Towards Integrated Audio-Visual Learning: From Vision-to-Audio Generation to a Unified Audio-Visual Framework

Kun Su

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

Doctor of Philosophy

University of Washington

2024

Reading Committee:
Eli Shlizerman, Chair
Linda Shapiro
Shwetak Patel

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Towards Integrated Audio-Visual Learning: From Vision-to-Audio Generation to a Unified Audio-Visual Framework

Kun Su

Chair of the Supervisory Committee:
Eli Shlizerman
Electrical and Computer Engineering

The interplay between audio and visual signals, rich in correlations across various scales, significantly impacts human perception and drives a consistent demand for audio-visual applications across fields such as video production, animation, and virtual reality. Historically, the creation and adaptation of audio content has been predominantly a manual process reliant on the expertise of Foley artists. Exploring automated systems capable of assisting with such tasks and achieving comparable proficiency suggests an intriguing prospect. In recent years, deep learning-based methodologies have shown considerable promise in handling image, video, and text data. Nonetheless, the task of integrating audio with visual inputs introduces distinct challenges that stem from the fundamental differences and complexities inherent to each modality. In particular, techniques for generating non-speech audio, such as music, object material impact sounds, natural sounds, and spatial sound effects, have received limited exploration.

Audio and visual signals can be represented in various ways, with their connections varying across different scenarios. The research domain of learning to connect and relate audio and visual signals, termed audio-visual learning, has traditionally focused on either audio-visual representation learning or on generative modeling of one modality conditioned on the other. My research traverses audio-visual learning and connects audio-visual representation

and generation from three distinct perspectives.

In the first category, my investigation focuses on **vision-to-audio generation through an intermediate representation**, which serves as a bridge to link visual and audio domains. For instance, musical notes can translate piano keystrokes into their corresponding sounds, and the rhythm of movement can connect dance videos to their accompanying music. The intermediate representation could also be derived from pre-trained deep learning models and act as semantic bridges to facilitate the transition from more general video and background music where the audio-visual relationship is largely subjective.

In the second category, my research delves into learning implicit representation through the vision-to-audio generative process. This angle seeks not only to achieve **vision-to-audio conversion** but also to **construct meaningful representations therein**. Innovations here include unsupervised models that deduce instrumental sounds from musicians' body movements and diffusion models capable of synthesizing impact sounds from visual cues of object-physical interaction. This exploration reveals the association between visual inputs and various timbre characteristics. Additionally, by mapping indoor scene geometry to room impulse responses at discrete locations, we can infer a continuous acoustic field that enables the rendering of high-fidelity audio for any emitter-listener locations and enhances the realism and immersion of auditory experiences.

Finally, in the third part of work, I proposed a **unified audio-visual framework** that seamlessly **merges representation learning and generative modeling**. This general approach enables the efficient generation of high-fidelity audio from visual stimuli while constructing robust semantic audio-visual representations. Its applications are broad, ranging from audio-visual retrieval and event classification to audio-only classification, paving the way for more immersive and contextually rich audio-visual experiences.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my advisor, Dr. Eli Shlizerman, for his invaluable guidance throughout my research journey. His insights have been pivotal in shaping my academic and industrial contributions, continually steering my research direction and refining my understanding. Beyond academic mentorship, Dr. Shlizerman's advice on life and career choices has been profoundly influential.

My sincere appreciation also goes to my supervisory committee members, Dr. Linda Shapiro, Dr. Shwetak Patel, and Dr. Vikram Iyer, for their dedicated oversight and support.

I am thankful for the camaraderie and encouragement from my colleagues and friends, Xiulong Liu, Yang Zheng, Jingyuan Li, Mingfei Chen, Jinlin Xiang, Jimin Kim, Trung Le, Rahul Biswas, Yan Jiang, Lyutianyang Zhang, Ricky Zhang, Yujie Zhou, Xiangyu Gao, Hao Yin, Ran Wei, Mingchen Li, Jiarou Quan, Yusu Chen, Sheng Yao, among others, whose presence has enriched my PhD journey.

My industry internships have been instrumental in broadening my perspective. At Adobe Research, Dr. Xue Bai provided invaluable supervision and support. My internship at the MIT-IBM AI Lab was enriched by mentorship from Dr. Chuang Gan and collaboration with Dr. Kaizhi Qian, offering critical research insights. Google Research offered a nurturing environment under the mentorship of Judith Yue Li and Dima Kuzmin, alongside the opportunity for a full-time position. I am grateful to all my collaborators at Google, including Qingqing Huang, Joonseok Lee, Chris Donahue, Fei Sha, Aren Jansen, Yu Wang, Mauro Verzetti, and Timo Denk, for their support.

Reflecting on the past five years, despite facing numerous research setbacks and

moments of self-doubt, the journey through my Ph.D. at the University of Washington has forged me into a significantly stronger individual. Armed with this resilience, I look forward to embracing future challenges with confidence.

DEDICATION

Dedicated with all my love to my dear Yi Yang, to my parents, my sister, and my entire family, who have been my unwavering support and source of strength.

Chapter 1

INTRODUCTION

1.1 Background

Vision and audio are two fundamental facets of human perception, each playing a critical role in how we understand and interact with the world around us. The integration of visual and auditory information enables a more comprehensive understanding of our environment, enhancing our ability to communicate, learn, and connect with others. The absence of either sense—vision or audio—leads to a perceptual gap, rendering only an incomplete experience of the world incomplete. This interdependence of these two modalities highlights the importance of exploring and understanding the relationship between them, beyond the individual contributions of each to our perception. Exploration of how the combined effect of these signals can lead to a more complete perceptual experience.

In video production, the juxtaposition of sound effects and background music plays a pivotal role. Indeed, Foley artists who specialize in the creation of such effects typically would craft a set of audio recordings and carry out extensive editing to synchronize and fit them with video content. While humans naturally correlate sound with visual information, turning the correlation process into a creation process presents a more complex challenge. For instance, in a scene from the animated film 'Ratatouille,' a chef meticulously cuts vegetables into pieces, accompanied by a perfectly synchronized soundtrack. The soundtrack in this case contains the sound effects of 'cutting vegetables' and the background music piece enhances the emotional impact. Interestingly, the sound we perceive as 'cutting' was actually produced by the act of rubbing two drumsticks together, rather than physically chopping vegetables. While skilled Foley artists possess the expertise to devise clever methods for crafting such sound effects, individuals without such level of expertise will usually find it challenging. In

fact, unlike speech audio streams, which often has temporally aligned text or language as a reference, non-speech audio, encompassing music, sound effects from object interactions, sounds from natural events, and environment-dependent acoustic properties, typically lack a stable and precise way of describing and reproducing their content. Hence, a pressing question arises: Can we develop an automated system capable of generating high-fidelity audio for visual content?

