pareto-frontier of quality vs. efficiency for linear inverse problems.

(a) (b)

Figure 9.9: Using noisy inpainting to tackle sparse point cloud reprojection. (a) Shows a sparse point cloud projected to a desired camera angle. (b) Shows the result after our method is used for noisy inpainting.

Chapter 10

CONCLUSION AND PERSPECTIVES

This thesis explored two complementary approaches for restoring noisy and incomplete signals in our reality: spatial audio and generative models. In Chapter 2, we introduced the fundamentals of multi-microphone array processing and spatial audio, discussing key considerations such as array geometry, beamforming methods, and the role of the head-related transfer function (HRTF) in spatial audio rendering. Building on these foundations, Chapter 3 presented The Cone of Silence, a binary search-based algorithm for simultaneous source separation and localization using a circular microphone array. This approach demonstrated robustness in handling an unknown number of potentially moving speakers in a variety of scenarios. The problem of speech isolation during noisy phone calls was explored further in Chapter 4 where we introduced ClearBuds. In this chapter, we showcased a custom built binaural headset that captured time-synchronized audio paired with a lightweight speech enhancement network on a mobile phone. This time-synchronized headset was further used as a basis for HRTF estimation in chapter 5, where we leveraged binaurally captured audio in everyday settings to estimate personalized HRTFs. This method enabled realistic spatial audio rendering without the need for complex measurement setups.

In the second part of this thesis, we shifted focus to generative models as a powerful tool for signal restoration. Unlike multi-microphone approaches, generative models eliminate the need for specialized hardware and generalize more easily across different modalities like audio and vision. Chapter 6 provided an overview of the principles underlying generative modeling, covering flow models, autoregressive models, score-based models, and diffusion models. We also showed how to frame signal restoration as a Bayesian inverse problem, with generative models serving as priors. Chapter 7 introduced the Bayesian Annealed Signal Informed Sepa-

ration (BASIS) algorithm, which leverages score-based and flow models for source separation. This chapter also explored applications such as image colorization and introduced new evaluation metrics for source separation. The BASIS algorithm was extended in Chapter 8 to autoregressive models, resulting in the Parallel and Flexible Sampling (PnF) method. Here, we demonstrated how discrete autoregressive models like WaveNet and PixelCNN++ could be adapted for gradient-based sampling, showcasing in-the-wild results on tasks such as musical instrument separation, audio and image inpainting, and cough removal from recordings. Finally, Chapter 9 presented a novel approach for solving noisy linear inverse problems using diffusion models. This method uniquely combines exact measurement constraint satisfaction with accelerated sampling techniques, setting it apart from prior work.

10.1 Future Work

We now consider some follow up works that would proceed directly from the work presented in this thesis.

Solving Inverse Problems with Latent Diffusion Models Chapter 9 presented a fast and exact method for solving linear inverse problems with diffusion models. Unfortunately, state-of-the-art diffusion models, like Stable Diffusion (Rombach et al., 2022) operate on a latent space parametrized by a VAE. We use $\mathbf{z}$ to denote a latent space measurement that corresponds to a pixel space measurement $\mathbf{x} = \mathcal{D}(\mathbf{z})$ for a VAE Decoder $\mathcal{D}$. Because the decoder is highly non-linear, methods like CDIM that rely on the Tweedie's Estimate $\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t]$ of the latent mean, cannot be used to estimate the pixel space expectation $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$ by simply passing the latent expectation through the decoder. Formally,

$$\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t] = \mathbb{E}[\mathcal{D}(\mathbf{z}_0)|\mathbf{z}_t] \neq \mathcal{D}(\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t]) \tag{10.1}$$

Existing works have attempted to use the flawed Tweedie's estimate $\mathcal{D}(\mathbb{E}[\mathbf{z}_0|\mathbf{z}_t])$ to guide latent diffusion towards pixel space observations with reasonable success (Rout et al., 2024a,b; Chung et al., 2023). However, the demonstrated tasks are often limited to simpler inverse

problems such as inpainting small regions of images. We imagine a future method that could learn $\mathbb{E}\left[\mathcal{D}(\mathbf{z}_0)|\mathbf{z}_t\right]$ through an auxiliary model, instead of relying on the inherently inaccurate $\mathcal{D}(\mathbb{E}\left[\mathbf{z}_0|\mathbf{z}_t\right])$.

In order to learn this model, it would be necessary to simulate many denoising trajectories for noisy latents $\mathbf{z}_t$ at all diffusion noise levels across many images. Those would then be averaged in the pixel space, giving paired examples of $\mathbf{z}_t$ and $\mathbb{E}\left[\mathcal{D}(\mathbf{z}_0)|\mathbf{z}_t\right]$. Next, an auxiliary network would be trained to predict $\mathbb{E}\left[\mathcal{D}(\mathbf{z}_0)|\mathbf{z}_t\right]$ from z_t using the supervised examples. Finally, during inference, we could guide the latent diffusion process towards desired pixel space observations with the gradient $\nabla_{\mathbf{z}_t}\,\mathbb{E}\left[\mathcal{D}(\mathbf{z}_0)|\mathbf{z}_t\right]$

Real-Time Language Translation One of the principal goals in this thesis was to help improve communication, particularly in a global and increasingly virtual world. Although we primarily explored communication through the audio and visual signals, a major barrier that still exists is language access. Across healthcare, media, international business, and politics, the need for real-time language translation has never been higher. Imagine a video conferencing software where participants can speak in their own language, while having their conversation translated in real-time without noticeable lag.

Unfortunately the key problem to overcome is the subject-verb order difference in languages. Suppose we want to translate a sentence from language A to B. Sometimes it is not possible to begin the translated sentence in language B until receiving the entirety of the sentence in language A. This single sentence latency would make real-time communication challenging and unnatural. Consider the sentence in Hindi: *Mujhe films aur sports nahin pasandh hain.* This word order literally translates to *I films and cricket don't like.* To produce the grammatically correct sentence *I don't like films and sports*, we would have to wait until receiving the entire hindi sentence, as the verb *don't like* comes after the subject *films and sports.*

We can imagine a possible solutions where we fine-tune an LLM to produce outputs that minimize the subject verb order difference between sentences. For example, the sentence *I*

don't like films and sports could also be phrased as *For me, films and sports are not enjoyable.* The latter aligns much more closely with the hindi word order, and would allow a streaming translation. A training approach would involve prompting an LLM to produce multiple candidate translations, and scoring them on the maximum translation latency that would be incurred based on the word order. The optimal output can then be used during fine-tuning to train the LLM to produce the desired translations.

10.2 Perspectives

Reflecting back on the progression of research conducted in this thesis, we see two overarching trends are poised to shape the future. The first is the rapid advancement of consumer hardware and edge computing capabilities, particularly for audio applications. During the course of this thesis, we witnessed the release of devices such as the Apple AirPods Pro (2019), Oculus Quest (2019), Meta Ray-Bans (2023), and Apple Vision Pro (2024). Each new generation of devices pushed the boundary of audio capture configurations and processing ability. For instance, the iPhone 12 Pro Max, used in the ClearBuds project, has been succeeded by four newer generations, with the iPhone 16 offering three times the flops (CPU Monkey, 2024). This trend suggests a future where sophisticated neural networks can run in real-time on edge devices, enabling new applications. For example, one can imagine selective sound isolation running directly on future AirPods or real-time language translation with spatially consistent audio playback running on headsets. Such advancements will undoubtedly unlock new possibilities for immersive and personalized audio experiences.

The second trend is the rapid and continued improvement of generative models. Early work in this thesis relied on models like Glow and NCSN, which have since been surpassed by state-of-the-art alternatives. Today, generative models can effortlessly produce high-resolution images, videos, and even full-length musical compositions, rendering evaluation on datasets like MNIST and CIFAR-10 obsolete. One advantage, however, is that the generative methods presented in this thesis (BASIS, PnF, CDIM) are agnostic to the underlying generative model and can be applied with newer and stronger models of the same model

type. For example, recent advances in pixel-space diffusion models have the potential to significantly enhance the performance of CDIM (Imagen-Team-Google et al., 2024). While the pace of progress in generative modeling is staggering, fundamental challenges such as fast and consistent sampling from posterior distributions will remain central to future research.

Overall, we are excited about the continued possibilities for enhancing communication, preserving memories, and improving creative workflows with spatial audio and generative models. We hope that future research can continue to bring this kind of cutting edge technology into the lives of people worldwide.

BIBLIOGRAPHY

https://www.microsoft.com/en-us/research/academic-program/deep-noise-suppression-challenge-interspeech-2021/.

https://en.wikipedia.org/wiki/Cone_of_Silence_(Get_Smart), 1960.

Setting up the timeslot api. https://devzone.nordicsemi.com/nordic/short-range-guides/b/software-development-kit/posts/setting-up-the-timeslot-api, Jul 2015.

Bluetooth Core Specification v5.0, 2016.

Wireless timer synchronization among nrf5 devices. https://devzone.nordicsemi.com/nordic/short-range-guides/b/bluetooth-low-energy/posts/wireless-timer-synchronization-among-nrf5-devices, Jul 2016.

Marwa A. Abd El-Fattah, Moawad I. Dessouky, Alaa M. Abbas, Salaheldin M. Diab, El-Sayed M. El-Rabaie, Waleed Al-Nuaimy, Saleh A. Alshebeili, and Fathi E. Abd El-Samie. Speech enhancement with an adaptive wiener filter. *Int. J. Speech Technol.*, 17(1):53–64, March 2014. ISSN 1381-2416. doi: 10.1007/s10772-013-9205-5. URL https://doi.org/10.1007/s10772-013-9205-5.

Sharath Adavanne, Archontis Politis, Joonas Nikunen, and Tuomas Virtanen. Sound event localization and detection of overlapping sources using convolutional recurrent neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 13(1):34–48, 2018.

Hidri Adel, Meddeb Souad, Abdulqadir Alaqeeli, and Amiri Hamid. Beamforming techniques for multichannel audio signal separation. *arXiv preprint arXiv:1212.6080*, 2012.

Amir Adler, Valentin Emiya, Maria G Jafari, Michael Elad, Rémi Gribonval, and Mark D Plumbley. Audio inpainting. *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

Karan Ahuja, Andy Kong, Mayank Goel, and Chris Harrison. Direction-of-voice (dov) estimation for intuitive speech interaction with smart devices ecosystems. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, page 1121–1131, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375146. doi: 10.1145/3379337.3415588. URL https://doi.org/10.1145/3379337.3415588.

V Ralph Algazi, Richard O Duda, Dennis M Thompson, and Carlos Avendano. The cipic hrtf database. In *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575)*, pages 99–102. IEEE, 2001.

JB Alien and DA Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 60(S1):S9–S9, 1976.

Jont B Allen and David A Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

Shun-ichi Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10 (2):251–276, 02 1998. ISSN 0899-7667. doi: 10.1162/089976698300017746. URL https://doi.org/10.1162/089976698300017746.

Amazon. Echo (3rd gen). https://www.amazon.com/all-new-echo/dp/b07nftvp7p, 2018.

Shahedul Amin, Riyasat Azim, Syed Prantik Rahman, Ferdous Habib, and Ashraful Hoque. Estimation of direction of arrival (doa) using real- time array signal processing and performance analysis. 2010. URL https://api.semanticscholar.org/CorpusID:268071642.

Apple. Apple airpods. https://www.apple.com/airpods/, 2018.

Apple. Cmheadphonemotionmanager, 2023a. URL https://developer.apple.com/documentation/coremotion/cmheadphonemotionmanager. Accessed on: June 1, 2023.

Apple. Airpods (3rd generation), 2023b. URL https://www.apple.com/airpods-3rd-generation/specs/. Accessed on: June 1, 2023.

Apple. Listen with personalized spatial audio for airpods and beats, 2023c. URL https://support.apple.com/en-us/HT213318. Accessed on: June 1, 2023.

Apple Insider. https://appleinsider.com/articles/21/03/30/apple-airpods-beats-dominated-audio-wearable-market-in-2020, 2021.

Futoshi Asano and Hideki Asoh. Sound source localization and separation based on the em algorithm. In *ISCA Tutorial and Research Workshop (ITRW) on Statistical and Perceptual Audio Processing*, 2004.

Francis R Bach and Michael I Jordan. Blind one-microphone speech separation: A spectral learning approach. In *Advances in neural information processing systems*, pages 65–72, 2005.

Jinglin Bai, Hao Li, Xueliang Zhang, and Fei Chen. Attention-based beamformer for multi-channel speech enhancement. *arXiv preprint arXiv:2409.06456*, 2024.

Jon Barker, Shinji Watanabe, Emmanuel Vincent, and Jan Trmal. The fifth'chime'speech separation and recognition challenge: dataset, task and baselines. *arXiv preprint arXiv:1803.10609*, 2018.

Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. Blazeface: Sub-millisecond neural face detection on mobile gpus, 2019.

Durand R Begault and Leonard J Trejo. 3-d sound for virtual reality and multimedia. Technical report, 2000.

Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

Pierre C Bellec, Guillaume Lecué, Alexandre B Tsybakov, et al. Slope meets lasso: improved oracle bounds and optimality. *The Annals of Statistics*, 46(6B):3603–3642, 2018.

zamir ben hur, david alon, philip w. robinson, and ravish mehra. localization of virtual sounds in dynamic listening using sparse hrtfs. *journal of the audio engineering society*, august 2020.

Laurent Benaroya, Frédéric Bimbot, and Rémi Gribonval. Audio source separation with a single sensor. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1): 191–199, 2005.

J. Benesty, J. Chen, and Y. Huang. *Microphone Array Signal Processing*. Springer Topics in Signal Processing. Springer Berlin Heidelberg, 2008. ISBN 9783540786122. URL https://books.google.com/books?id=rFfF6BStEGIC.

Nancy Bertin, Roland Badeau, and Emmanuel Vincent. Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):538–549, 2010.

Çağdaş Bilen, Alexey Ozerov, and Patrick Pérez. Joint audio inpainting and source separation. In *International Conference on Latent Variable Analysis and Signal Separation*, 2015.

Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT Press, 1987.

Jens Blauert. *Spatial hearing: the psychophysics of human sound localization*. MIT press, 1997.

Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning*, 2017.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. Audiolm: a language modeling approach to audio generation, 2023. URL https://arxiv.org/abs/2209.03143.

Benjamin Boys, Mark Girolami, Jakiw Pidstrigach, Sebastian Reich, Alan Mosca, and O. Deniz Akyildiz. Tweedie moment projected diffusions for inverse problems, 2023.

Michael Brandstein and Darren Ward. *Microphone arrays: signal processing techniques and applications*. Springer Science & Business Media, 2001.

Michael S. Brandstein and Harvey F. Silverman. A practical methodology for speech source localization with microphone arrays. *Comput. Speech Lang.*, 11:91–126, 1997. URL https://api.semanticscholar.org/CorpusID:14278721.

Andreas Brendel and Walter Kellermann. Learning-based acoustic source-microphone distance estimation using the coherent-to-diffuse power ratio. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 61–65. IEEE, 2018.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020a. URL https://arxiv.org/abs/2005.14165.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language

models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020b.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Nam Bui, Nhat Pham, Jessica Jacqueline Barnitz, Zhanan Zou, Phuc Nguyen, Hoang Truong, Taeho Kim, Nicholas Farrow, Anh Nguyen, Jianliang Xiao, Robin Deterding, Thang Dinh, and Tam Vu. Ebp: An ear-worn device for frequent and comfortable blood pressure monitoring. *Commun. ACM*, 64(8):118–125, jul 2021. ISSN 0001-0782. doi: 10.1145/3470446. URL https://doi.org/10.1145/3470446.

Cox R. Neto S.F.de C. Lamblin C. and Sherif M.H. Itu-t coders for wideband, superwideband, and fullband speech communication. IEEE, 2009.

J. Capon. High-resolution frequency-wavenumber spectrum analysis. *Proceedings of the IEEE*, 57(8):1408–1418, 1969. doi: 10.1109/PROC.1969.7278.

Gabriel Cardoso, Yazid Janati El Idrissi, Sylvain Le Corff, and Eric Moulines. Monte carlo guided diffusion for bayesian linear inverse problems, 2023. URL https://arxiv.org/abs/2308.07983.

J-F Cardoso. Blind signal separation: statistical principles. *Proceedings of the IEEE*, 86(10): 2009–2025, 1998.

CCITT. Pulse code modulation (pcm) of voice frequencies. International Telecommunication Union, 1988.