Recent advancements of deep learning models have catalyzed significant advancements and introduced novel applications, especially within the realms of image, video, and text data. However, leveraging deep learning for understanding content in videos with accompanying audio and generating audio from visual stimuli requires additional advances. Such additional complexity stems from the inherent distinction between visual and auditory domains, each characterized by representations and modalities that are significantly different from each other. Bridging these modalities, i.e., translating visual inputs into corresponding audio outputs, demands a nuanced understanding of the complex cross-modal relationships that exist between them.

In particular, the additional complexities stem from video and audio, unlike static images, unfolding over time, adding a temporal dimension that complicates their interrelation. For instance, the occurrence of a dog barking is closely tied to the moment it opens its mouth, rather than merely the presence of the dog itself, highlighting the crucial role of timing in the audio-visual correlation. While certain scenarios might allow for a direct mapping from visual cues to sound—such as identifying specific piano keys being pressed in a top-down piano performance video—this level of precision is not universally applicable. In other contexts, such as music dance videos, the audio-visual correspondence becomes more subjective, emphasizing dance beats and styles rather than explicit visual cues which introduce ambiguity.

Sound, with its intricate composition of frequencies, timbres, and dynamics, presents a formidable challenge in capturing its full spectrum based on the visual information alone. Generating realistic audio that faithfully represents the complexity of sound is far from

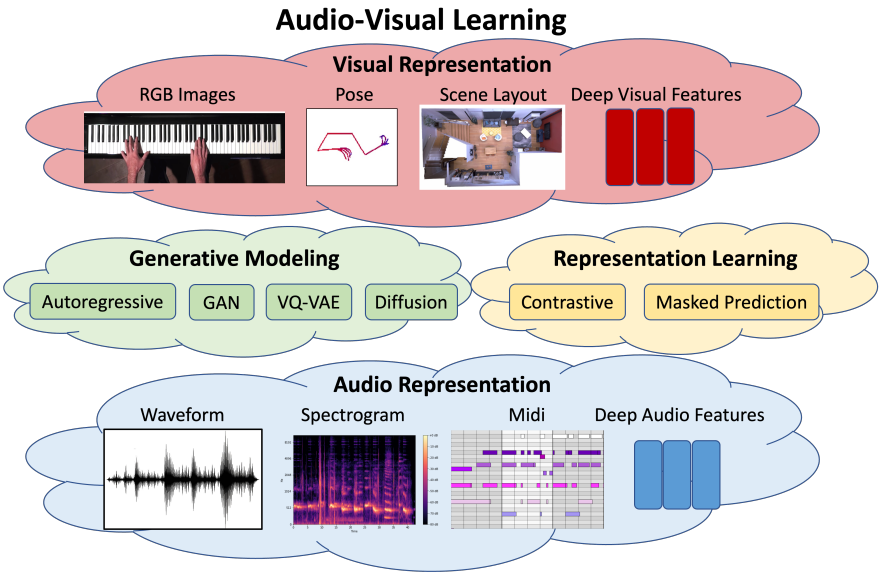

Figure 1.1: A high-level overview of major components in audio-visual learning. The figure illustrates potential visual representations, audio representations, representation learning techniques, and deep generative approaches. The implementation of suitable representations and methodologies is a critical factor and varies significantly depending on the specific application.

straightforward since it needs to correlate key audio characteristics with visual cues and then synthesize a diverse array of acoustic patterns. Given the varied nature of audio-visual correspondences—from speech and music to environmental sounds and beyond—there's a pressing need for deep learning tools capable to comprehend video content and synthesize a wide spectrum of acoustic phenomena.

In the domain of deep learning, the area of study that aims to connect audio and visual representations is encapsulated by audio-visual learning (shown in Figure 1.1). Within this field, two primary avenues have been established by previous research. One avenue employs deep generative models to map visual representations v directly to audio representations a via a function $f_{v \to a}$ parameterized by deep neural networks. While such vision-to-audio genera-

tive models show promise in generation tasks, they tend to be specialized with limited utility toward a broader spectrum of audio-visual comprehension and applications. Conversely, the other avenue focuses on self-supervised audio-visual representation learning, aiming to understand the semantic interplay between audio and visual representations. This approach usually corresponds to mapping audio a and visual v representations into a shared latent space av, represented by two distinct functions ($f_{v\to av}$ and $f_{a\to av}$) parameterized by deep neural networks. While such an approach is useful in the interpretation and mapping of signals, the models developed in current audio-visual representation learning are of a deterministic nature which makes them unsuitable for generative tasks. Figure 1.1 delineates the extensive range of audio and visual representations, along with the methodologies of deep generative models and audio-visual representation learning.

In my research, I aim to bridge the gap between audio-visual representation and generation using a three-pronged approach. The first dimension of my work focuses on enabling high-quality vision-to-audio generation through an intermediate audio-visual representation which serves as a bridge between visual and audio domains. This intermediate representation requires pre-selection based on its known strong correlations with both visual and audio modalities. For instance, in the context of a top-down piano performance video, musical notes emerged as an optimal representation to connect piano keystrokes and their corresponding sounds. Likewise, the rhythm of body movements can synchronize human activities with corresponding rhythmic music. These instances of representation, dictated by our prior knowledge, are precise yet inherently confined to particular scenarios.

Alternatively, to enable a more general transformation between video and audio, a pre-trained deep-learning model could be developed to extract semantic features as intermediate representations. Indeed, in my research, I have proposed such representation. The approach of leveraging intermediate representations in vision-to-audio generation can clarify our understanding of the modeled concepts, including their strengths and limitations. However, the reliance on intermediate representations often requires specialized multi-stage modeling strategies thereby posing challenges in adapting these models to broad audio-visual tasks.