Enea Ceolini, Jens Hjortkjær, Daniel Wong, James O'Sullivan, Vinay Raghavan, Jose Herrero, Ashesh Mehta, Shih-Chii Liu, and Nima Mesgarani. Brain-informed speech separation (biss) for enhancement of target speaker in multitalker speech perception. *NeuroImage*, 223:117282, 08 2020. doi: 10.1016/j.neuroimage.2020.117282.

Justin Chan, Sharat Raju, Rajalakshmi Nandakumar, Randall Bly, and Shyamnath Gollakota. Detecting middle ear fluid using smartphones. *Science Translational Medicine*, 11: eaav1102, 05 2019. doi: 10.1126/scitranslmed.aav1102.

Justin Chan, Ali Najafi, Mallory Baker, Julie Kinsman, Lisa Mancl, Susan Norton, Randall Bly, and Shyamnath Gollakota. Performing tympanometry using smartphones. *Communications Medicine*, 06 2022.

Stanley H. Chan, Xiran Wang, and Omar A. Elgendy. Plug-and-play admm for image restoration: Fixed point convergence and applications, 2016. URL https://arxiv.org/abs/1605.01710.

Ya-Liang Chang, Kuan-Ying Lee, Po-Yu Wu, Hung-yi Lee, and Winston Hsu. Deep long audio inpainting. *arXiv preprint arXiv:1911.06476*, 2019.

Ishan Chatterjee, Maruchi Kim, Vivek Jayaram, Shyamnath Gollakota, Ira Kemelmacher, Shwetak Patel, and Steven M. Seitz. ClearBuds. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*. ACM, jun 2022. doi: 10.1145/3498361.3538933. URL https://doi.org/10.1145%2F3498361.3538933.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, 2014.

Tzu-Yu Chen, Tzu-Hsuan Kuo, and Tai-Shih Chi. Autoencoding hrtfs for dnn based hrtf personalization using anthropometric features. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 271–275. IEEE, 2019.

Zhuo Chen, Xiong Xiao, Takuya Yoshioka, Hakan Erdogan, Jinyu Li, and Yifan Gong. Multi-channel overlapped speech recognition with location guided speech extraction network. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 558–565. IEEE, 2018.

Corey I Cheng and Gregory H Wakefield. Introduction to head-related transfer functions (hrtfs): Representations of hrtfs in time, frequency, and space. In *Audio Engineering Society Convention 107*. Audio Engineering Society, 1999.

Amit Chhetri, Philip Hilmes, Trausti Kristjansson, Wai Chu, Mohamed Mansour, Xiaoxue Li, and Xianxian Zhang. Multichannel audio front-end for far-field automatic speech recognition. In *2018 EUSIPCO*, pages 1527–1531. IEEE, 2018.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Hyeong-Seok Choi, Jang-Hyun Kim, Jaesung Huh, Adrian Kim, Jung-Woo Ha, and Kyogu Lee. Phase-aware speech enhancement with deep complex u-net. 2019.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models, 2021.

Chan Jun Chun, Jung Min Moon, Geon Woo Lee, Nam Kyun Kim, and Hong Kook Kim. Deep neural network based hrtf personalization using anthropometric measurements. In *Audio Engineering Society Convention 143*. Audio Engineering Society, 2017.

Hyungjin Chung, Jeongsol Kim, Sehui Kim, and Jong Chul Ye. Parallel diffusion models of operator and image for blind inverse problems, 2022a. URL https://arxiv.org/abs/2211.10656.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2022b.

Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction, 2022c.

Hyungjin Chung, Jong Chul Ye, Peyman Milanfar, and Mauricio Delbracio. Prompt-tuning latent diffusion models for inverse problems, 2023.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints, 2024.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3): 287–314, 1994.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation, 2024. URL https://arxiv.org/abs/2306.05284.

H. Cox, R. Zeskind, and M. Owen. Robust adaptive beamforming. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(10):1365–1376, 1987. doi: 10.1109/TASSP.1987.1165054.

CPU Monkey. A14 bionic vs a18, 2024. Available at: https://www.cpu-monkey.com/en/compare_cpu-apple_a14_bionic-vs-apple_a18.

Ryan Dahl, Mohammad Norouzi, and Jonathon Shlens. Pixel recursive super resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

Jonathan Davidson, James Peters, Manoj Bhatia, Satish Kalidindi, and Sudipto Mukherjee. *Voice over IP Fundamentals (2nd Edition) (Fundamentals)*. Cisco Press, 2006. ISBN 1587052571.

Mike E Davies and Christopher J James. Source separation using single channel ica. *Signal Processing*, 87(8):1819–1832, 2007.

Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of the ACM*, 2004.

Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Demucs: Deep extractor for music sources with extra unlabeled data remixed. *arXiv preprint arXiv:1909.01174*, 2019a.

Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Music source separation in the waveform domain. *arXiv preprint arXiv:1911.13254*, 2019b.

Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain. 2020.

Antoine Deleforge, Florence Forbes, and Radu Horaud. Acoustic space learning for sound-source separation and localization on binaural manifolds. *International journal of neural systems*, 25(01):1440003, 2015.

Manik Dhar, Aditya Grover, and Stefano Ermon. Modeling sparse deviations for compressed sensing using generative models. In *International Conference on Machine Learning*, 2018.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020a.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music, 2020b. URL https://arxiv.org/abs/2005.00341.

Elio D Di Claudio and Raffaele Parisi. Waves: Weighted average of signal subspaces for robust wideband direction finding. *IEEE Transactions on Signal Processing*, 49(10):2179–2191, 2001.

Joseph Hector DiBiase. *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays*. Brown University Providence, RI, 2000.

Klaus Diepold, Marko Durkovic, and Florian Sagstetter. Hrtf measurements with recorded reference signal. In *Audio Engineering Society Convention 129*. Audio Engineering Society, 2010.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *CoRR*, abs/1605.08803, 2016. URL http://arxiv.org/abs/1605.08803.

S. Doclo and M. Moonen. Superdirective beamforming robust against microphone mismatch. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pages V–V, 2006. doi: 10.1109/ICASSP.2006.1661207.

Simon Doclo and Marc Moonen. Superdirective beamforming robust against microphone mismatch. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(2):617–631, 2007.

Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *International Conference on Learning Representations*, 2019.

Yuval Dorfan, Dani Cherkassky, and Sharon Gannot. Speaker localization and separation using incremental distributed expectation-maximization. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 1256–1260. IEEE, 2015.

Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2023.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, 2019.

Zhiyao Duan, Gautham J. Mysore, and Paris Smaragdis. Speech enhancement by online non-negative spectrogram decomposition in non-stationary noise environments. INTERSPEECH 2012, pages 594–597, 2012. ISBN 9781622767595.

Ngoc QK Duong, Emmanuel Vincent, and Rémi Gribonval. Under-determined reverberant audio source separation using a full-rank spatial covariance model. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(7):1830–1840, 2010.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Embedded. Dev kit for alexa supports multiple microphones. Available at: https://www.embedded.com/dev-kit-for-alexa-supports-multiple-microphones/.

Hakan Erdogan and Emad M Grais. Semi-blind speech-music separation using sparsity and continuity priors. In *20th International Conference on Pattern Recognition*, pages 4573–4576. IEEE, 2010.

Steven Errede. The human ear hearing, sound intensity and loudness levels. *UIUC Physics*, 406:1–33, 2002.

Sefik Emre Eskimez, Kazuhito Koishida, and Zhiyao Duan. Adversarial training for speech super-resolution. *IEEE Journal of Selected Topics in Signal Processing*, 2019.

Patrick Esser, Robin Rombach, Andreas Blattmann, and Björn Ommer. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis, 2021. URL https://arxiv.org/abs/2108.08827.

LE Audio FAQs. https://www.bluetooth.com/media/le-audio/le-audio-faqs.

Igor Fedorov, Marko Stamenovic, Carl Jensen, Li-Chia Yang, Ari Mandell, Yiming Gan, Matthew Mattina, and Paul N. Whatmough. Tinylstms: Efficient neural speech enhance-

ment for hearing aids. *Interspeech 2020*, Oct 2020. doi: 10.21437/interspeech.2020-1864. URL http://dx.doi.org/10.21437/Interspeech.2020-1864.

Andrea Ferlini, Alessandro Montanari, Chulhong Min, Hongwei Li, Ugo Sassi, and Fahim Kawsar. In-ear ppg for vital signs. *IEEE Pervasive Computing*, pages 1–10, 2021. doi: 10.1109/MPRV.2021.3121171.

Cédric Févotte, Emmanuel Vincent, and Alexey Ozerov. Single-channel audio source separation with nmf: divergences, constraints and algorithms. In *Audio Source Separation*, pages 1–24. Springer, 2018.

Maurice Frank and Maximilian Ilse. Problems using deep generative models for probabilistic audio source separation. In *"I Can't Believe It's Not Better!" NeurIPS 2020 workshop*, 2020.

O.L. Frost. An algorithm for linearly constrained adaptive array processing. *Proceedings of the IEEE*, 60(8):926–935, 1972a. doi: 10.1109/PROC.1972.8817.

Otis Lamont Frost. An algorithm for linearly constrained adaptive array processing. *Proceedings of the IEEE*, 60(8):926–935, 1972b.

Szu-Wei Fu, Chien-Feng Liao, Yu Tsao, and Shou-De Lin. Metricgan: Generative adversarial networks based black-box metric scores optimization for speech enhancement. 2019.

Yossi Gandelsman, Assaf Shocher, and Michal Irani. Double-dip": Unsupervised image decomposition via coupled deep-image-priors. In *The IEEE Conference on Computer Vision and Pattern Recognition*, volume 6, page 2, 2019.

S. Gannot, D. Burshtein, and E. Weinstein. Signal enhancement using beamforming and nonstationarity with applications to speech. *IEEE Transactions on Signal Processing*, 49 (8):1614–1626, 2001. doi: 10.1109/78.934132.

Bill Gardner, Keith Martin, et al. Hrft measurements of a kemar dummy-head microphone. 1994.

Alexandra Garfinkle. https://finance.yahoo.com/news/amazon-has-sold-more-than-500-million-alexa-enabled-devices-drops-4-new-echo-products-140013808.html, 2023.

John Garofalo, David Graff, Doug Paul, and David Pallett. Csr-i (wsj0) complete. *Linguistic Data Consortium, Philadelphia*, 2007.

Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741, 1984.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

Francois G. Germain, Qifeng Chen, and Vladlen Koltun. Speech denoising with deep feature losses. 2018.

Harshvardhan GM, Mahendra Kumar Gourisaria, Manjusha Pandey, and Siddharth Swarup Rautaray. A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, 38:100285, 2020. ISSN 1574-0137. doi: https://doi.org/10.

1016/j.cosrev.2020.100285. URL https://www.sciencedirect.com/science/article/pii/S1574013720303853.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

Google. Mediapipe, 2023. URL https://github.com/google/mediapipe. Accessed on: June 1, 2023.

Google Meet. meet.google.com.

Ivan Grishchenko, Artsiom Ablavatski, Yury Kartynnik, Karthik Raveendran, and Matthias Grundmann. Attention mesh: High-fidelity face mesh prediction in real-time, 2020.

François Grondin and James Glass. Multiple sound source localization with svd-phat. *arXiv preprint arXiv:1906.11913*, 2019.

Aditya Grover and Stefano Ermon. Uncertainty autoencoders: Learning compressed representations via variational information maximization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

J. Grythe. Beamforming algorithms-beamformers. 2015.

Rongzhi Gu, Shi-Xiong Zhang, Lianwu Chen, Yong Xu, Meng Yu, Dan Su, Yuexian Zou, and Dong Yu. Enhancing end-to-end multi-channel speech separation via spatial feature learning. *arXiv preprint arXiv:2003.03927*, 2020.

Aoqi Guo, Sichong Qian, Baoxiang Li, and Dazhi Gao. Dual-path transformer based neural beamformer for target speech extraction. *arXiv preprint arXiv:2308.15990*, 2023.

Gaëtan Hadjeres, François Pachet, and Frank Nielsen. Deepbach: a steerable model for bach chorales generation. In *International Conference on Machine Learning*, 2017.

Tavi Halperin, Ariel Ephrat, and Yedid Hoshen. Neural separation of observed and unobserved distributions. *arXiv preprint arXiv:1811.12739*, 2018.

Tavi Halperin, Ariel Ephrat, and Yedid Hoshen. Neural separation of observed and unobserved distributions. In *International Conference on Machine Learning*, 2019.

Cong Han, Yi Luo, and Nima Mesgarani. Real-time binaural speech separation with preserved spatial cues. *arXiv preprint arXiv:2002.06637*, 2020a.

Cong Han, Yi Luo, and Nima Mesgarani. Real-time binaural speech separation with preserved spatial cues. 2020b.

Per Christian Hansen. *Discrete inverse problems: insight and algorithms*. SIAM, 2010.

Simon Haykin and Zhe Chen. The cocktail party problem. *Neural computation*, 17(9): 1875–1902, 2005.

Weipeng He, Petr Motlicek, and Jean-Marc Odobez. Deep neural networks for multiple speaker detection and localization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 74–79. IEEE, 2018.

John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 31–35. IEEE, 2016.

T. Herzke, H. Kayser, F. Loshaj, G. Grimm, and V. Hohmann. Open signal processing software platform for hearing aid research (openmha). 2017.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018. URL https://arxiv.org/abs/1706.08500.

Jahn Heymann, Lukas Drude, and Reinhold Haeb-Umbach. Neural network based spectral mask estimation for acoustic beamforming. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 196–200. IEEE, 2016.

Takuya Higuchi, Keisuke Kinoshita, Marc Delcroix, Katerina Zmolíková, and Tomohiro Nakatani. Deep clustering-based beamforming for separation with unknown number of sources. In *Interspeech*, pages 1183–1187, 2017.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design, 2019. URL https://arxiv.org/abs/1902.00275.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

Maryam Hosseini, Luca Celotti, and Éric Plourde. Speaker-independent brain enhanced speech denoising. In *ICASSP 2021*, pages 1310–1314. doi: 10.1109/ICASSP39728.2021.9414969.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. 2017.

Hongmei Hu, Lin Zhou, Jie Zhang, Hao Ma, and Zhenyang Wu. Head related transfer function personalization based on multiple regression analysis. In *2006 International conference on computational intelligence and security*, volume 2, pages 1829–1832. IEEE, 2006.

Po-Sen Huang, Scott Deeann Chen, Paris Smaragdis, and Mark Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 57–60. IEEE, 2012.

Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *International Symposium on Music Information Retrieval*, pages 477–482, 2014.

Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Joint optimization of masks and deep recurrent neural networks for monaural source separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(12):2136–2147, 2015.

Tomi Huttunen, Eira T Seppälä, Ole Kirkeby, Asta Kärkkäinen, and Leo Kärkkäinen. Simulation of the transfer function for a head-and-torso model over the entire audible frequency range. *Journal of Computational Acoustics*, 15(04):429–448, 2007.

Tomi Huttunen, Antti Vanne, Stine Harder, Rasmus Reinhold Paulsen, Sam King, Lee Perry-Smith, and Leo Kärkkäinen. Rapid generation of personalized hrtfs. In *Audio Engineering Society Conference: 55th International Conference: Spatial Audio*. Audio Engineering Society, 2014.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL http://jmlr.org/papers/v6/hyvarinen05a.html.

Imagen-Team-Google, :, Jason Baldridge, Jakob Bauer, Mukul Bhutani, Nicole Brichtova, Andrew Bunner, Lluis Castrejon, Kelvin Chan, Yichang Chen, Sander Dieleman, Yuqing Du, Zach Eaton-Rosen, Hongliang Fei, Nando de Freitas, Yilin Gao, Evgeny Gladchenko, Sergio Gómez Colmenarejo, Mandy Guo, Alex Haig, Will Hawkins, Hexiang Hu, Huilian Huang, Tobenna Peter Igwe, Christos Kaplanis, Siavash Khodadadeh, Yelin Kim, Ksenia Konyushkova, Karol Langner, Eric Lau, Rory Lawton, Shixin Luo, Soňa Mokrá, Henna Nandwani, Yasumasa Onoe, Aäron van den Oord, Zarana Parekh, Jordi Pont-Tuset, Hang Qi, Rui Qian, Deepak Ramachandran, Poorva Rane, Abdullah Rashwan, Ali Razavi, Robert Riachi, Hansa Srinivasan, Srivatsan Srinivasan, Robin Strudel, Benigno Uria, Oliver Wang, Su Wang, Austin Waters, Chris Wolff, Auriel Wright, Zhisheng Xiao, Hao Xiong, Keyang Xu, Marc van Zee, Junlin Zhang, Katie Zhang, Wenlei Zhou, Konrad Zolna, Ola Aboubakar, Canfer Akbulut, Oscar Akerlund, Isabela Albuquerque, Nina Anderson, Marco Andreetto, Lora Aroyo, Ben Bariach, David Barker, Sherry Ben, Dana Berman, Courtney Biles, Irina Blok, Pankil Botadra, Jenny Brennan, Karla Brown, John Buckley, Rudy Bunel, Elie Bursztein, Christina Butterfield, Ben Caine, Viral Carpenter, Norman Casagrande, Ming-Wei Chang, Solomon Chang, Shamik Chaudhuri, Tony Chen, John Choi, Dmitry Churbanau, Nathan Clement, Matan Cohen, Forrester Cole, Mikhail Dektiarev, Vincent Du, Praneet Dutta, Tom Eccles, Ndidi Elue, Ashley Feden, Shlomi Fruchter, Frankie Garcia, Roopal Garg, Weina Ge, Ahmed Ghazy, Bryant Gipson, Andrew Goodman, Dawid Górny, Sven Gowal, Khyatti Gupta, Yoni Halpern, Yena Han, Susan Hao, Jamie Hayes, Jonathan Heek, Amir Hertz, Ed Hirst, Emiel Hoogeboom, Tingbo Hou, Heidi Howard, Mohamed Ibrahim, Dirichi Ike-Njoku, Joana Iljazi, Vlad Ionescu, William Isaac, Reena Jana, Gemma Jennings, Donovon Jenson, Xuhui Jia, Kerry Jones, Xiaoen Ju, Ivana Kajic, Christos Kaplanis, Burcu Karagol Ayan, Jacob Kelly, Suraj Kothawade, Christina Kouridi, Ira Ktena, Jolanda Kumakaw, Dana Kurniawan, Dmitry Lagun, Lily Lavitas, Jason Lee, Tao Li, Marco Liang, Maggie Li-Calis, Yuchi Liu, Javier Lopez Alberca, Matthieu Kim Lorrain, Peggy Lu, Kristian Lum, Yukun Ma, Chase Malik, John Mellor, Thomas Mensink, Inbar Mosseri, Tom Murray, Aida Nematzadeh,

Paul Nicholas, Signe Nørly, João Gabriel Oliveira, Guillermo Ortiz-Jimenez, Michela Paganini, Tom Le Paine, Roni Paiss, Alicia Parrish, Anne Peckham, Vikas Peswani, Igor Petrovski, Tobias Pfaff, Alex Pirozhenko, Ryan Poplin, Utsav Prabhu, Yuan Qi, Matthew Rahtz, Cyrus Rashtchian, Charvi Rastogi, Amit Raul, Ali Razavi, Sylvestre-Alvise Rebuffi, Susanna Ricco, Felix Riedel, Dirk Robinson, Pankaj Rohatgi, Bill Rosgen, Sarah Rumbley, Moonkyung Ryu, Anthony Salgado, Tim Salimans, Sahil Singla, Florian Schroff, Candice Schumann, Tanmay Shah, Eleni Shaw, Gregory Shaw, Brendan Shillingford, Kaushik Shivakumar, Dennis Shtatnov, Zach Singer, Evgeny Sluzhaev, Valerii Sokolov, Thibault Sottiaux, Florian Stimberg, Brad Stone, David Stutz, Yu-Chuan Su, Eric Tabellion, Shuai Tang, David Tao, Kurt Thomas, Gregory Thornton, Andeep Toor, Cristian Udrescu, Aayush Upadhyay, Cristina Vasconcelos, Alex Vasiloff, Andrey Voynov, Amanda Walker, Luyu Wang, Miaosen Wang, Simon Wang, Stanley Wang, Qifei Wang, Yuxiao Wang, Ágoston Weisz, Olivia Wiles, Chenxia Wu, Xingyu Federico Xu, Andrew Xue, Jianbo Yang, Luo Yu, Mete Yurtoglu, Ali Zand, Han Zhang, Jiageng Zhang, Catherine Zhao, Adilet Zhaxybay, Miao Zhou, Shengqi Zhu, Zhenkai Zhu, Dawn Bloxwich, Mahyar Bordbar, Luis C. Cobo, Eli Collins, Shengyang Dai, Tulsee Doshi, Anca Dragan, Douglas Eck, Demis Hassabis, Sissie Hsiao, Tom Hume, Koray Kavukcuoglu, Helen King, Jack Krawczyk, Yeqing Li, Kathy Meier-Hellstern, Andras Orban, Yury Pinsky, Amar Subramanya, Oriol Vinyals, Ting Yu, and Yori Zwols. Imagen 3, 2024. URL https://arxiv.org/abs/2408.07009.

International Telecommunication Union. *Series G: Transmission Systems and Media, Digital Systems and Networks*, 2003.

InvenSense. Microphone array beamforming. Technical Report AN-1140-00, InvenSense Inc., 1745 Technology Drive, San Jose, CA 95110 U.S.A, December 2013.

InvenSense. Microphone array beamforming. https://invensense.tdk.com/wp-content/uploads/2015/02/microphone-array-beamforming.pdf, 2015.

Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey. Single-

channel multi-speaker separation using deep clustering. *arXiv preprint arXiv:1607.02173*, 2016.

Kousuke Itakura, Yoshiaki Bando, Eita Nakamura, Katsutoshi Itoyama, Kazuyoshi Yoshii, and Tatsuya Kawahara. Bayesian multichannel audio source separation based on integrated source and spatial models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(4):831–846, 2018.

Yuki Ito, Tomohiko Nakamura, Shoichi Koyama, and Hiroshi Saruwatari. Head-related transfer function interpolation from spatially sparse measurements using autoencoder with source position conditioning, 2022.

Andreas Jansson, Eric Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde. Singing voice separation with deep u-net convolutional networks. 2017.

Nikhil Javeri, Prabal Bijoy Dutta, Kaushik Sunder, and Kapil Jain. Predicting personalized head related transfer functions using acoustic scattering neural networks. In *Audio Engineering Society Convention 153*. Audio Engineering Society, 2022.

Vivek Jayaram and John Thickstun. Source separation with deep generative priors. *arXiv preprint arXiv:2002.07942*, 2020.

Vivek Jayaram and John Thickstun. Parallel and flexible sampling from autoregressive models via langevin dynamics. In *International Conference on Machine Learning*, 2021.

Vivek Jayaram, Ira Kemelmacher-Shlizerman, Steven M. Seitz, and John Thickstun. Constrained diffusion implicit models, 2024. URL https://arxiv.org/abs/2411.00359.

Lloyd A Jeffress. A place theory of sound localization. *Journal of comparative and physiological psychology*, 41(1):35, 1948.

Teerapat Jenrungrot, Vivek Jayaram, Steve Seitz, and Ira Kemelmacher-Shlizerman. The cone of silence: Speech separation by localization. 2020.

Ge Jian and Shi Jian-ren. Computer simulation model for room diffuse sound field. *Journal of Zhejiang University-SCIENCE A*, 1(4):402–407, 2000.

Daniel Johnson, Daniel Gorelik, Ross E Mawhorter, Kyle Suver, Weiqing Gu, Steven Xing, Cody Gabriel, and Peter Sankhagowit. Latent gaussian activity propagation: using smoothness and structure to separate and localize sounds in large noisy environments. In *Advances in Neural Information Processing Systems*, pages 3465–3474, 2018.

Don H. Johnson and Dan E. Dudgeon. *Array Signal Processing: Concepts and Techniques*. Simon & Schuster, Inc., USA, 1992. ISBN 0130485136.

Don H. Johnson and Dan E. Dudgeon. Array signal processing: Concepts and techniques. 1993. URL https://api.semanticscholar.org/CorpusID:61048422.

Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models, 2021. URL https://arxiv.org/abs/2105.14080.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. URL https://arxiv.org/abs/1812.04948.

Brian FG Katz. Boundary element method calculation of individual head-related transfer function. i. rigid model calculation. *The Journal of the Acoustical Society of America*, 110 (5):2440–2448, 2001.

Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically, 2021.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models, 2022.

Fahim Kawsar, Chulhong Min, Akhil Mathur, and Alessandro Montanari. Earables for personal-scale behavior analytics. *IEEE Pervasive Computing*, 17(3):83–89, 2018. doi: 10.1109/MPRV.2018.03367740.

Mikolaj Kegler, Pierre Beckmann, and Milos Cernak. Deep speech inpainting of time-frequency masks. *arXiv preprint arXiv:1910.09058*, 2019.

F. Keyrouz. Robotic binaural localization and separation of multiple simultaneous sound sources. In *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*, pages 188–195, 2017.

Sungwon Kim, Sang-Gil Lee, Jongyoon Song, Jaehyeon Kim, and Sungroh Yoon. Flowavenet: A generative flow for raw audio. In *International Conference on Machine Learning*, 2018.

Sungwon Kim, Sang gil Lee, Jongyoon Song, Jaehyeon Kim, and Sungroh Yoon. Flowavenet : A generative flow for raw audio, 2019. URL https://arxiv.org/abs/1811.02155.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions, 2018a. URL https://arxiv.org/abs/1807.03039.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014a.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014b.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8245. doi: 10.1561/2200000056. URL http://dx.doi.org/10.1561/2200000056.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018b.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *Science*, 1983.

Bartek T Knapik, Aad W Van Der Vaart, J Harry van Zanten, et al. Bayesian inverse problems with gaussian priors. *The Annals of Statistics*, 2011.

WE Kock. Binaural localization and masking. *The Journal of the Acoustical Society of America*, 22(6):801–804, 1950.

Qiuqiang Kong, Yong Xu, Philip J. B. Jackson, Wenwu Wang, and Mark D. Plumbley. Single-channel signal separation and deconvolution with generative adversarial networks. In *International Joint Conference on Artificial Intelligence*, 2019.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021a.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis, 2021b. URL https://arxiv.org/abs/2009.09761.

Hamid Krim and Mats Viberg. Two decades of array signal processing research: the parametric approach. *IEEE signal processing magazine*, 13(4):67–94, 1996.

Krisp. www.krisp.ai.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009a.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009b.

Volodymyr Kuleshov, S Zayd Enam, and Stefano Ermon. Audio super-resolution using neural nets. In *International Conference on Learning Representations (Workshop Track)*, 2017.

Solomon Kullback. Kullback-leibler divergence, 1951.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In *Advances in Neural Information Processing Systems*, 2019.

Erno Langendijk and Adelbert Bronkhorst. Fidelity of three-dimensional-sound reproduction using a virtual auditory display. *The Journal of the Acoustical Society of America*, 107: 528–37, 02 2000. doi: 10.1121/1.428321.

Charles Laroche, Andrés Almansa, and Eva Coupete. Fast diffusion em: a diffusion model for blind inverse problems with application to deconvolution, 2023. URL https://arxiv.org/abs/2309.00287.

Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. Sdr–half-baked or well done? In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 626–630. IEEE, 2019.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017. doi: 10.1109/CVPR.2017.19.

Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, 2001.

Geon Woo Lee, Jung Hyuk Lee, Seong Ju Kim, and Hong Kook Kim. Directional audio rendering using a neural network based personalized hrtf. In *Interspeech 2019*, pages 2364–2365, 2019.

Te-Won Lee, Michael S Lewicki, Mark Girolami, and Terrence J Sejnowski. Blind source separation of more sources than mixtures using overcomplete representations. *IEEE signal processing letters*, 6(4):87–90, 1999.

Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, pages 689–694. 2004.

Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022.

Junfeng Li, Shuichi Sakamoto, Satoshi Hongo, Masato Akagi, and Yôiti Suzuki. Two-stage binaural speech enhancement with wiener filter for high-quality speech communication. *Speech Communication*, 53(5):677–689, 2011.

Song Li and Jürgen Peissig. Fast estimation of 2d individual hrtfs with arbitrary head movements. In *2017 22nd International Conference on Digital Signal Processing (DSP)*, pages 1–5, 2017. doi: 10.1109/ICDSP.2017.8096086.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization, 2024a. URL https://arxiv.org/abs/2406.11838.

Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens, 2024b. URL https://arxiv.org/abs/2410.01756.

Yuanqing Li, Shun-Ichi Amari, Andrzej Cichocki, Daniel WC Ho, and Shengli Xie. Underdetermined blind source separation based on sparse representation. *IEEE Transactions on signal processing*, 54(2):423–437, 2006.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2021.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds, 2022. URL https://arxiv.org/abs/2202.09778.

Antoine Liutkus and Roland Badeau. Generalized wiener filtering with fractional power spectrograms. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 266–270. IEEE, 2015.

Francesc Lluis, Jordi Pons, and Xavier Serra. End-to-end music source separation: is it possible in the waveform domain? *Interspeech*, 2019.

Logitech. Personalized spatial audio with head tracking, 2023. URL https://embody.co/pages/gaming-logitech. Accessed on: June 1, 2023.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed, 2021. URL https://arxiv.org/abs/2101.02388.

Xuan Luo, Xuaner (Cecilia) Zhang, Paul Yoo, Ricardo Martin-Brualla, Jason Lawrence, and Steven M. Seitz. Time-travel rephotography. *ACM Transactions on Graphics*, 40 (6):1–12, December 2021. ISSN 1557-7368. doi: 10.1145/3478513.3480485. URL http://dx.doi.org/10.1145/3478513.3480485.

Y. Luo, Z. Chen, J. R. Hershey, J. Le Roux, and N. Mesgarani. Deep clustering and conventional networks for music separation: Stronger together. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 61–65, 2017.

Yi Luo and Nima Mesgarani. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 696–700. IEEE, 2018.

Yi Luo and Nima Mesgarani. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8):1256–1266, 2019.

Yi Luo, Zhuo Chen, Nima Mesgarani, and Takuya Yoshioka. End-to-end microphone permutation and number invariant multi-channel speech separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6394–6398. IEEE, 2020a.

Yi Luo, Zhuo Chen, Nima Mesgarani, and Takuya Yoshioka. End-to-end microphone permutation and number invariant multi-channel speech separation, 2020b.

Richard Lyon. A computational model of binaural localization and separation. In *ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pages 1148–1151. IEEE, 1983.

Dong Ma, Andrea Ferlini, and Cecilia Mascolo. Oesense: Employing occlusion effect for in-ear human sensing. MobiSys '21, page 175–187, 2021a. ISBN 9781450384438. doi: 10.1145/3458864.3467680. URL https://doi.org/10.1145/3458864.3467680.

Dong Ma, Andrea Ferlini, and Cecilia Mascolo. *OESense: Employing Occlusion Effect for in-Ear Human Sensing*, page 175–187. 2021b. ISBN 9781450384438.

Ning Ma, Tobias May, and Guy J Brown. Exploiting deep neural networks and head movements for robust binaural localization of multiple sources in reverberant environments. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2444–2453, 2017.

Xuezhe Ma, Xiang Kong, Shanghang Zhang, and Eduard Hovy. Macow: Masked convolutional generative flow. In *Advances in Neural Information Processing Systems*, 2019.

Craig Macartney and Tillman Weyde. Improved speech enhancement with the wave-u-net. 2018.

Piotr Majdak, Yukio Iwaya, Thibaut Carpentier, Rozenn Nicol, Matthieu Parmentier, Agnieszka Roginska, Yôiti Suzuki, Kankji Watanabe, Hagen Wierstorf, Harald Ziegelwanger, et al. Spatially oriented format for acoustics: A data exchange format representing head-related transfer functions. In *Audio Engineering Society Convention 134*. Audio Engineering Society, 2013.

James C Makous and John C Middlebrooks. Two-dimensional sound localization by human listeners. *The journal of the Acoustical Society of America*, 87(5):2188–2200, 1990.

Michael I Mandel, Daniel P Ellis, and Tony Jebara. An em algorithm for localizing multiple sound sources in reverberant environments. In *Advances in neural information processing systems*, pages 953–960, 2007.

Michael I Mandel, Ron J Weiss, and Daniel PW Ellis. Model-based expectation-maximization source separation and localization. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):382–394, 2009.

Andrés Marafioti, Nathanaël Perraudin, Nicki Holighaus, and Piotr Majdak. A context encoder for audio inpainting. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.

Iain A McCowan, Jason Pelecanos, and Sridha Sridharan. Robust speaker recognition using microphone arrays. In *2001: A Speaker Odyssey-The Speaker Recognition Workshop*, 2001.

Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. In *International Conference on Learning Representations*, 2017.

Xiangming Meng and Yoshiyuki Kabashima. Diffusion model based posterior sampling for noisy linear inverse problems. *arXiv preprint arXiv:2211.12343*, 2022.

A. Meshram, Ravish Mehra, and Dinesh Manocha. Efficient hrtf computation using adaptive rectangular decomposition. *Proceedings of the AES International Conference*, 2014, 01 2014a.

Alok Meshram, Ravish Mehra, and Dinesh Manocha. Efficient hrtf computation using adaptive rectangular decomposition. In *Audio Engineering Society Conference: 55th International Conference: Spatial Audio*. Audio Engineering Society, 2014b.

Alok Meshram, Ravish Mehra, Hongsheng Yang, Enrique Dunn, Jan-Michael Franm, and Dinesh Manocha. P-hrtf: Efficient personalized hrtf computation for high-fidelity spatial sound. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 53–61, 2014c. doi: 10.1109/ISMAR.2014.6948409.

Andy Meyer, Dirk Döbler, Jan Hambrecht, and Manuel Matern. Acoustic mapping on three-dimensional models. pages 216–220, 06 2011. doi: 10.1145/2023607.2023645.

Michael Michelashvili and Lior Wolf. Audio denoising with deep network priors. *arXiv preprint arXiv:1904.07612*, 2019.

John C Middlebrooks. Individual differences in external-ear transfer functions reduced by scaling in frequency. *The Journal of the Acoustical Society of America*, 106(3):1480–1492, 1999a.

John C Middlebrooks. Virtual localization improved by scaling nonindividualized external-ear transfer functions in frequency. *The Journal of the Acoustical Society of America*, 106 (3):1493–1510, 1999b.

Chulhong Min, Akhil Mathur, and Fahim Kawsar. Exploring audio and kinetic sensing on earable devices. WearSys '18, page 5–10, 2018. ISBN 9781450358422. doi: 10.1145/3211960.3211970. URL https://doi.org/10.1145/3211960.3211970.

Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. 2017.

Nasser Mohammadiha, Paris Smaragdis, and Arne Leijon. Supervised and unsupervised speech enhancement using nonnegative matrix factorization. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(10):2140–2151, 2013a. ISSN 1558-7924. doi: 10.1109/tasl.2013.2270369. URL http://dx.doi.org/10.1109/TASL.2013.2270369.

Nasser Mohammadiha, Paris Smaragdis, and Arne Leijon. Supervised and unsupervised speech enhancement using nonnegative matrix factorization. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(10):2140–2151, 2013b.

A. Mohan, R. Duraiswami, D.N. Zotkin, D. DeMenthon, and L.S. Davis. Using computer vision to generate customized spatial audio. In *2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698)*, volume 3, pages III–57, 2003. doi: 10.1109/ICME.2003.1221247.

Ondrej Mokrỳ, Pavel Rajmic, and Pavel Záviška. Flexible framework for audio reconstruction. *Proceedings of the 23rd DAFx2020*, 2020.

Henrik Møller. Fundamentals of binaural technology. *Applied acoustics*, 36(3-4):171–218, 1992.

Henrik Møller, Michael Friis Sørensen, Dorte Hammershøi, and Clemen Boje Jensen. Head-related transfer functions of human subjects. *Journal of The Audio Engineering Society*, 43:300–321, 1995a.

Henrik Møller, Michael Friis Sørensen, Dorte Hammershøi, and Clemen Boje Jensen. Head-related transfer functions of human subjects. *Journal of the Audio Engineering Society*, 43(5):300–321, 1995b.

Ali Mousavi, Gautam Dasarathy, and Richard G Baraniuk. A data-driven and distributed approach to sparse signal representation and recovery. In *International Conference on Learning Representations*, 2019.

Eliya Nachmani, Yossi Adi, and Lior Wolf. Voice separation with an unknown number of multiple speakers. *arXiv preprint arXiv:2003.01531*, 2020.

Or Nadiri and Boaz Rafaely. Localization of multiple speakers under high reverberation using a spherical microphone array and the direct-path dominance test. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1494–1505, 2014.

Tomohiro Nakatani, Riki Takahashi, Tsubasa Ochiai, Keisuke Kinoshita, Rintaro Ikeshita, Marc Delcroix, and Shoko Araki. Dnn-supported mask-based convolutional beamforming for simultaneous denoising, dereverberation, and source separation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6399–6403. IEEE, 2020.

Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.

Francesco Nesta, Piergiorgio Svaizer, and Maurizio Omologo. Convolutive bss of short mixtures by ica recursively regularized across frequencies. *IEEE transactions on audio, speech, and language processing*, 19(3):624–639, 2010.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

Mohammad Nikzad, Aaron Nicolson, Yongsheng Gao, Jun Zhou, Kuldip K. Paliwal, and Fanhua Shang. Deep residual-dense lattice network for speech enhancement. 2020.

Shigeto Nishida, Masatoshi Nakamura, Akio Ikeda, and Hiroshi Shibasaki. Signal separation of background eeg and spike by using morphological filter. *Medical engineering & physics*, 21(9):601–608, 1999.

Feng Niu, Benjamin Recht, Christopher Ré, and Stephen J Wright. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2011.

Aditya Arie Nugraha, Antoine Liutkus, and Emmanuel Vincent. Multichannel audio source separation with deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(9):1652–1664, 2016.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Tom Le Paine, Pooya Khorrami, Shiyu Chang, Yang Zhang, Prajit Ramachandran, Mark A. Hasegawa-Johnson, and Thomas S. Huang. Fast wavenet generation algorithm. 2016.

Hanjie Pan, Robin Scheibler, Eric Bezzam, Ivan Dokmanić, and Martin Vetterli. Frida: Fri-based doa estimation for arbitrary array layouts. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3186–3190. IEEE, 2017.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *International Conference on Machine Learning*, 2018.

Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks, 2014. URL https://arxiv.org/abs/1301.3584.

Santiago Pascual, Antonio Bonafonte, and Joan Serrà. Segan: Speech enhancement generative adversarial network. 2017.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Despoina Pavlidi, Anthony Griffin, Matthieu Puigt, and Athanasios Mouchtaris. Real-time multiple sound source localization and counting using a circular microphone array. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(10):2193–2206, 2013.

Caslav Pavlovic, Volker Hohmann, Hendrik Kayser, Louis Wong, Tobias Herzke, S. R. Prakash, zezhang Hou, and Paul Maanen. Open portable platform for hearing aid research. *The Journal of the Acoustical Society of America*, 143(3):1738–1738, 2018. doi: 10.1121/1.5035670. URL https://doi.org/10.1121/1.5035670.

Barak A Pearlmutter and Lucas C Parra. Maximum likelihood blind source separation: A context-sensitive generalization of ica. In *Advances in Neural Information Processing Systems*, pages 613–619, 1997.

Donald A. Pierce and Daniel W. Schafer. Residuals in generalized linear models. *Journal of the American Statistical Association*, 81(396):977–986, 1986. ISSN 01621459, 1537274X. URL http://www.jstor.org/stable/2289071.

Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel wave generation in end-to-end text-to-speech. In *International Conference on Learning Representations*, 2019.

Wei Ping, Kainan Peng, Kexin Zhao, and Zhao Song. Waveflow: A compact flow-based model for raw audio. In *International Conference on Machine Learning*, 2020a.

Wei Ping, Kainan Peng, Kexin Zhao, and Zhao Song. Waveflow: A compact flow-based model for raw audio, 2020b. URL https://arxiv.org/abs/1912.01219.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022.

Jovan Powar and Alastair R. Beresford. A data sharing platform for earables research. In *Proceedings of the 1st International Workshop on Earable Computing*, EarComp'19, page 30–35, New York, NY, USA, 2019. ISBN 9781450369022. doi: 10.1145/3345615.3361139. URL https://doi.org/10.1145/3345615.3361139.

Jay Prakash, Zhijian Yang, Yu-Lin Wei, Haitham Hassanieh, and Romit Roy Choudhury. Earsense: Earphones as a teeth activity sensor. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, New

York, NY, USA, 2020. ISBN 9781450370851. doi: 10.1145/3372224.3419197. URL https://doi.org/10.1145/3372224.3419197.

Daryl Pregibon. Logistic regression diagnostics. *Annals of Statistics*, 9:705–724, 1981. URL https://api.semanticscholar.org/CorpusID:121371059.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis, 2018. URL https://arxiv.org/abs/1811.00002.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

Sound Professionals. Sp-tfb-2 – low noise in-ear binaural microphones, 2022. URL https://soundprofessionals.com/product/SP-TFB-2/. Accessed on: June 1, 2023.

Project Gutenberg. Project gutenberg. https://www.gutenberg.org/. Accessed: 2021-12-20.

Xinyuan Qian, Alessio Brutti, Maurizio Omologo, and Andrea Cavallaro. 3d audio-visual speaker tracking with an adaptive particle filter. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2896–2900. IEEE, 2017.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Boaz Rafaely. Analysis and design of spherical microphone arrays. *IEEE Transactions on speech and audio processing*, 13(1):135–143, 2004.

Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel. mir_eval: A transparent implementation of common mir metrics. In *In Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*. Citeseer, 2014.

Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The MUSDB18 corpus for music separation, 2017. URL https://doi.org/10.5281/zenodo.1117372.

Ankit Raj, Yuqi Li, and Yoram Bresler. Gan-based projector for faster recovery with convergence guarantees in linear inverse problems. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

Bhiksha Raj, Tuomas Virtanen, Sourish Chaudhuri, and Rita Singh. Non-negative matrix factorization based compensation of music for automatic speech recognition. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021. URL https://arxiv.org/abs/2102.12092.

Lord Rayleigh. Xii. on our perception of sound direction. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 13(74):214–232, 1907. doi: 10.1080/14786440709463595. URL https://doi.org/10.1080/14786440709463595.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, 2019.

RecordingNow. Do airpods, airpods pro have a microphone where? Available at: https://recordingnow.com/blog/do-airpods-have-a-microphone.

Chandan K. A. Reddy, Harishchandra Dubey, Vishak Gopal, Ross Cutler, Sebastian Braun, Hannes Gamper, Robert Aichner, and Sriram Srinivasan. Icassp 2021 deep noise suppression challenge. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6623–6627, 2021. doi: 10.1109/ICASSP39728.2021.9415105.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

Jonas Reijniers, Bart Partoens, and Herbert Peremans. Diy measurement of your personal hrtf at home: Low-cost, fast and validated. *journal of the audio engineering society*, october 2017.

Jonas Reijniers, Bart Partoens, Jan Steckel, and Herbert Peremans. Hrtf measurement by means of unsupervised head movements with respect to a single fixed speaker. *IEEE Access*, 8:92287–92300, 2020. doi: 10.1109/ACCESS.2020.2994932.

Klaus Reindl, Yuanhang Zheng, and Walter Kellermann. Speech enhancement for binaural hearing aids based on blind source separation. In *2010 4th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pages 1–6. IEEE, 2010.

Antimatter Research. Acusis s linear microphone array, 2020. Available at: https://www.digikey.com/en/product-highlight/a/antimatter-research/acusis-s-linear-microphone-array.

Alexander Richard, Peter Dodds, and Vamsi Krishna Ithapu. Deep impulse responses: Estimating and parameterizing filters with deep networks, 2022.

JH Rick Chang, Chun-Liang Li, Barnabas Poczos, BVK Vijaya Kumar, and Aswin C Sankaranarayanan. One network to solve them all–solving linear inverse problems using deep projection models. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

M. Risoud, J.-N. Hanson, F. Gauvrit, C. Renard, P.-E. Lemesre, N.-X. Bonne, and C. Vincent. Sound source localization. *European Annals of Otorhinolaryngology, Head and Neck Diseases*, 135(4):259–264, 2018. ISSN 1879-7296. doi: https://doi.org/10.

1016/j.anorl.2018.04.009. URL https://www.sciencedirect.com/science/article/pii/S187972961830067X.

Antoine Rolet, Vivien Seguy, Mathieu Blondel, and Hiroshi Sawada. Blind source separation with optimal transport non-negative matrix factorization. *EURASIP Journal on Advances in Signal Processing*, 2018(1):53, 2018.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. 2015a.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015b.

Andrew Rouditchenko, Hang Zhao, Chuang Gan, Josh McDermott, and Antonio Torralba. Self-supervised audio-visual co-segmentation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2357–2361. IEEE, 2019.

Litu Rout, Yujia Chen, Abhishek Kumar, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Beyond first-order tweedie: Solving inverse problems using latent diffusion. In *Conference on Computer Vision and Pattern Recognition*, 2024a.

Litu Rout, Negin Raoof, Giannis Daras, Constantine Caramanis, Alex Dimakis, and Sanjay Shakkottai. Solving linear inverse problems provably via posterior sampling with latent diffusion models. *Advances in Neural Information Processing Systems*, 2024b.

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R. Hershey. SDR - half-baked or well done? *CoRR*, abs/1811.02508, 2018. URL http://arxiv.org/abs/1811.02508.

Sam T Roweis. One microphone source separation. In *Advances in Neural Information Processing Systems*, pages 793–799, 2001.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. URL https://arxiv.org/abs/1409.0575.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.

Tara N Sainath, Ron J Weiss, Kevin W Wilson, Bo Li, Arun Narayanan, Ehsan Variani, Michiel Bacchiani, Izhak Shafran, Andrew Senior, Kean Chin, et al. Multichannel signal processing with deep neural networks for automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(5):965–979, 2017.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations*, 2017.

Samsung. Galaxy s5 explained: Audio. https://news.samsung.com/global/galaxy-s5-explained-audio, Jun 2014.

Narayan Sankaran, James Hillis, Marina Zannoli, and Ravish Mehra. Perceptual thresholds of spatial audio update latency in virtual auditory and audiovisual environments. *The Journal of the Acoustical Society of America*, 140(4):3008–3008, 2016.

H. Sawada, S. Araki, R. Mukai, and S. Makino. Blind extraction of dominant target sources using ica and time-frequency masking. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(6):2165–2173, 2006.

Hiroshi Sawada, Shoko Araki, and Shoji Makino. Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(3):516–527, 2010.

Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 351–355. IEEE, 2018.

Mikkel N Schmidt and Rasmus K Olsson. Single-channel speech separation using sparse non-negative matrix factorization. In *International Conference on Spoken Language Processing*, 2006.

Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.

Ben Schoon. Samsung galaxy buds 2 pro can now record 360-degree binaural audio for videos from your phone, 2023. URL https://9to5google.com/2023/01/12/samsung-buds-binaural-audio-recording. Accessed on: June 1, 2023.

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. doi: 10.1109/CVPR.2016.445.

Seed. https://wiki.seeedstudio.com/ReSpeaker_Mic_Array_v2.0/, 2016.

Viktor Seib, Benjamin Lange, and Stefan Wirtz. Mixing real and synthetic data to enhance neural network training – a review of current approaches, 2020.

Sennheiser. Earbuds that put sound first. https://en-de.sennheiser.com/newsroom/earbuds-that-put-sound-first, Mar 2020.

Nikhil Shankar, Gautam Shreedhar Bhat, and Issa Panahi. Efficient two-microphone speech enhancement using basic recurrent neural network cell for hearing and hearing aids. *The Journal of the Acoustical Society of America*, 148:389–400, 07 2020. doi: 10.1121/10.0001600.

Sheng Shen, Nirupam Roy, Junfeng Guan, Haitham Hassanieh, and Romit Roy Choudhury. Mute: Bringing iot to noise cancellation. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '18, page 282–296, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355674. doi: 10.1145/3230543.3230550. URL https://doi.org/10.1145/3230543.3230550.

Zhengshan Shi, Craig Sapp, Kumaran Arul, Jerry McBride, and Julius O Smith III. Supra: Digitizing the stanford university piano roll archive. In *International Symposium on Music Information Retrieval*, 2019.

Galit Shmueli et al. To explain or to predict? *Statistical Science*, 25(3):289–310, 2010.

Paris Smaragdis and Shrikant Venkataramani. A neural network alternative to non-negative audio models. In *International Conference on Acoustics, Speech and Signal Processing*, pages 86–90. IEEE, 2017.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023a. URL https://openreview.net/pdf?id=9_gsMA8MRKQ.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32483–32498. PMLR, 23–29 Jul 2023b. URL https://proceedings.mlr.press/v202/song23k.html.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in Neural Information Processing Systems*, 2020.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation, 2019. URL https://arxiv.org/abs/1905.07088.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pages 574–584. PMLR, 2020b.