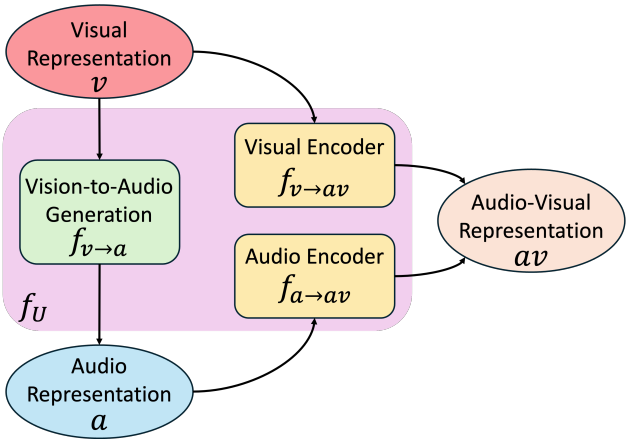

Figure 1.2: This thesis introduces a unified audio-visual framework (f_U) designed to integrate vision-to-audio generative modeling ($f_{v\to a}$) and audio-visual representation learning that maps visual (v) and audio (a) representations into a joint latent space (av) utilizing visual and audio encoders $f_{v\to av}$ and $f_{a\to av}$, respectively.

In addition to leveraging explicit audio-visual representation to connect vision and audio through multi-stage generative modeling, I have studied mapping of vision-to-audio in single-stage generative modeling and learning of implicit audio-visual representations during the vision-to-audio generation process. In this direction, I introduced unsupervised approaches capable of producing instrumental sounds based on musicians' body movements. The training process of the generation of diverse instrumental playing movements turns out to naturally form various clusters. These effectively capture the characteristics of timbre. A similar functionality appeared in our study in the process of synthesizing impact sounds from visual cues of object-physical interactions using a diffusion model. Furthermore, by associating the geometry of indoor scenes with room impulse responses at discrete locations, it appeared that we were able to learn a neural network model implicitly representing a continuous acoustic field. These works lay a solid groundwork for simultaneously learning audio-visual representations and vision-to-audio generation. Nevertheless, their applicability remains confined to particular applications, precluding their use as general-purpose models.

As depicted in Figure 1.2, to facilitate audio-visual representation learning and vision-to-audio generation across a spectrum of general audio-visual tasks, in the third direction of my study, I proposed an innovative unified audio-visual framework. This general framework is designed to efficiently produce high-fidelity audio from visual inputs and is ready for adaptation to acquire robust semantic audio-visual representations through targeted pre-training on a specific task. Its versatility extends to a wide array of applications, including audio-visual retrieval, event classification, and audio-only classification. Such a novel approach is expected to contribute to both audio-visual learning and to set the stage for the creation of more immersive and contextually rich audio-visual experiences.

1.2 Summary of Lessons Learned and My Research Contributions

A major outcome of my research that I was able to achieve towards the end of my Ph.D. work is addressing the gap between vision-to-audio generation and audio-visual representation learning by introducing a unified audio-visual framework, termed 'Vision to Audio and Beyond' (***VAB***). This framework is able to generate high-quality audio from any video in a fast manner, bypassing the need for complex multi-stage generative modeling. The representations it learns are general and adaptable to a broad spectrum of general audio-visual tasks.

I have built this methodology with a 'bottom-up' approach, first by considering high-fidelity audio generation for various vision-to-music applications. The representations that the models leveraged were *explicit* intermediate representations ranging from specific scenarios to more general contexts. The first application that I addressed is of translating piano performances from top-down videos into piano music, a task characterized by strong audio-visual correspondence. For that purpose, I developed ***Audeo***, a novel system for precise mapping, converting sequential RGB images into symbolic music in MIDI format [KS1]. This system combines a visual classification model with a generative adversarial network to generate piano music. This marked the first successful effort to transcribe music from non-controlled piano performance videos and with no reliance on the original sound. The robust audio out-

put would consistently match expected musical pieces as confirmed by music identification software. I have then proceeded to a less direct audio-visual correlation, synchronizing music soundtracks with human body movements, leading to the work of ***RhythmicNet***. This system generates drum patterns from visual rhythms and integrates additional instrument tracks for a fully aligned music soundtrack [KS2]. Proceeding to an even weaker correspondence level, I developed ***V2Meow***, a system that semantically aligns music with arbitrary video content using a multi-stage transformer-based approach for generating semantic music features [KS5].

In subsequent studies I have then worked on single-stage generative modeling that can learn useful intermediate audio-visual representations as well. In particular, I considered three specific scenarios. 1) Generative modeling of music based on musicians' movements, where I considered generating instrumental sounds through a single generative model and capturing the timbre characteristics of sounds. Here, I first studied human body motion and proposed an unsupervised learning paradigm named ***Predict & Cluster*** to establish associations between sequences of human body poses in the latent space without the need for annotation [KS6]. ***Predict & Cluster*** employs an encoder-decoder architecture and includes a weak decoder training strategy along with a reconstruction task. The latent representation derived from the final state of the encoder effectively represents the input sequence. This approach showed the potential to connect music instrument timbres with human body poses in an unsupervised manner and paved the way for VAB's pre-training task, which predicts masked audio tokens conditioned on visual inputs, laying the foundation for a deeper understanding of audio-visual modalities. 2) Subsequently, I proposed ***MI-Net*** [KS3], a single generative model utilizing musicians' movements to infer neural representations for timbre attributes and generate instrumental music. 3) Extending this concept to physical-object interactions, ***Physics-Diff*** demonstrates how visual cues of object materials can predict their impact sounds and enhance the synthesis of realistic sounds [KS7]. 4) Beyond generating audio content, I investigated representations of acoustic scene properties with ***INRAS*** and proposed a system that models indoor acoustic fields and renders high-fidelity