Yang Song, Chenlin Meng, Renjie Liao, and Stefano Ermon. Nonlinear equation solving: A faster alternative to feedforward computation. *arXiv preprint arXiv:2002.03629*, 2020c.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Meet H. Soni, Neil Shah, and Hemant A. Patil. Time-frequency masking-based speech enhancement using generative adversarial network. In *ICASSP 2018*.

Sony. 360 reality audio, 2023. URL https://electronics.sony.com/360-reality-audio. Accessed on: June 1, 2023.

University of Southampton. Hrtf mesaurement system, 2003. URL https://resource.isvr.soton.ac.uk/FDAG/VAP/html/facilities.html.

Martin Spiertz and Volker Gnann. Source-filter based clustering for monaural blind source separation. In *International Conference on Digital Audio Effects*, 2009.

Rahulram Sridhar and Edgar Y Choueiri. A method for efficiently calculating head-related transfer functions directly from head scan point clouds. In *143rd Audio Engineering Society Convention 2017*, 2017.

rahulram sridhar, joseph g. tylka, and edgar choueiri. a database of head-related transfer functions and morphological measurements. *journal of the audio engineering society*, october 2017.

Rahulram Sridhar, Joseph G Tylka, and Edgar Choueiri. A database of head-related transfer functions and morphological measurements. In *Audio Engineering Society Convention 143*. Audio Engineering Society, 2017.

Statista. https://www.statista.com/statistics/677096/vr-headsets-worldwide/, 2021.

Christian J. Steinmetz, Vamsi Krishna Ithapu, and Paul Calamia. Filtered noise shaping for time domain room impulse response estimation from reverberant speech. In *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 221–225, 2021. doi: 10.1109/WASPAA52581.2021.9632680.

Daniel Stoller, Sebastian Ewert, and Simon Dixon. Adversarial semi-supervised audio source

separation applied to singing voice extraction. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2391–2395. IEEE, 2018a.

Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv preprint arXiv:1806.03185*, 2018b.

Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito. The 2018 signal separation evaluation campaign. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 293–305. Springer, 2018.

Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito. The 2018 signal separation evaluation campaign. 2018.

Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. Attention is all you need in speech separation, 2021.

Y Cem Subakan and Paris Smaragdis. Generative adversarial source separation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 26–30. IEEE, 2018.

Xingwei Sun, Risheng Xia, Junfeng Li, and Yonghong Yan. A deep learning based binaural speech enhancement approach with spatial cues preservation. In *ICASSP 2019*, pages 5766–5770, 2019. doi: 10.1109/ICASSP.2019.8683589.

Pawel Swietojanski, Arnab Ghoshal, and Steve Renals. Convolutional neural networks for distant speech recognition. *IEEE Signal Processing Letters*, 21(9):1120–1124, 2014.

Naoya Takahashi, Sudarsanam Parthasaarathy, Nabarun Goswami, and Yuki Mitsufuji. Recursive speech separation for unknown number of speakers. *arXiv preprint arXiv:1904.03065*, 2019.

Ke Tan, Xueliang Zhang, and DeLiang Wang. Real-time speech enhancement using an

efficient convolutional recurrent network for dual-microphone mobile phones in close-talk scenarios. In *ICASSP 2019*, pages 5751–5755, 2019. doi: 10.1109/ICASSP.2019.8683385.

Ke Tan, Xueliang Zhang, and Deliang Wang. Deep learning based real-time speech enhancement for dual-microphone mobile phones. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 1–1, 2021. doi: 10.1109/TASLP.2021.3082318.

Bluetooth Audio Telephony and Automotive Working Group. Hands-free profile: Bluetooth® profile specification. Technical Report v1.8, Bluetooth SIG, Apr 2020.

L. Theis and M. Bethge. Generative image modeling using spatial lstms. In *Advances in Neural Information Processing Systems*, 2015.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *International Conference on Learning Representations*, 2016a.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models, 2016b. URL https://arxiv.org/abs/1511.01844.

J. Traa and P. Smaragdis. A wrapped kalman filter for azimuthal speaker tracking. *IEEE Signal Processing Letters*, 20(12):1257–1260, 2013.

J. Traa and P. Smaragdis. Multichannel source separation and tracking with ransac and directional statistics. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):2233–2243, 2014.

Johannes Traa, Paris Smaragdis, Noah D Stein, and David Wingate. Directional nmf for joint source localization and separation. In *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–5. IEEE, 2015.

Joel A. Tropp and Anna C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007. doi: 10.1109/TIT.2007.909108.

Joel A. Tropp and Stephen J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, 2010a. doi: 10.1109/JPROC.2010.2044010.

Joel A Tropp and Stephen J Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 2010b.

Efthymios Tzinis, Shrikant Venkataramani, and Paris Smaragdis. Unsupervised deep clustering for source separation: Direct learning from mixtures using spatial information. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 81–85. IEEE, 2019.

Panagiotis Tzirakis, Anurag Kumar, and Jacob Donley. Multi-channel speech enhancement using graph neural networks. 2021.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

Benigno Uria, Iain Murray, and Hugo Larochelle. Rnade: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems*, 2013.

Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 2016.

J-M Valin, François Michaud, Jean Rouat, and Dominic Létourneau. Robust sound source localization using a microphone array on a mobile robot. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 2, pages 1228–1233. IEEE, 2003.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016a. URL https://arxiv.org/abs/1609.03499.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 2016b.

Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders, 2016c. URL https://arxiv.org/abs/1606.05328.

Aäron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. Parallel wavenet: Fast high-fidelity speech synthesis. *CoRR*, abs/1711.10433, 2017a. URL http://arxiv.org/abs/1711.10433.

Aaron van den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, 2017b.

Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. In *International Conference on Machine Learning*, 2018a.

Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. Parallel WaveNet: Fast high-fidelity speech synthesis. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3918–3926. PMLR, 10–15 Jul 2018b. URL https://proceedings.mlr.press/v80/oord18a.html.

Richard van Hoesel, Melanie Böhm, Jörg Pesch, Andrew Vandali, Rolf D Battmer, and Thomas Lenarz. Binaural speech unmasking and localization in noise with bilateral cochlear implants using envelope and fine-timing based strategies. *The Journal of the Acoustical Society of America*, 123(4):2249–2263, 2008.

Harry L Van Trees. *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons, 2002.

Barry D Van Veen and Kevin M Buckley. Beamforming: A versatile approach to spatial filtering. *IEEE assp magazine*, 5(2):4–24, 1988a.

B.D. Van Veen and K.M. Buckley. Beamforming: a versatile approach to spatial filtering. *IEEE ASSP Magazine*, 5(2):4–24, 1988b. doi: 10.1109/53.665.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. 2016a.

Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. 2016b.

Shrikant Venkataramani, Cem Subakan, and Paris Smaragdis. Neural network alternatives toconvolutive audio models for source separation. In *International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2017.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 2011.

Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE transactions on audio, speech, and language processing*, 15(3):1066–1074, 2007.

Michael Vorländer. *Auralization: fundamentals of acoustics, modelling, simulation, algorithms and acoustic virtual reality*. Springer Science & Business Media, 2007.

Bo Wahlberg, Stephen Boyd, Mariette Annergren, and Yang Wang. An admm algorithm for a class of total variation regularized estimation problems, 2012. URL https://arxiv.org/abs/1203.1828.

Julie Wall, L.J. Mcdaid, Liam Maguire, and T.M. Mcginnity. Spiking neuron models of the medial and lateral superior olive for sound localisation. pages 2641 – 2647, 07 2008. doi: 10.1109/IJCNN.2008.4634168.

Anran Wang, Maruchi Kim, Hao Zhang, and Shyamnath Gollakota. Hybrid neural networks for on-device directional hearing, AAAI 2022.

DeLiang Wang. On ideal binary mask as the computational goal of auditory scene analysis. In *Speech separation by humans and machines*, pages 181–197. Springer, 2005.

Hengkang Wang, Xu Zhang, Taihui Li, Yuxiang Wan, Tiancong Chen, and Ju Sun. Dmplug: A plug-in method for solving inverse problems with diffusion models, 2024. URL https://arxiv.org/abs/2405.16749.

Hong Wang and Mostafa Kaveh. Coherent signal-subspace processing for the detection and estimation of angles of arrival of multiple wide-band sources. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(4):823–831, 1985.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *The Eleventh International Conference on Learning Representations*, 2022.

Zheng Wang, Johnathan M Bardsley, Antti Solonen, Tiangang Cui, and Youssef M Marzouk. Bayesian inverse problems with l_1 priors: a randomize-then-optimize approach. *SIAM Journal on Scientific Computing*, 2017.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Kanji Watanabe, Yukio Iwaya, Yôiti Suzuki, Shouichi Takane, and Sojun Sato. Dataset of head-related transfer functions measured with a circular loudspeaker array. *Acoustical science and technology*, 35(3):159–165, 2014.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

Chao Weng, Dong Yu, Michael L Seltzer, and Jasha Droppo. Deep neural networks for single-channel multi-talker speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(10):1670–1679, 2015.

Felix Weninger, Hakan Erdogan, Shinji Watanabe, Emmanuel Vincent, Jonathan Roux, John R. Hershey, and Björn Schuller. Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr. LVA/ICA 2015, page 91–99. Springer-Verlag, 2015. ISBN 9783319224817. doi: 10.1007/978-3-319-22482-4_11. URL https://doi.org/10.1007/978-3-319-22482-4_11.

Elizabeth M Wenzel, Marianne Arruda, Doris J Kistler, and Frederic L Wightman. Localization using nonindividualized head-related transfer functions. *The Journal of the Acoustical Society of America*, 94(1):111–123, 1993.

Nils L. Westhausen and Bernd T. Meyer. Dual-signal transformation lstm network for real-time noise suppression, arxiv, 2020. URL https://arxiv.org/abs/2005.07551.

Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. Wham!: Extending speech separation to noisy environments. *arXiv preprint arXiv:1907.01160*, 2019.

Auke J Wiggers and Emiel Hoogeboom. Predictive sampling with forecasting autoregressive models. In *International Conference on Machine Learning*, 2020.

Frederic L Wightman and Doris J Kistler. Headphone simulation of free-field listening. i: Stimulus synthesis. *The Journal of the Acoustical Society of America*, 85(2):858–867, 1989.

Kevin W Wilson, Bhiksha Raj, Paris Smaragdis, and Ajay Divakaran. Speech denoising using nonnegative matrix factorization with priors. In *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2008.

Jiacheng Wu, Jian-Xun Wang, and Shawn C Shadden. Adding constraints to bayesian inverse problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019a.

Shanshan Wu, Alex Dimakis, Sujay Sanghavi, Felix Yu, Daniel Holtmann-Rice, Dmitry Storcheus, Afshin Rostamizadeh, and Sanjiv Kumar. Learning a compressed sensing measurement matrix via gradient unrolling. In *International Conference on Machine Learning*, 2019b.

Xiong Xiao, Shinji Watanabe, Hakan Erdogan, Liang Lu, John Hershey, Michael L Seltzer, Guoguo Chen, Yu Zhang, Michael Mandel, and Dong Yu. Deep beamforming networks for multi-channel speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5745–5749. IEEE, 2016.

Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22428–22437, 2023.

Chenglin Xu, Wei Rao, Xiong Xiao, Eng Siong Chng, and Haizhou Li. Single channel speech separation with constrained utterance level permutation invariant training using

grid lstm. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6–10. IEEE, 2018.

Yilun Xu, Yang Song, Sahaj Garg, Linyuan Gong, Rui Shu, Aditya Grover, and Stefano Ermon. Anytime sampling for autoregressive models via ordered autoencoding. In *International Conference on Learning Representations*, 2021.

Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1):7–19, 2015. doi: 10.1109/TASLP.2014.2364452.

Satoshi Yairi, Yukio Iwaya, and Yôiti Suzuki. Influence of large system latency of virtual auditory display on behavior of head movement in sound localization task. *Acta Acustica united with Acustica*, 94(6):1016–1023, 2008.

Kazuhiko Yamamoto and Takeo Igarashi. Fully perceptual-based 3d spatial sound individualization with an adaptive variational autoencoder. *ACM Transactions on Graphics (TOG)*, 36(6):1–13, 2017.

Lingxiao Yang, Shutong Ding, Yifan Cai, Jingyi Yu, Jingya Wang, and Ye Shi. Guidance with spherical gaussian constraint for conditional diffusion, 2024. URL https://arxiv.org/abs/2402.03201.

Zhijian Yang and Romit Roy Choudhury. Personalizing head related transfer functions for earables. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, SIGCOMM '21, page 137–150, New York, NY, USA, 2021a. Association for Computing Machinery. ISBN 9781450383837. doi: 10.1145/3452296.3472907. URL https://doi.org/10.1145/3452296.3472907.

Zhijian Yang and Romit Roy Choudhury. Personalizing head related transfer functions for earables. SIGCOMM '21, page 137–150, New York, NY, USA, 2021b. ISBN

9781450383837. doi: 10.1145/3452296.3472907. URL https://doi.org/10.1145/3452296.3472907.

Mariam Yiwere and Eun Joo Rhee. Distance estimation and localization of sound sources in reverberant conditions using deep neural networks. *International Journal of Applied Engineering Research*, 12(22):12384–12389, 2017.

Yeo-Sun Yoon, Lance M Kaplan, and James H McClellan. Tops: New doa estimator for wideband signals. *IEEE Transactions on Signal processing*, 54(6):1977–1989, 2006.

Takuya Yoshioka, Hakan Erdogan, Zhuo Chen, and Fil Alleva. Multi-microphone neural speech separation for far-field multi-talker speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5739–5743. IEEE, 2018.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014.

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Yongsheng Yu, Libo Zhang, Heng Fan, and Tiejian Luo. High-fidelity image inpainting with gan inversion, 2022. URL https://arxiv.org/abs/2208.11850.

Navid H Zandi, Awny M El-Mohandes, and Rong Zheng. Individualizing head-related transfer functions for binaural acoustic applications. In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 105–117. IEEE, 2022.

Pavel Záviška, Pavel Rajmic, Alexey Ozerov, and Lucas Rencker. A survey and an extensive evaluation of popular audio declipping methods. *arXiv preprint arXiv:2007.07663*, 2020.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric, 2018a. URL https://arxiv.org/abs/1801.03924.

Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4786–4794, 2018b.

Xueliang Zhang and DeLiang Wang. Deep learning based binaural speech separation in reverberant environments. *IEEE/ACM transactions on audio, speech, and language processing*, 25(5):1075–1084, 2017.

Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018c.

Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 570–586, 2018.

Manlin Zhao, Zhichao Sheng, and Yong Fang. Magnitude modeling of personalized hrtf based on ear images and anthropometric measurements. *Applied Sciences*, 12(16):8155, 2022.

Bowen Zhi, Dmitry N. Zotkin, and Ramani Duraiswami. Towards fast and convenient end-to-end hrtf personalization. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 441–445, 2022. doi: 10.1109/ICASSP43922.2022.9746315.

Hang Zhou, Ziwei Liu, Xudong Xu, Ping Luo, and Xiaogang Wang. Vision-infused deep audio inpainting. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

Yaxuan Zhu, Zehao Dou, Haoxin Zheng, Yasi Zhang, Ying Nian Wu, and Ruiqi Gao. Think twice before you act: Improving inverse problem solving with mcmc, 2024. URL https://arxiv.org/abs/2409.08551.

Michael Zibulevsky and Barak A Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neural computation*, 13(4):863–882, 2001.

Harald Ziegelwanger, Wolfgang Kreuzer, and Piotr Majdak. Mesh2hrtf: Open-source software package for the numerical calculation of head-related transfer functions. In *22nd International Congress on Sound and Vibration*, 2015.

Dmitry N Zotkin, Ramani Duraiswami, and Larry S Davis. Customizable auditory displays. Georgia Institute of Technology, 2002.

D.Y.N. Zotkin, J. Hwang, R. Duraiswaini, and L.S. Davis. Hrtf personalization using anthropometric measurements. In *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684)*, pages 157–160, 2003. doi: 10.1109/ASPAA.2003.1285855.

Appendix A

EXTENDED BASIS RESULTS

In this section, we show extended visual results from the BASIS sampling algorithm (Chapter 7).