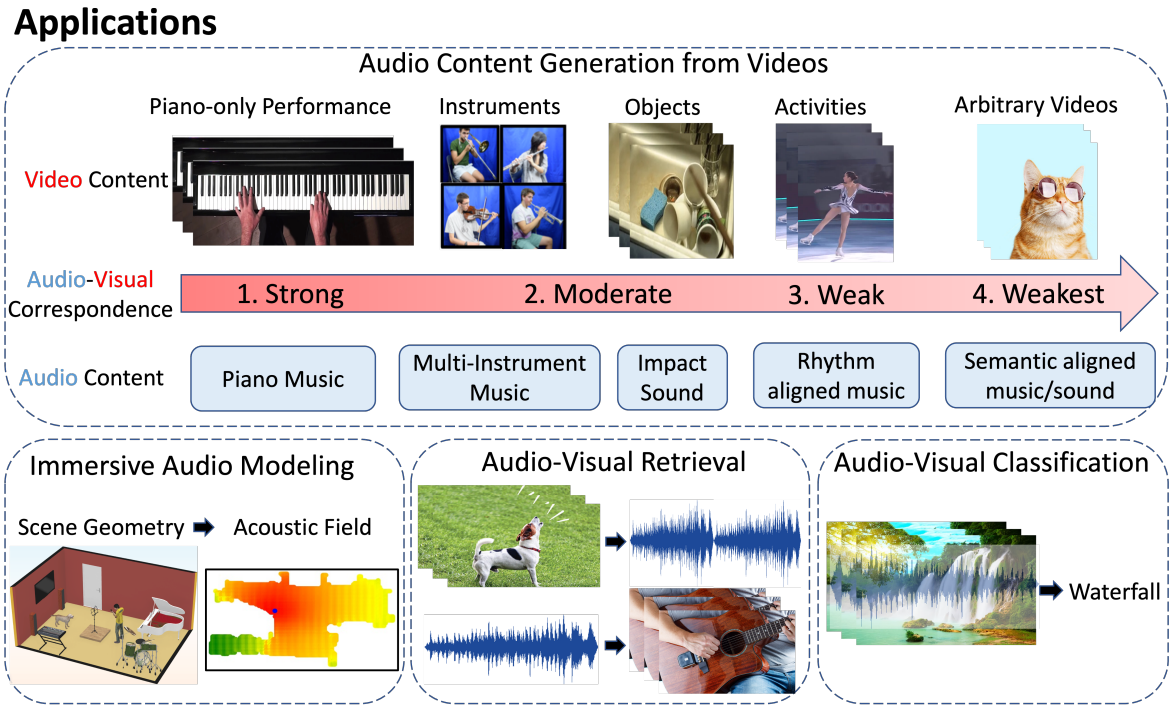

Figure 1.3: An overview of unlocked audio-visual applications in thesis.

impulse responses for any location [KS4].

The VAB model, which is inspired by the above specific tasks and models in audio-visual learning, is able to generalize and propose a pre-trained model. Such pre-trained ***VAB*** model can be then adapted to these specific tasks by fine-tuning. The fine-tuned model can reach state-of-the-art performance, cementing ***VAB***'s role as a unified audio-visual framework that bridges the disconnect between audio-visual generation and representation learning. This approach not only advances the field but also enriches audio-visual experiences with enhanced realism and contextual awareness. A graphical overview of the tasks and models that I have considered is shown in Figure 1.3.

1.3 Thesis Outline

This thesis is organized as follows: Chapter 2 provides a comprehensive overview of fundamental concepts and related works. Chapter 3 describes in detail the model for the generation of music from top-down view piano performances. Chapter 4 details and outcomes of considering the generation of music from arbitrary human movements, expanding the application of video-to-music generation techniques. Chapter 5 outlines a general video-to-music pipeline, offering a broader perspective on audio generation from visual inputs. In Chapter 6, I introduce a self-supervised learning approach that demonstrates how temporal series data, such as human body pose sequences, can be associated without the need for annotations. Chapter 7 builds upon Chapter 6 and explores the generation of instrument music from musicians' body movements. Chapter 8 presents the technique for synthesizing object impact sounds from videos, extending the scope of audio generation. Chapter 9 introduces the spatial audio representation tailored for indoor scenes, enhancing the realism of audio experiences. Chapter 10 presents an audio-visual foundation framework and demonstrates its capability to rapidly generate various types of audio from general videos. Additionally, it demonstrated how this framework can learn audio-visual representations for a range of audio-visual tasks beyond mere generation. Finally, Chapter 11 concludes the thesis by summarizing the key findings and contributions. This thesis includes the material in the author's previous papers published at NeurIPS [KS1, KS2, KS4], AAAI [KS5] and CVPR conferences [KS6, KS7].[1]

[1]In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of University of Washington's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink. If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Chapter 2

FUNDAMENTAL CONCEPTS AND RELATED WORKS

In this Chapter, I present the fundamental concepts and previous research relevant to this thesis. This material encompasses the representation of audio and video data in modeling, audio-visual representation learning, and the utilization of deep learning generative models employed in the proposed methods.

2.1 Audio Representations

Audio representation refers to the way in which sound or audio data is transformed or encoded into a format that can be processed, analyzed, or used by computers and algorithms. It involves converting raw audio waveforms into structured data that retains important information about the sound, such as its frequency, amplitude, and time characteristics. Common audio representations include time-domain waveforms, spectrograms, Mel-frequency cepstral coefficients (MFCCs), and symbolic representations like MIDI. With deep learning, neural representations have emerged as novel audio representations. These representations enable the efficient analysis, manipulation, and modeling of audio data for various applications, including audio generation, recognition, and processing.

Waveform Representation. A general raw audio waveform is a one-dimensional time series sequence $\mathbf{x} = \{x_1, ..., x_t, ..., x_T\}$, where T indicates the length of the audio and $x_t \in \mathbb{R}$. While employing the raw audio waveform as the fundamental audio representation is conceptually straightforward, it is relatively uncommon to designate it as the training target for deep generative models. This is because most approaches rely on autoregressive generation methods (as described below), which generate one sample at a time. The high sampling rate (ranging from 16kHz to 44.1kHz) of raw audio data leads to the modeling of lengthy

sequences, introducing challenges such as gradient explosion or vanishing and significantly extended sampling times [Re1]. As a result, empirical practice often involves converting the waveform from the time domain into the frequency-time domain.

Frequency-time Representation. An illustrative example of an audio representation in the frequency-time domain is known as a spectrogram. To construct the spectrogram, the initial step involves the computation of a Short-Time Fourier Transform (STFT). The STFT is derived by performing Fourier transforms on successive frames of the audio signal $X(m,\omega) = \sum_t x(t)w(t-m)e^{-j\omega t}$. As the m increases, we slide the window function w to the right. We then compute the Fourier transform for the resulting frame $x(t)w(t-m)$, and the final X is a function of both time m and frequency ω. Given the STFT result $X(m,\omega)$, a spectrogram is simply the squared magnitude of the STFT: $S(m,\omega) = |X(m,\omega)|^2$. The spectrogram, which visualizes the intensity of frequencies over time, offers a more information-rich representation compared to raw audio waveforms. Furthermore, it mitigates the challenge of handling long sequences by dramatically reducing the number of temporal frames, achieved by adjusting the hop length when sliding the analysis window (often resulting in a reduction of more than $100\times$). To better align with human perception, a logarithmic operation is frequently applied to the spectrogram since our perception of sound intensity follows a logarithmic scale [Re2].

Another common audio representation is the Mel scale spectrogram. The Mel scale is a perceptual pitch scale that gauges pitch distances based on listeners' judgments of equal perceptual intervals between them. A formula to convert normal frequency f hertz into m Mels is: $m = 2595\log_{10}(1+\frac{f}{700})$. Consequently, a Mel scale spectrogram is essentially a spectrogram in which the frequencies are transformed into the Mel scale through a non-linear conversion. The Mel scale representation employs a more compact set of Mel bins compared to the Hz bins, effectively serving as a dimension reduction technique. Given that both the logarithmic and Mel-spectrograms can be viewed as 2-D matrices, resembling grayscale images, it becomes feasible to adapt successful deep learning approaches from the image domain to the audio domain with relative ease.

Music Symbolic Representations. Modeling music could be in the form of standard audio using audio waveforms, spectrograms, or Mel-spectrograms. However, due to the compositional structure of the music, there exist additional symbolic representations that could be leveraged for music modeling, such as a symbolic representation which uses an explicit encoding of notes or other musical events. These include machine-readable data formats such as Musical Instrument Digital Interface (MIDI) [Re3]. Digital data formats may be regarded as symbolic since these are based on a finite alphabet of letters or symbols. One of the most straightforward symbolic representations is called the **Piano-Roll** representation. The Piano-Roll is a two-dimensional matrix $M \in \mathbb{R}^2$ where the horizontal axis represents time, and the vertical axis represents music notes. The magnitude of the matrix M_{ij} indicates the velocity of j-th music note at time i. Note that the 'velocity' in MIDI terminology refers to the note intensity. The **MIDI** representations take one more step to encode information for each note event, such as the note onset, note offset, and intensity, and represent them as a list of messages similar to human text language. Due to its essence being similar to natural language, symbolic representations are widely used due to easy adaptation of them into large language models. Indeed, a whole branch of unconditional music generation models use various types of symbolic representation to model music [Re4, Re5, Re6].

Deep Audio Representations. While time-domain and frequency-domain audio representations encapsulate audio information, effectively modeling data in these formats can pose significant challenges due to the complexity and high dimensionality of these representations. Recently, self-supervised learning approaches have been proposed employing encoder-decoder neural network models [Re7, Re8, Re9]. In these models, the encoder compresses either the audio waveform or the spectrogram through down-sampling, converting them into a sequence of low-dimensional latent vectors which can further be quantized into tokens. Subsequently, the decoder network facilitates the up-sampling process reconstructing the latent vectors into their original audio representations.

Due to the condensed nature of these latent vectors, audio generation tasks can leverage the latent space as opposed to dealing directly with the original audio representations. These

encoder-decoder models are often referred to as 'neural audio codecs' due to their ability to perform audio signal compression.

2.2 Visual Representations

Raw Video Representation. The most standard representation of a video is through 4-D tensors $\mathbf{v} \in \mathbb{R}^{T,C,H,W}$ where T represents the number of video frames, C represents the number of channels, H and W represent the height and width of each frame, respectively.

Body Pose Representation. In certain scenarios, raw video data may contain redundant information. For instance, human activities can be adequately represented solely by capturing human body movements, eliminating the need to consider the visual background and appearance. This efficiency principle also applies to certain applications of visual-driven audio generation. For instance, in cases where music accompanies a dance performance, the primary audio-video connection is through the dance rhythm rather than other features such as the visual background or the dancer's appearance. Consequently, it is often more efficient to extract a concise visual representation, a task achievable through human pose estimation techniques such as Openpose [Re10]. The human pose can be represented as a 3-D tensor $\mathbf{h} \in \mathbb{R}^{T,J,D}$ where T represents the time, J indicates the number of joints of the human body, and D represents the 2D or 3D coordinates of the joint positions.

Deep Visual Representations. Beyond utilizing raw video data or human body key point representations, valuable visual information can be harnessed from pre-trained deep learning models in various ways. For instance, well-established pre-trained image classification models like ResNet [Re11] and Inception Network [Re12] are frequently employed to extract high-level visual features represented by a single vector $f_t \in \mathbb{R}^{D_f}$ for each frame, where D_f represents the embedding feature dimension. Alternatively, multi-modal embeddings that are acquired through the training of multimodal correspondence models, such as Contrastive Language Image Pre-training (CLIP), are used in various tasks, including generative applications.

Apart from frame-level visual representations, we can also employ pre-trained video un-

derstanding models to obtain video-level representations [Re13, Re14, Re15].

2.3 Implicit Neural Representations

Traditionally, images and audio have been represented as discrete signals. For instance, images are often considered as grids of pixels, while audio is characterized by discrete samples that represent amplitude variations in a waveform. While these discrete representations have become the standard in many applications, they inherently contain only a finite amount of information about the signal.

To illustrate this limitation, consider an image with a size of 256×256 pixels. Scaling it up to 512×512 pixels is challenging because the original image lacks the additional necessary information. This raises an intriguing question: What if we could employ a continuous function, denoted as f, capable of accurately representing the image signal? With such a function, when we provide a continuous pixel coordinate as input, f would yield the correct RGB value for that pixel. Consequently, we could sample pixel grids at any desired resolution from this continuous function f.

Expressing such a function through conventional mathematical formulas can be exceedingly complex. In recent years, a novel concept known as 'implicit neural representation' (INR) [Re16] has been introduced. INR leverages neural networks to estimate the function f and to achieve continuous signal representation by training on discretely sampled instances of the same signal. More precisely, it belongs to a class of functions Φ that adhere to equations of the following form:

$$f(x, \Phi, \nabla_x \Phi, \nabla_x^2 \Phi, ...) = 0, \Phi : x \rightarrow \Phi(x). \tag{2.1}$$

This implicit problem formulation takes spatial or spatial-temporal coordinates x and, optionally, derivatives of Φ concerning these coordinates as input. The objective is to train a neural network that parameterizes Φ in such a way that it maps x to a specific quantity of interest while satisfying the presented constraint. Consequently, Φ is implicitly defined through the relationship established by f. Neural networks that serve as parameters for such

implicitly defined functions are commonly referred to as 'implicit neural representations.' Recent advancements have demonstrated that these parameterized implicit and continuous representations have made significant strides in the field of computer graphics [Re17].

The characteristic of continuity in a function is advantageous when it comes to modeling and could represent the acoustic field within an indoor environment. This advantage allows us to employ a lightweight neural network instead of the need to store extensive sets of room impulse responses for the comprehensive representation of the acoustic field.