A.1 Intermediate Samples During the Annealing Process

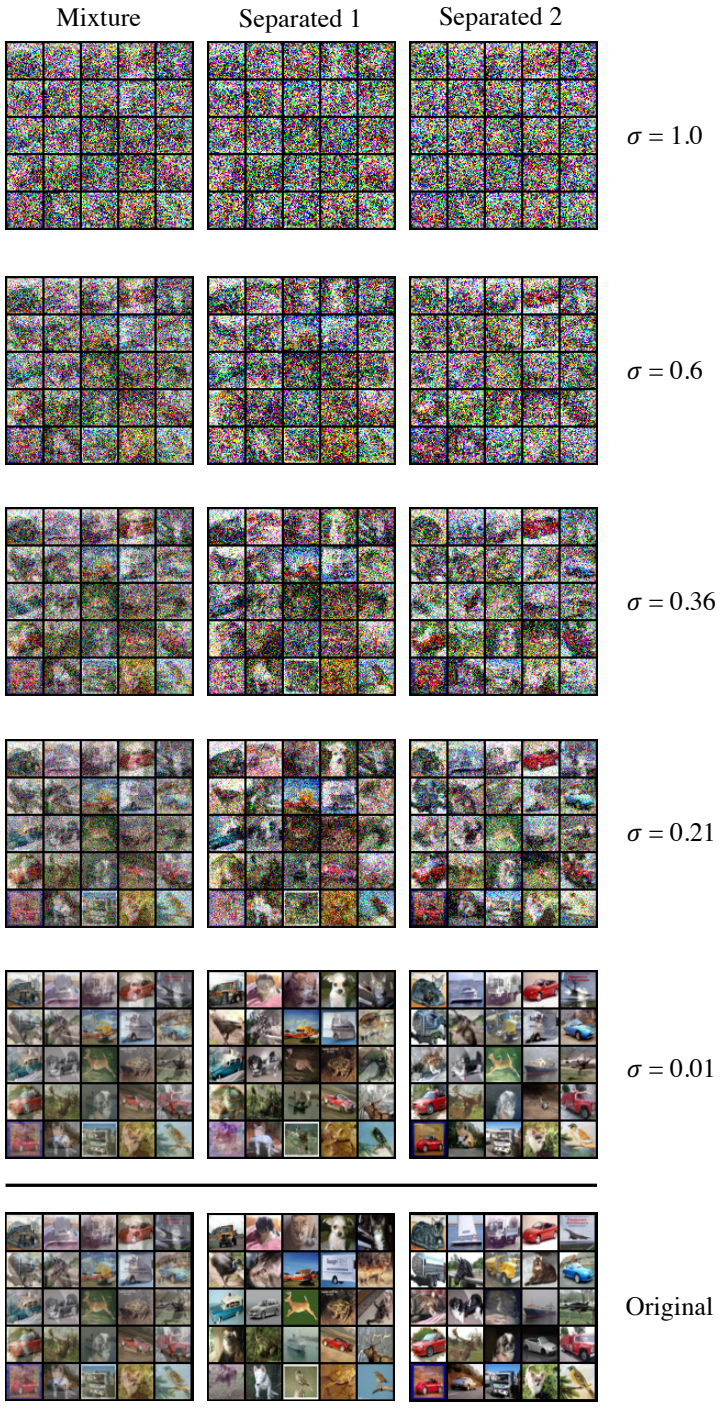


Figure A.1: Intermediate CIFAR-10 separation results taken at noise levels σ.

A.2 MNIST Separation Results Under Different Models and Sampling Procedures

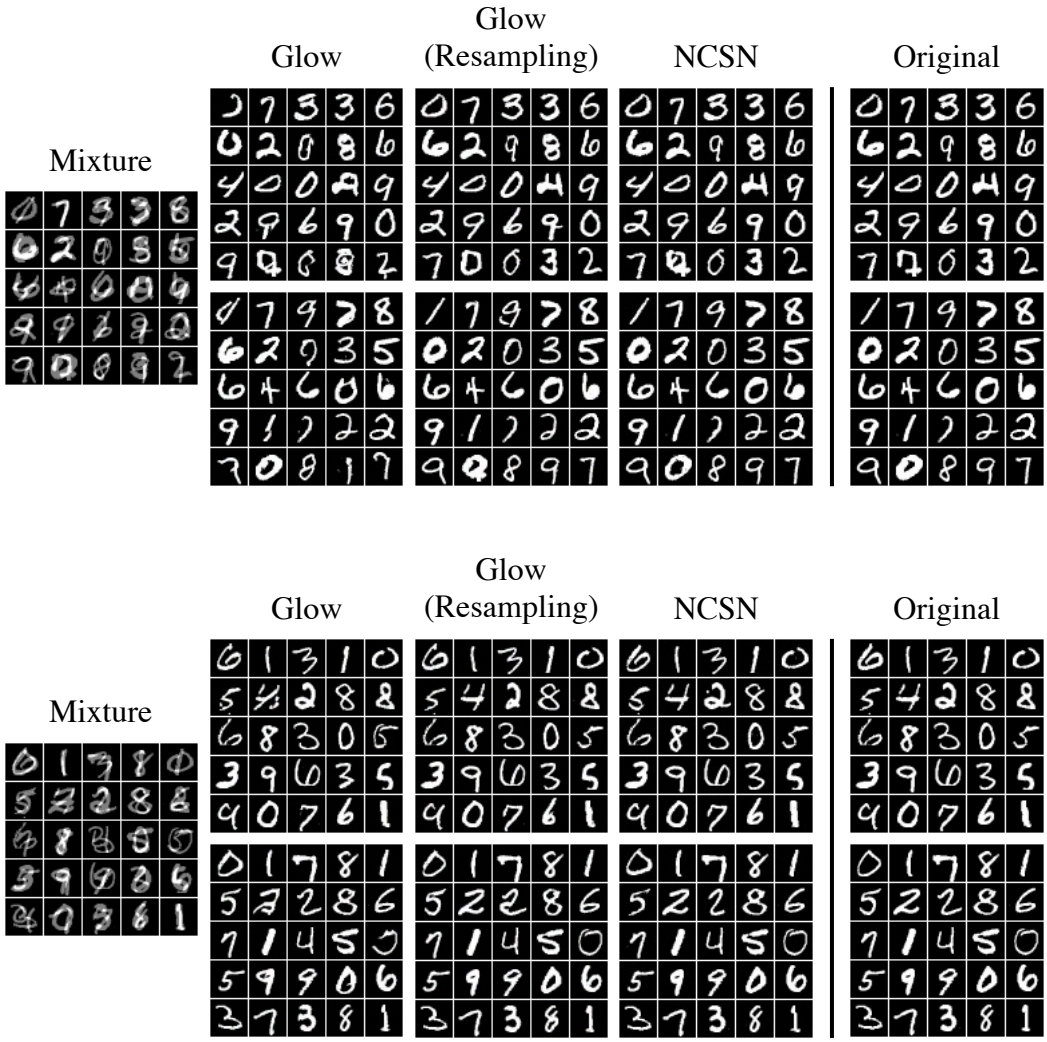


Figure A.2: Uncurated class-agnostic separation results using: (1) samples from the posterior with Glow as a prior (2) an approximate MAP estimate using the maximum over 10 samples from the posterior with Glow as a prior (3) samples from the posterior with NCSN as a prior.

A.3 Extended CIFAR-10 Separation Results

A.3.1 NCSN Prior

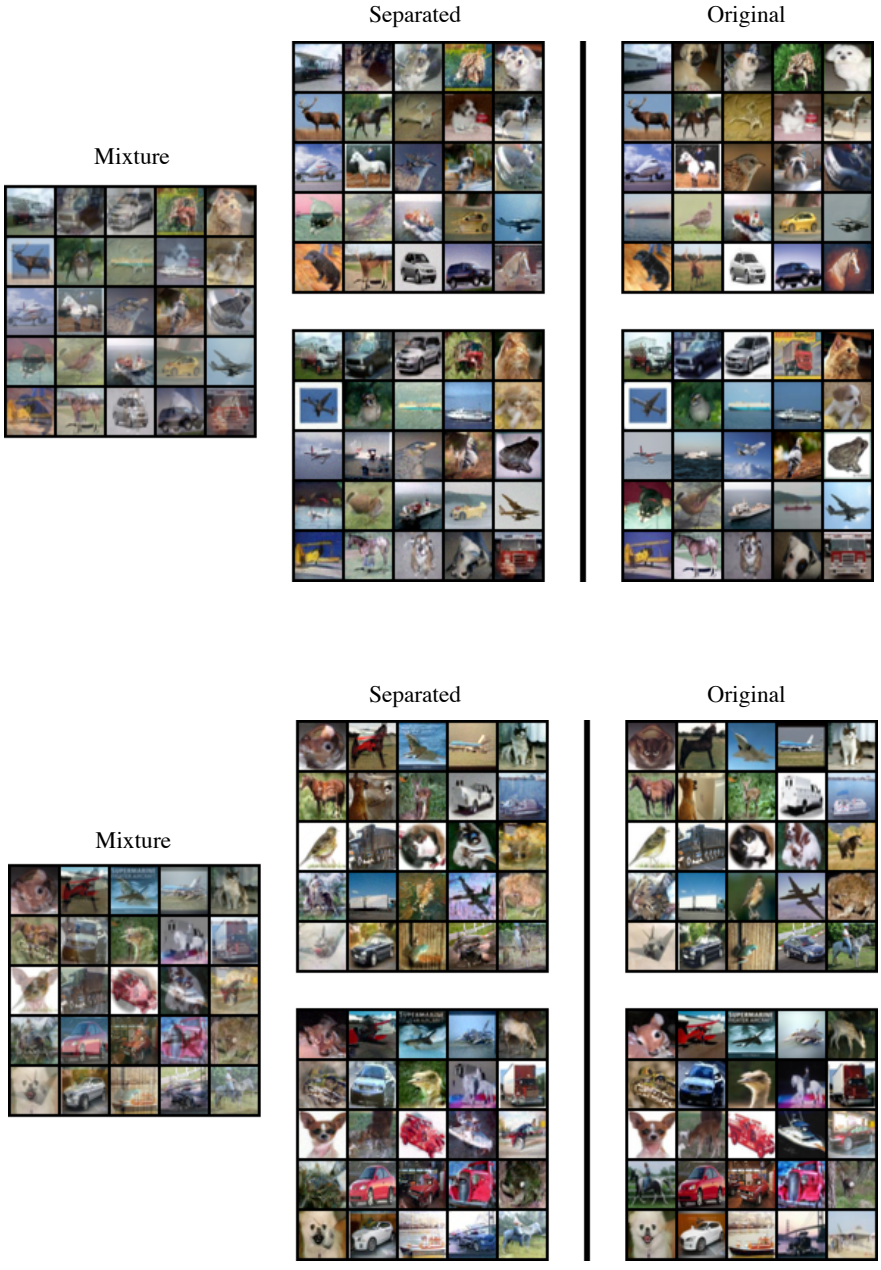


Figure A.3: Uncurated class-agnostic CIFAR-10 separation results using NCSN as a prior.

A.3.2 Glow Prior

Figure A.4: Uncurated class-agnostic CIFAR-10 separation results using Glow as a prior.

A.4 Extended CIFAR-10 Colorization Results

A.4.1 NCSN Prior

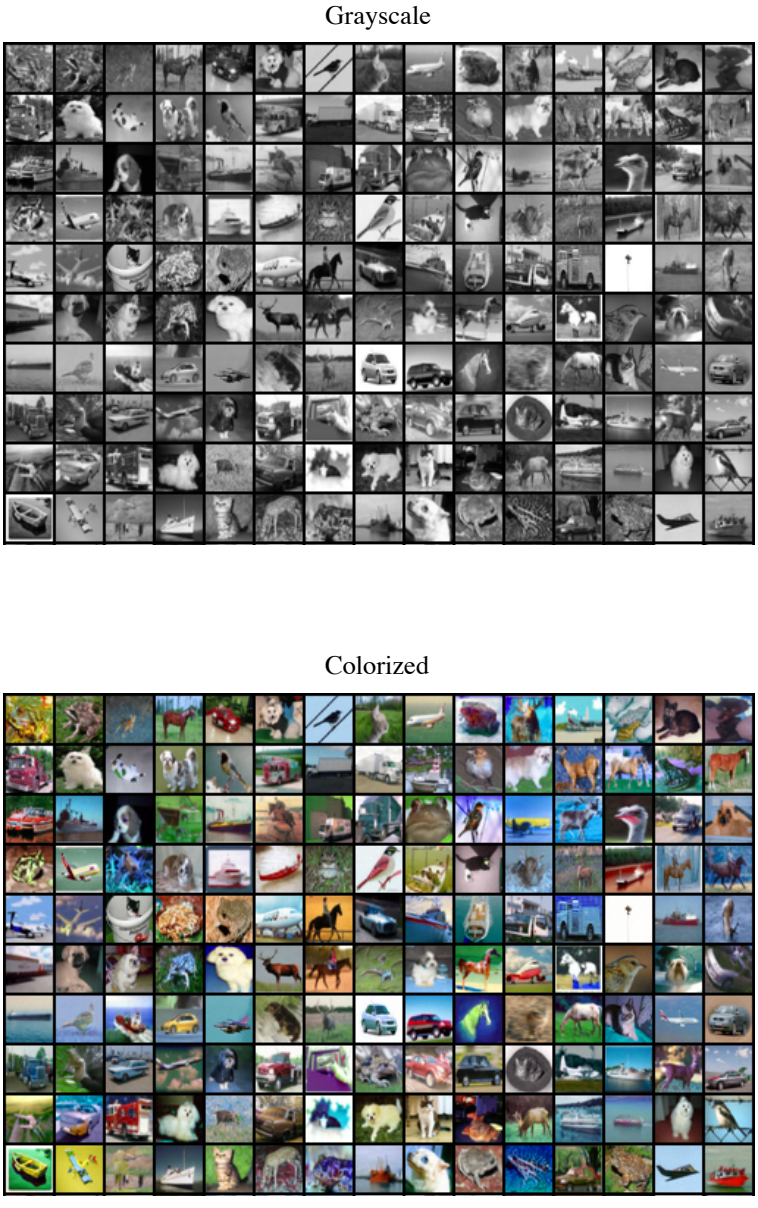


Figure A.5: Uncurated CIFAR-10 colorization results using NCSN as a prior.

A.4.2 Glow Prior

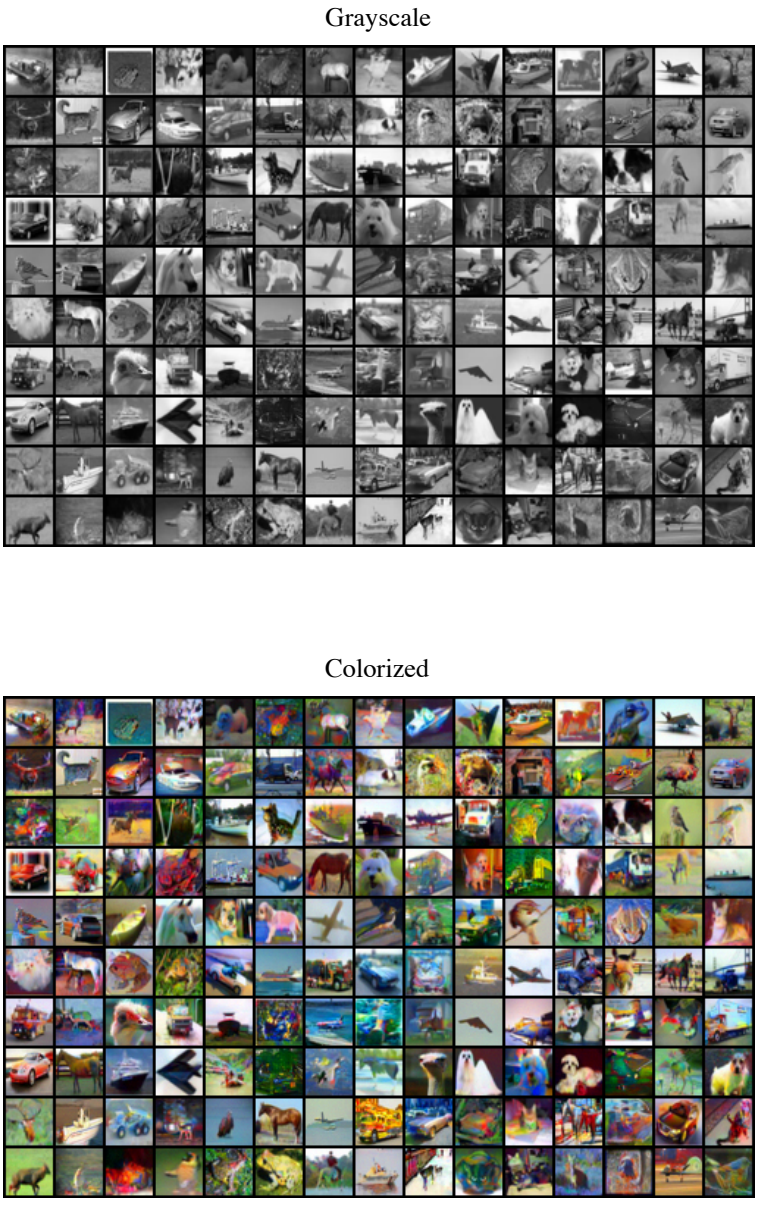


Figure A.6: Uncurated CIFAR-10 colorization results using Glow as a prior.

A.5 Extended LSUN Separation Results

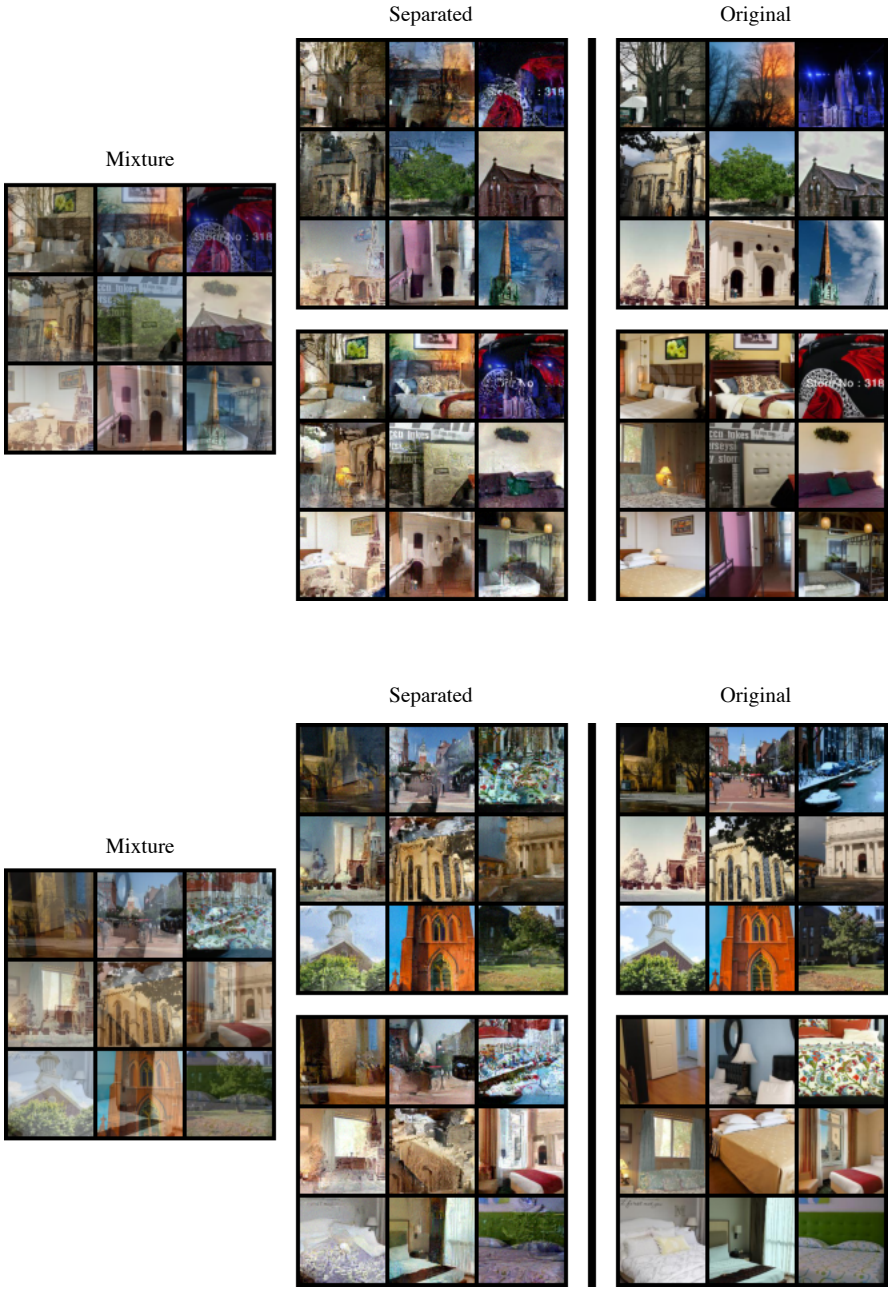


Figure A.7: Uncurated church/bedroom LSUN separation results using Glow as a prior.

Appendix B

CDIM SUPPLEMENTARY MATERIALS

B.1 Calculating $\mathbb{E}\left\|\nabla_{\mathbf{x}_{t-\delta}}\right\|$

To calculate our expected gradient magnitude, we first start with simple gradient normalization: $\eta \leftarrow 1/\left\|\nabla_{\mathbf{x}_{t-\delta}}\right\|$, which normalizes our step size by the gradient magnitude on the fly at every optimization step. We run the full CDIM algorithm on the target task with the desired number of steps T and K on images from the training set. We calculate and store each gradient magnitude $\left\|\nabla_{\mathbf{x}_{t-\delta}}\right\|$ during the optimization process at every step. Finally, we average the empirical gradient magnitudes at each step $t-\delta$ to find $\mathbb{E}\left\|\nabla_{\mathbf{x}_{t-\delta}}\right\|$ across data points and inner optimization steps k. In practice we find that very few images are required to calculate a stable value for the expected gradient magnitude. In all experiments the value was calculated by running an initial optimization on 10 images from the training set.

B.2 Additional Experimental Details

B.2.1 Task Details

We describe additional details for each inverse task used in our experiments.

Super Resolution Images are downsampled to 64×64 using bicubic downsampling with a factor of 4.

Box Inpainting A random box of size 128×128 is chosen uniformly within the image. Those pixels are masked out affected all three of the RGB channels.

Gaussian Deblur A Gaussian Kernel of size 61×61 and intensity 3 is applied to the entire image.

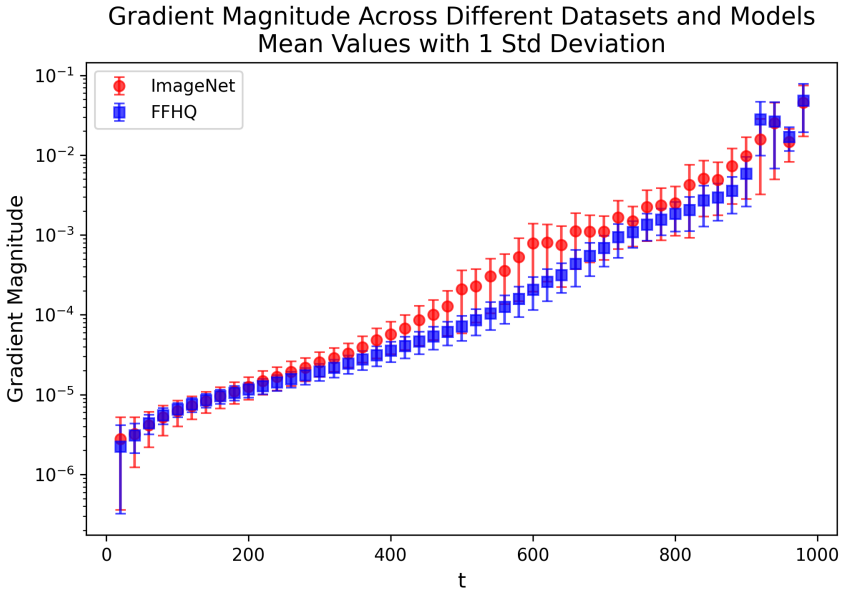


Figure B.1: A plot of $||\nabla_{\mathbf{x}_{t-\delta}}||$ for two models and datasets, ImageNet and FFHQ. In each task 100 images were used. First, note the variance in a single task/model, shown by the error bars, is small. Second, note that the variance across the two tasks/models is also small.

Random Inpainting Each pixel is masked out with probability 92% affecting all three of the RGB channels

50% Inpainting In various figures, we showcase a a 50% inpainting task where the top half of an image is masked out. This task is more challenging than box inpainting and can better illustrate differences between results.

B.2.2 Measuring Runtime

To measure wall-clock runtime, we used a single A100 and ran all the inverse problems (super-resolution, box inpainting, gaussian deblur, random inpainting) on the FFHQ dataset. We

only consider the runtime of the algorithm, without considering the python initialization time, model loading, or image io. For each task, we measured the runtime on 10 images and averaged the result to produce the final result. We note that the baseline runtimes are taken from Dou and Song (2023), where only the box inpainting task was considered. The runtime does not vary much between tasks when using CDIM, so we report our average runtime across tasks as a fair comparison metric.

B.2.3 Number of Inference Steps Ablation Studies

CDIM offers the flexibility to trade off quality for faster inference time on demand. We investigate how generation quality changes as we vary the total computational budget during inference. Recall that the total number of network passes during inference is $T'(K+1)$, where T' is the number of denoising steps and K is the number of optimization steps per denoising step. We use the random inpainting task on the FFHQ dataset with the setup described in the previous section. For this experiment we use KL optimization (Algorithm 6). The total network forward passes are varied from 200 to 20, and we show qualitative results. Notably, CDIM yields high quality samples with as few as 50 total inference steps, with quality degradations after that.

B.2.4 $\mathbf{T}'$ *vs* $\mathbf{K}$ *Trade-Off*

We consider the optimal balance between T' and K when the total number of inference steps $T'(K+1)$ is fixed. Using the random inpainting task on the FFHQ dataset with the previously described setup, we set $T'(K+1) = 200$ and analyze how PSNR, FID, and LPIPS change based on the chosen T' and K values. Results are plotted in Figure B.3. FID results consistently favor the maximum number of denoising steps T' with minimal optimization steps K. This is because FID evaluates overall distribution similarity rather than per-sample fidelity, and thus is not penalized by lower reconstruction-observation fidelity. In contrast, PSNR and LPIPS achieve optimal results with a balanced mix of denoising and optimization steps.

Input	10 Steps	20 Steps	50 Steps	200 Steps
Random Inpainting	$T' = 5$ $K = 1$	$T' = 10$ $K = 1$	$T' = 25$ $K = 1$	$T' = 50$ $K = 3$

Figure B.2: We reduce the total number of inference steps $T'(K+1)$ and visualize the results. There is almost no visible degradation until fewer than 50 total steps.

B.2.5 ImageNet Results

In Table B.3 we report FID and LPIPS for ImageNet.

B.2.6 PSNR Results

See Tables B.2 and B.3

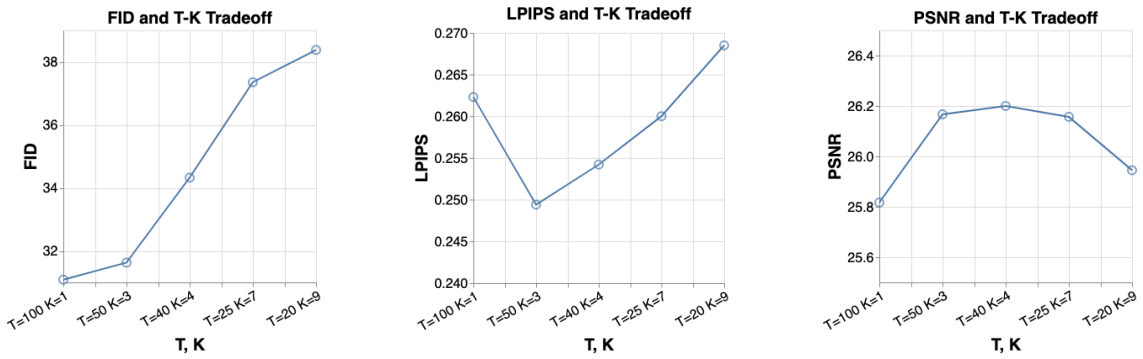


Figure B.3: We fix the total number of inference steps at 200 and evaluate different combinations of T' and K. FID always prefers more denoising steps T', while LPIPS and PSNR are best at a mix of T' and K steps.

Table B.1: Quantitative results (FID, LPIPS) of our model and existing models on various linear inverse problems on the Imagenet 256 × 256-1k validation dataset. (Lower is better)

Imagenet	**Super Resolution**		**Inpainting (box)**		**Gaussian Deblur**		**Inpainting (random)**	
Methods	FID	LPIPS	FID	LPIPS	FID	LPIPS	FID	LPIPS
CDIM - KL fast	59.10	0.398	58.75	0.311	73.74	0.480	53.91	0.364
CDIM - L2 fast	53.70	0.378	52.00	0.267	56.10	0.393	51.96	0.370
CDIM - KL	47.77	0.347	48.26	0.2348	57.72	0.390	45.86	0.331
CDIM - L2	47.45	0.339	50.31	0.251	38.69	0.347	46.20	0.332
FPS-SMC	47.30	0.316	33.24	0.212	54.21	0.403	42.77	0.328
DPS	50.66	0.337	38.82	0.262	62.72	0.444	35.87	0.303
DDRM	59.57	0.339	45.95	0.245	63.02	0.427	114.9	0.665
MCG	144.5	0.637	39.74	0.330	95.04	0.550	39.19	0.414
PnP-ADMM	97.27	0.433	78.24	0.367	100.6	0.519	114.7	0.677
Score-SDE	170.7	0.701	54.07	0.354	120.3	0.667	127.1	0.659
ADMM-TV	130.9	0.523	87.69	0.319	155.7	0.588	189.3	0.510

Table B.2: Quantitative results (PSNR) of our model and existing models on various linear inverse problems on the FFHQ 256-1k validation dataset. (Higher is better)

Imagenet	**Super Resolution**	**Inpainting (box)**	**Gaussian Deblur**	**Inpainting (random)**
Methods	PSNR	PSNR	PSNR	PSNR
CDIM - KL fast	26.94	22.84	24.8	26.38
CDIM - L2 fast	27.08	23.20	26.77	26.49
CDIM - KL	27.11	23.54	25.68	26.97
CDIM - L2	27.30	23.47	27.03	27.10
FPS-SMC	28.10	24.70	26.54	27.33
DPS	25.67	22.47	24.25	25.23
DDRM	25.36	22.24	23.36	9.19
MCG	20.05	19.97	6.72	21.57
PnP-ADMM	26.55	11.65	24.93	8.41
Score-SDE	17.62	18.51	7.21	13.52
ADMM-TV	23.86	17.81	22.37	22.03

Table B.3: Quantitative results (PSNR) of our model and existing models on various linear inverse problems on the Imagenet 256 $\times$ 256-1k validation dataset. (Higher is better)

Imagenet	**Super Resolution**	**Inpainting (box)**	**Gaussian Deblur**	**Inpainting (random)**
Methods	PSNR	PSNR	PSNR	PSNR
CDIM - KL fast	23.17	19.64	21.26	21.95
CDIM - L2 fast	23.67	19.67	22.78	22.38
CDIM - KL	23.36	19.98	22.48	22.07
CDIM - L2	23.92	20.06	23.32	22.61
FPS-SMC	24.78	22.03	23.81	24.12
DPS	23.87	18.90	21.97	22.20
DDRM	24.96	18.66	22.73	14.29
MCG	13.39	17.36	16.32	19.03
PnP-ADMM	23.75	12.70	21.81	8.39
Score-SDE	12.25	16.48	15.97	18.62
ADMM-TV	22.17	17.96	19.99	20.96