2.4 Deep Generative Models

Autoregressive Models. Among primary generative methods for audio, the autoregressive model stands out, since it learns an explicit distribution shaped by a prior determined by the structure of the model. The joint probability of an audio waveform $\mathbf{x} = \{x_1, ..., x_t, ..., x_T\}$ can be factored into a product of conditional probabilities as follow:

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t | x_1, ..., x_{t-1}). \tag{2.2}$$

Each individual audio sample x_t is conditioned on the samples from all preceding time steps. Autoregressive models are advantageous, including the possibility to consider explicit probability densities, stable training processes, and natural compatibility with audio signals. As a result, autoregressive models are commonly used in various audio generation applications. To train a deep autoregressive model, one may consider using a 1D convolutional network [Re18], a transformer-based network [Re19], or a recurrent neural network [Re20], all of which are viable candidates.

GANs. A Generative Adversarial Network (GAN) [Re21] is another approach that has demonstrated favorable results in numerous generative tasks, including music generation [Re22]. It is inspired by game theory, where a generator and a critic engage in a competitive yet mutually beneficial dynamic. GAN comprises two essential models. 1) Discriminator (D): This model estimates the probability that a given sample originates from the real dataset. It serves as a critic and is trained to distinguish between fake and real samples. 2) Generator

(G): The generator produces synthetic samples based on a noise variable input, denoted as z. Its training objective is to capture the distribution of the real data, ensuring that the generated samples closely resemble real samples and deceive the discriminator into assigning them high probabilities.

During training, these two models engage in a zero-sum game, where the generator G strives to outsmart the discriminator by producing synthetic samples that appear genuine, while the critic (discriminator D) endeavors not to be fooled. Such intriguing adversarial relationship motivates both models to continually enhance their capabilities.

Mathematically, our goal is to have the discriminator D accurately classify real data, which is achieved by maximizing the term $\mathbb{E}_{x\sim p_r(x)}[[\log D(x)]$. Simultaneously, when presented with a fake sample generated by $G(z), z \sim p_z(z)$, the discriminator should output a probability, $D(G(z))$, close to zero. This is accomplished by maximizing the term $\mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$. The generator, on the other hand, seeks to minimize $\mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$, effectively increasing the likelihood that D assigns a high probability to a fake example. In essence, D and G engage in a minimax game, aiming to optimize the following loss function

$$\min_G \max_D L(D,G) = \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))] \tag{2.3}$$

$$= \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{z\sim p_g(z)}[log(1-D(x))]. \tag{2.4}$$

While GANs have achieved success in various generative tasks, their training process often invovles slow convergence and instability. To address these challenges, it is a common practice to combine autoregressive models with adversarial training. This hybrid approach not only enhances the training stability but also improves the quality of the generated samples.

Diffusion Models. Recently, a novel class of generative models, known as diffusion models [Re23], has garnered significant attention for their ability to generate high-quality samples [Re24, Re25, Re26, Re27, Re28, Re29, Re30]. These models draw inspiration from non-equilibrium thermodynamics and employ a Markov chain of diffusion steps. This process gradually introduces random noise to data and subsequently learns to reverse the diffusion

process to reconstruct desired data samples from the noise. Below, I provide a high-level introduction to the diffusion model here. In the following description, equations and derivations are from [Re31].

In the context of diffusion models, the procedure is initiated with a data point sampled from a real data distribution denoted as $x_0 \sim p(x)$. The forward diffusion process proceeds by adding incremental small amounts of Gaussian noise to the initial sample over T steps. This results in a sequence of noisy samples represented as $x_1, ..., x_T$. The step sizes during this process are regulated by a predetermined variance schedule $\{\beta_t \in (0,1)\}_{t=1}^{T}$.

$$p(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{2.5}$$

$$p(x_{1:T}|x_0) = \prod_{t=1}^{T} p(x_t|x_{t-1}) \tag{2.6}$$

As the diffusion process advances and the number of steps t increases, the initial data sample x_0 progressively loses its discernible characteristics. In the limit where $T \to \infty$, the final sample x_T converges to an isotropic Gaussian distribution.

The advantageous aspect of the aforementioned procedure is its ability to generate samples x_t at any chosen time step t in a closed form through a reparameterization. Let $\alpha_t = 1-\beta_t$ and $\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i$:

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1} \tag{2.7}$$

$$= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\overline{\epsilon}_{t-2} \tag{2.8}$$

$$= \cdots \tag{2.9}$$

$$= \sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, \tag{2.10}$$

where $\epsilon_{t-1}, \epsilon_{t-2}, \cdots, \sim \mathcal{N}(0, I)$, and $\overline{\epsilon}_{t-2}$ merges two Gaussians. We can substitute back to equation to obtain $p(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)I)$. Note that the merged standard deviation is $\sqrt{(1-\alpha_t)+\alpha_t(1-\alpha_{t-1})} = \sqrt{1-\alpha_t\alpha_{t-1}}$.

If we are able to reverse the aforementioned process and sample from $p(x_{t-1}|x_t)$, we would be able to reconstruct the true sample from a Gaussian noise input, $x_T \sim \mathcal{N}(0, I)$. However, estimating $p(x_{t-1}|x_t)$ is not straightforward as it requires access to the entire dataset.

Consequently, an auxiliary model q_θ is used to approximate these conditional probabilities in order to execute the reverse diffusion process

$$q_\theta(x_{0:T}) = q(x_T) \prod_{t=1}^{T} q_\theta(x_{t-1})|x_t) \tag{2.11}$$

$$q_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{2.12}$$

We thereby train μ_θ to predict $\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_t)$. Since x_t is accessible as input during training, we can reparameterize the Gaussian noise term to predict ϵ_t based on the input x_t at time step t:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(x_t, t)) \tag{2.13}$$

$$x_{t-1} = \mathcal{N}(x_{t-1}; \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\theta(x_t, t)), \Sigma_\theta(x_t, t)) \tag{2.14}$$

The loss term L_t is used to minimize the difference from $\tilde{\mu}$:

$$L_t = \mathbb{E}_{x_0,\epsilon}\Big[\frac{1}{2\|\Sigma_\theta(x_t, t)\|^2}\|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2\Big] \tag{2.15}$$

After simplifying the expressions, the final training objective is as follows

$$L_t = \mathbb{E}_{t\sim[1,T],x_0,\epsilon_t}\Big[\|\epsilon_t - \epsilon_\theta(\sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon_t, t)\|^2\Big]. \tag{2.16}$$

Diffusion models offer both analytical tractability and flexibility when compared to GANs. However, they face a significant challenge as they rely on long Markov chains of diffusion steps for sample generation and thus can be computationally expensive. Fortunately, numerous new methods have been introduced to significantly expedite this process.