B.2.7 Extended Results

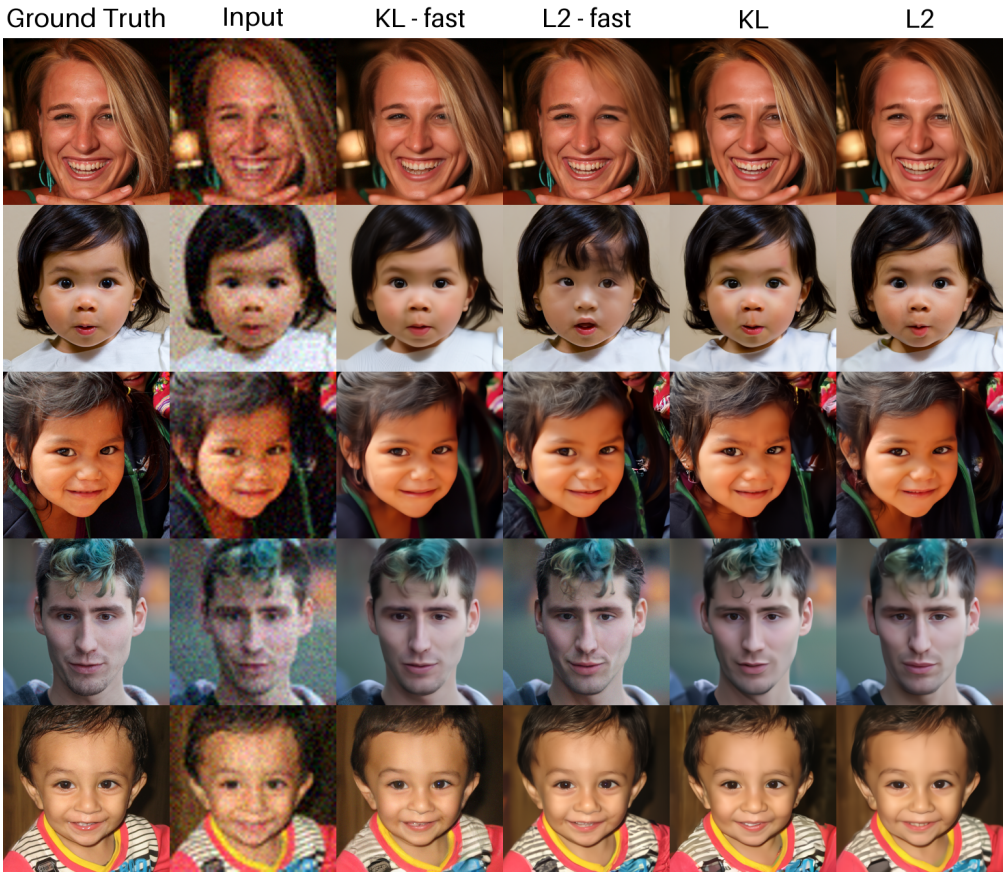


Figure B.4: FFHQ Super-resolution extended results

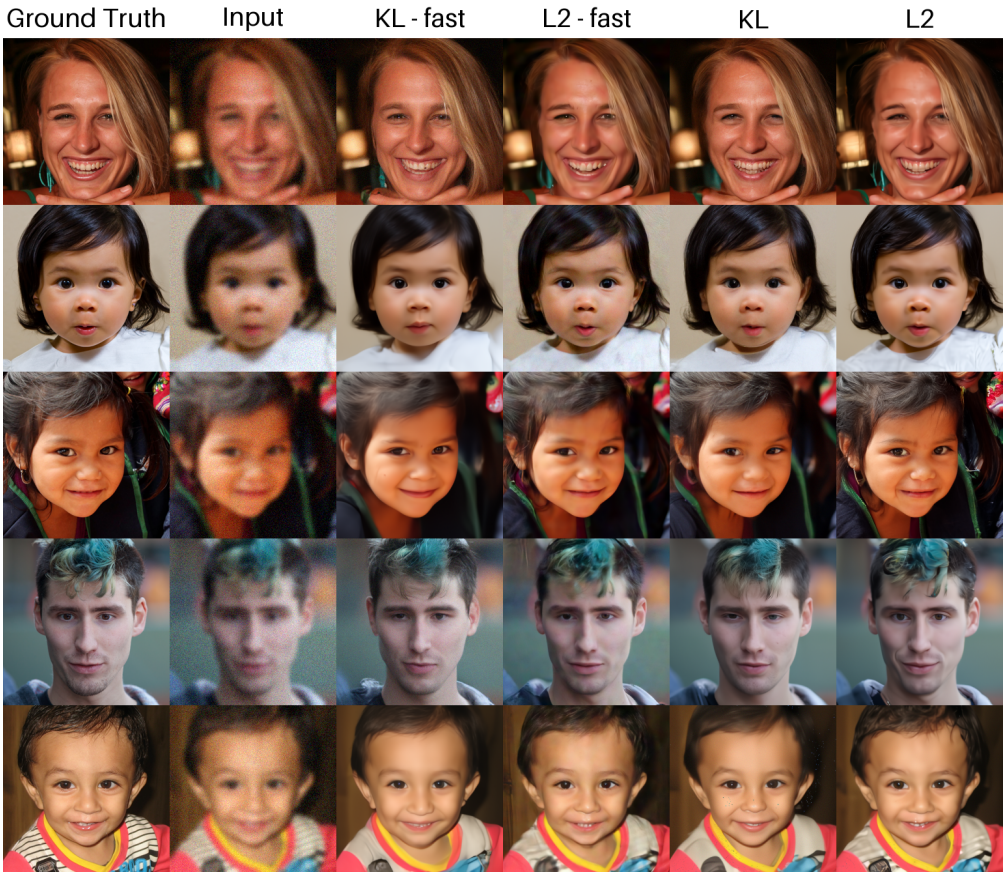


Figure B.5: FFHQ Gaussian deblur extended results

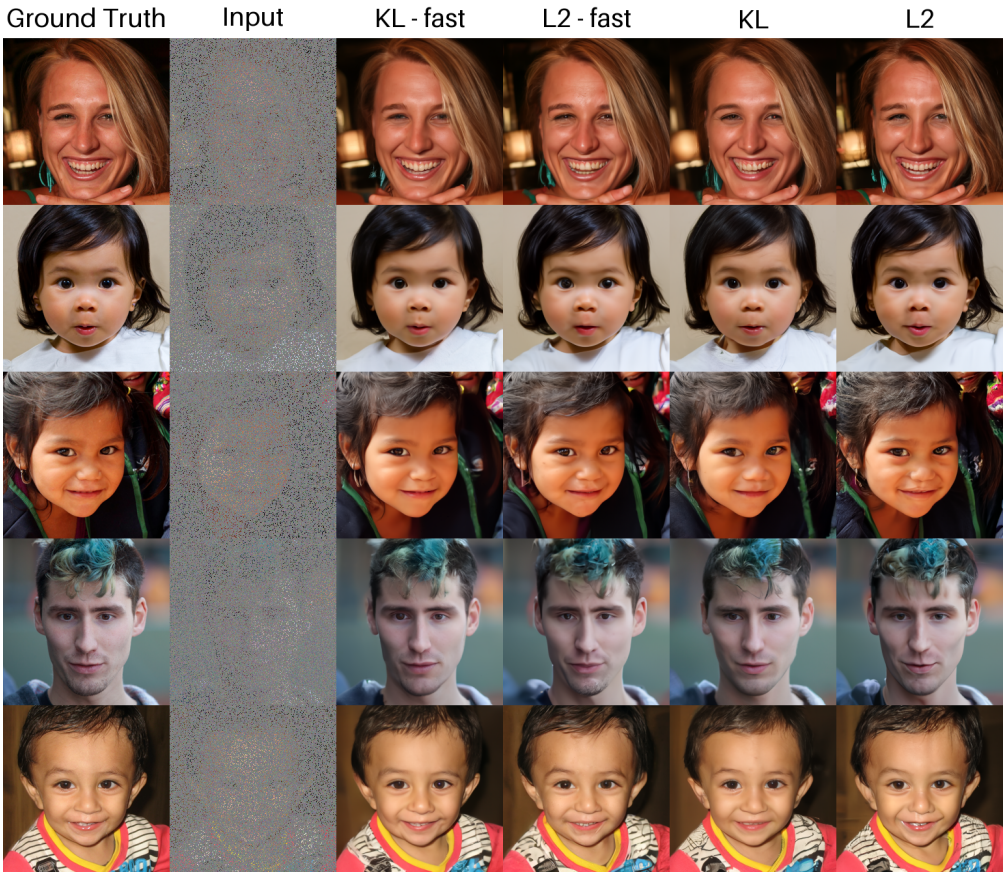


Figure B.6: FFHQ random inpainting extended results

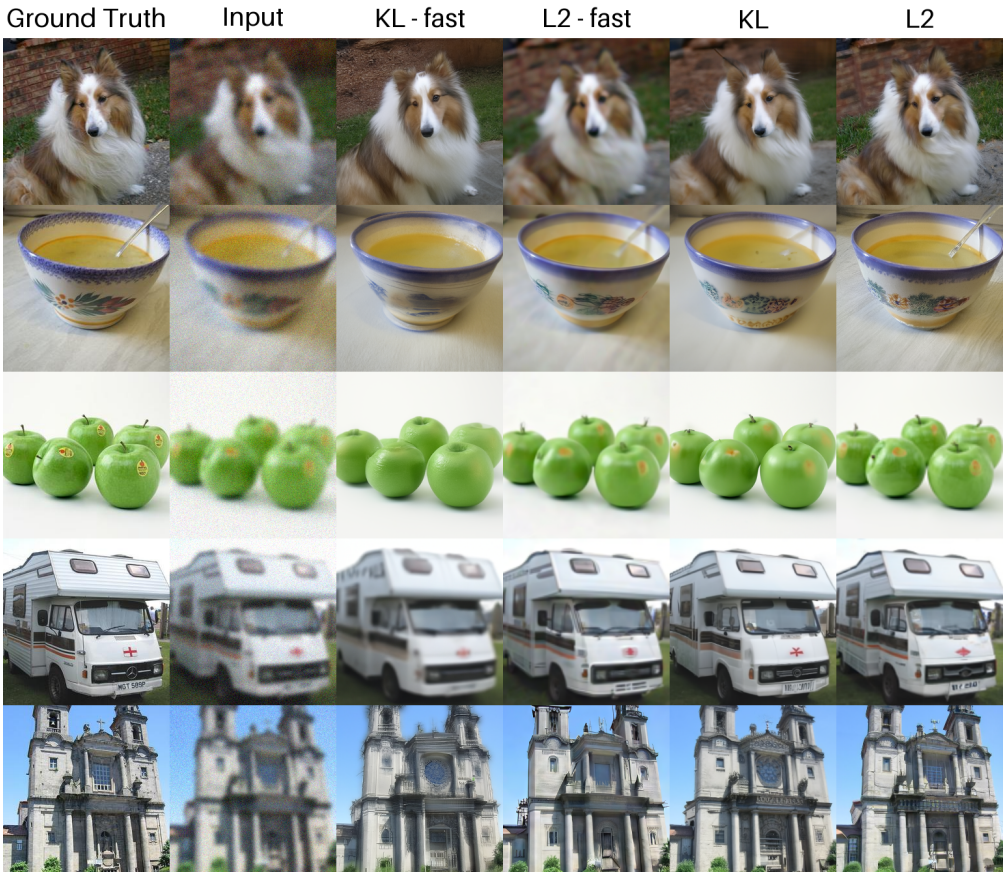


Figure B.7: ImageNet Gaussian deblur extended results

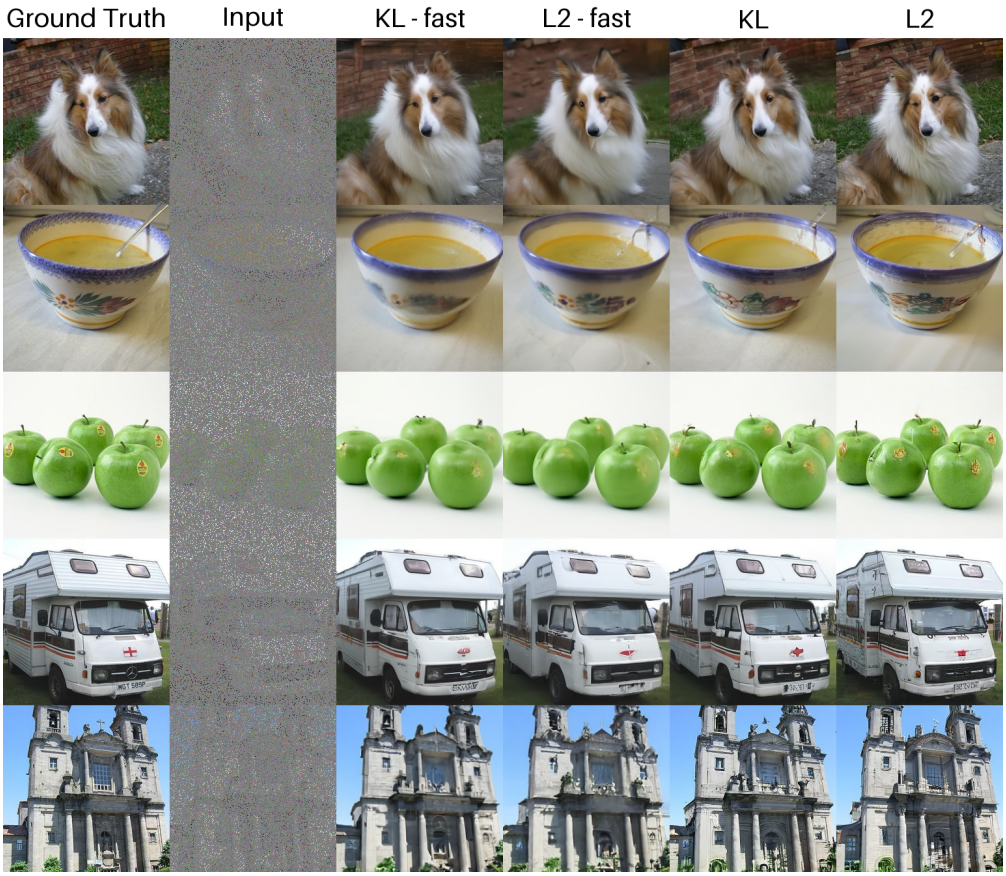


Figure B.8: ImageNet random inpainting extended results

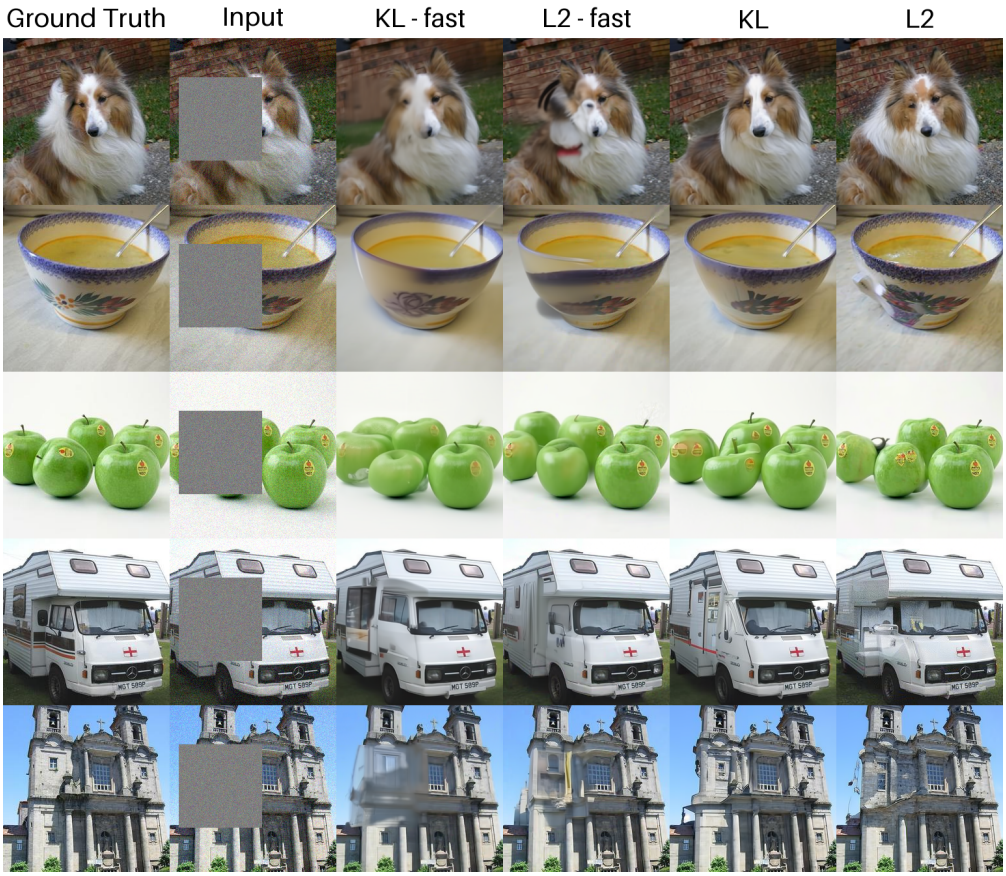


Figure B.9: ImageNet box inpainting extended results