2.5 Video-to-Audio Generation

Audio generation from visual cues, a field with a rich history, has undergone significant transformations, especially with advancements in deep learning. Traditional methods predominantly utilized linear modal synthesis for sound production related to rigid objects [Re32].

While adept at creating realistic sounds reflective of object geometry, these techniques necessitated extensive efforts in simulating and fine-tuning parameters for different virtual object materials during modal analysis, a process that became increasingly cumbersome when applied to complex scenes with diverse materials [Re33].

In recent years, the advent of deep learning methodologies has shifted the landscape of sound synthesis. A notable contribution by Owens et al. [Re34] involved predicting sounds from object interactions in uncontrolled settings, such as using wooden drumsticks. The approach initially focused on sound feature prediction, followed by sound generation through an exemplar-based retrieval algorithm. Concurrently, natural sound generation also gained traction, exemplified by Zhou et al.'s SampleRNN-based method [Re35] for directly predicting audio waveforms from YouTube videos, although confined to ten predefined sound categories.

There have been significant strides in generating synchronized audio for input videos, often employing perceptual loss [Re36] and information bottleneck [Re37] strategies. Innovative approaches have been also proposed in the musical domain, particularly videos showcasing instrumental performances. A ResNet-based method [Re38] was introduced for deducing pitch and onset events from piano performance videos. Extending this concept, Foley Music [Re39] developed a Graph-Transformer network capable of producing MIDI events from human body keypoints leading to convincing music synthesis. These methods, while effective, were generally limited to specific sound types and lacked a unified model approach.

In addition to models capitalizing on visual cues from human motion, a music Transformer model emerged, designed for generating background music tailored to videos [Re40]. However, many of these techniques rely on symbolic music representations, confronting the challenge of scarce training data comprising transcribed music paired with corresponding instrument-playing videos. This limitation often restricts the generative scope to certain instruments and visual contexts.

More recently, the field has seen the development of more versatile video-to-audio generation models [Re41, Re42, Re43] suitable for 'in-the-wild' videos. These novel methods

integrate generative modeling within discrete tokenized audio and music signals and achieve higher fidelity in audio outputs. Additionally, audio synthesis from visual data through diffusion-based vision-to-audio modeling [Re44] is an emergent research direction.

2.6 Audio-Visual Representation Learning

The exploration of learning representations for audio and visual data encompasses both supervised and self-supervised methodologies. Initially, deep neural networks were applied to annotated audio-visual pairs in a supervised context [Re45, Re46, Re47]. More recently, self-supervised learning techniques [Re48, Re49, Re50] have leveraged the inherent correspondence between audio and visual elements to derive representations beneficial for a range of downstream tasks including audio-visual classification and retrieval. Among these, contrastive learning and masked autoencoder-based strategies for processing audio and visual signals turned out to be specifically effective [Re51, Re52, Re53, Re54].

Audio-visual contrastive learning, in particular, exploits the similarities within audio-visual pairs both within the same video and across disparate videos, utilizing these relationships as self-supervised signals to articulate a discriminative goal. A typical audio-visual contrastive learning framework comprises an audio encoder E_a and a visual encoder E_v, each of which processes the audio signal a and visual signal v respectively, producing corresponding representations a^c and v^c. The training phase involves optimizing both encoders using a contrastive loss L_c formulated as:

$$l_c = -\frac{1}{N}\sum_{i=1}^{N}\log\left[\frac{\exp\left(s_{i,i}/\tau\right)}{\sum_j \exp\left(s_{i,j}/\tau\right)}\right] \tag{2.17}$$

where $s_{i,j} = ||a_i^c||^T||v_j^c||$, a^c and v^c denote audio and video representations, and τ represents a temperature parameter. This loss function aims to align audio and visual pairs from the same video while separating those from different videos. Such loss is particularly effective for retrieval applications.

Conversely, masked autoencoder-based approaches draw inspiration from context-aware learning strategies exemplified by models like BERT in natural language processing, extend-

ing these concepts to image, video and audio signals. The synergy of these two approaches has yielded significant enhancements in audio-visual representation quality. Central to masked autoencoder methods is the use of corrupted versions of audio ($\hat{a}$) and visual ($\hat{v}$) signals as inputs to an encoder-decoder model, which outputs denoised versions (a and v). Optimization incorporates a reconstruction loss.

Predominantly, audio-visual representation learning has focused on utilizing raw video frames and audio spectrogram data, striving to capture a comprehensive array of information. Yet, this approach poses significant challenges in generative modeling due to the complex patterns and high dimensionality of raw data. In this thesis, we propose an innovative route by employing latent audio tokens, aiming to facilitate both representation learning and generative modeling within a cohesive framework as a novel perspective on audio-visual synthesis.

Chapter 3

MUSIC GENERATION FROM TOP-DOWN VIEW PIANO PERFORMANCE

3.1 Motivation

The harmonious fusion of a musician's expertise with the enchanting tones of a musical instrument gives rise to the captivating essence of 'live music.' This experience is not only enthralling due to the melodies that fill the air but also profoundly moving as one witnesses the remarkable synchrony between the musician and their instrument.

A challenging test in the realm of music generation involves the task of transcribing music solely from visual cues. This entails reconstructing the audio stream of a musical performance based solely on visual information, such as the positions of the musician's hands, body, key presses, and even the use of pedals, all combined together to recreate the musical composition. This task goes beyond simple synchronization, as it demands precise timing between visual cues and audio output. The complexity arises from the inherent differences in the perception rates of visual and audio streams. While visual perception typically operates at a slower pace, the combination of both audio and visual streams necessitates rapid synchronization, ensuring that the audio aligns seamlessly with the visual input. This intricate process involves not only establishing associations between video frames and audio segments but also 'filling' the audio gap between these frames by extrapolating and interpolating sound events effectively bridging the past and future frames.

Within video frames, a plethora of visual information is available, some of which may not directly contribute to the generation of music. In light of this, it is reasonable to consider the utilization of intermediate features for the translation from video to audio. These intermediary features should effectively capture both the mechanical and perceptual aspects of

the interaction between the musician and the musical instrument, serving as valuable tools for sound representation and synthesis. One such candidate for this purpose is the Musical Instrument Digital Interface (MIDI) protocol, which facilitates the exchange of musical information among instruments, encoding various keyboard functions and musical attributes. Variants of MIDI, such as Pseudo-MIDI (binary, without expressive velocities), offer an even more concise means of encoding keyboard functions and musical attributes collectively. Additionally, connecting visual actions with the changing frequencies of the audio signal over time, as represented in a spectrogram, can serve as a valuable intermediary step in this process.

For such a research problem, I have introduced a comprehensive pipeline known as ***Audeo***, designed to generate music from silent piano performance videos. ***Audeo*** orchestrates the transformation of visual performance cues into their corresponding audio counterparts through a three-stage process, ultimately recovering Midi signals. In the initial stage, when presented with a top-view video, I employ a multi-scale feature attention deep residual network to both capture the visual intricacies and predict the activation of piano keys at each frame, essentially forming a 'Piano-Roll' representation (referred to as **Video2Roll Net**). This stage involves a multi-label classification task, although it doesn't account for long-term temporal dependencies, which can result in less faithful music generation. Consequently, in the second stage, I introduce a Generative Adversarial Network (GAN) [Re21] to refine and enrich the 'Piano-Roll' with essential musical attributes, yielding the Midi signal (termed **Roll2Midi Net**). This step plays a pivotal role in furnishing symbolic musical representation. The third and final phase of the ***Audeo*** pipeline involves the synthesis of Midi data into audio signals (Midi Synth). Given that the predicted Midi data is binary and lacks expressive velocity information, I propose employing the same velocity values for synthesizing the mechanical audio. This synthesis can be realized through either a conventional Midi synthesizer or a deep synthesizer for a more lifelike auditory experience. The deep synthesizer translates Pseudo-Midi into a spectrogram, followed by audio generation. An overview of the ***Audeo*** system is shown in Fig. 3.1.

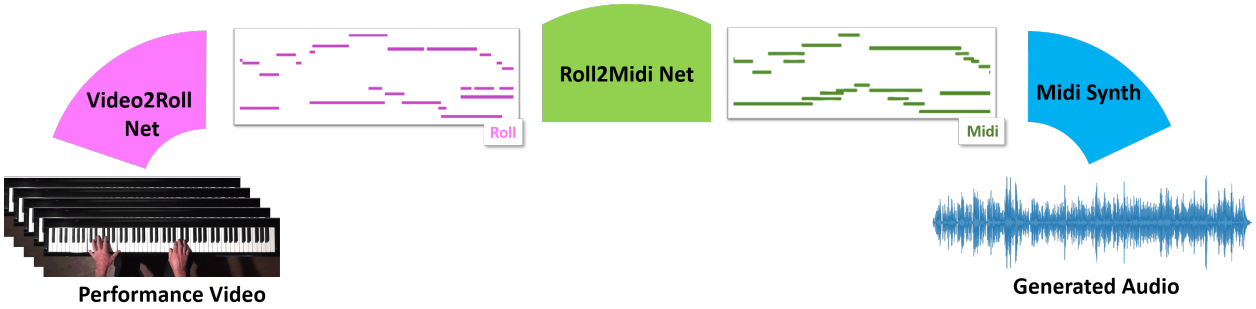

Figure 3.1: Given an input of video frames of a musician playing the piano, ***Audeo*** generates the music for that video. (Figure from [KS1])

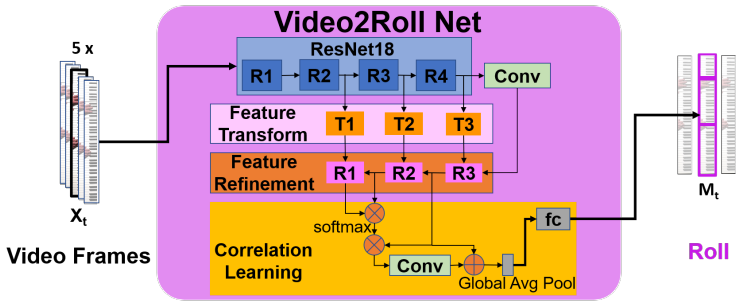

Figure 3.2: Detailed schematics of the components in VIDEO2Roll Net (Figure from [KS1])

3.2 Methods

I utilize Midi as an intermediate signal for converting piano video frames into audio output. Since most piano performance videos found online lack accompanying Ground Truth Midi, I extract the Pseudo Ground Truth (GT) Midi from the audio using the Onset and Frames framework [Re55]. The Pseudo GT Midi is a two-dimensional binary matrix $M \in \mathbb{R}^{K \times T}$ where K is the number of pitches, and T is the number of frames. For each entry, $M_{k,j}$, 1 indicates if the key k is sustained at frame j and 0 otherwise. The subsequent sections detail each component of the *Audeo* system.

Video2Roll Net: The initial stage of our process is characterized as a multi-label im-

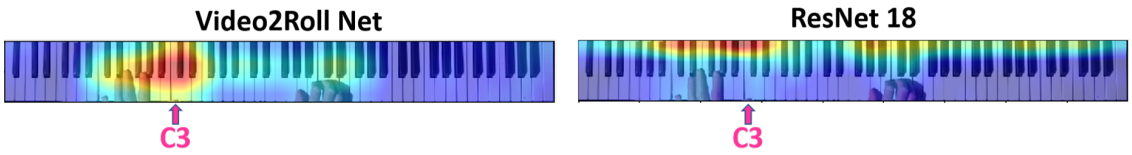

Figure 3.3: Visualized feature maps comparison between Video2Roll Net (left) and ResNet18 (right) using Scored Weighted Class Activation Heatmap (Score-CAM). This example demonstrates that our method can locate the delicate visual cues of the pressed C3 key more accurately. (Figure from [KS1])

age classification problem. Each video clip X is treated as a four-dimensional tensor $X \in \mathbb{R}^{T\times C\times H\times W}$ where T, C, H, W are time, channel, height, and width dimension respectively. For input into the Video2Roll Net, I employ five consecutive grayscale frames $X_{t-2,t-1,t,t+1,t+2}$, aiming to predict the keys pressed in the central frames X_t. This involves estimating the conditional probability of key presses at frame t, given the video frames $X_{t-2:t+2}$, with the probability of estimated keys at frame t denoted as $P(\hat{M}_{:,t}) = P(M_{:,t}|X_{t-2,t-1,t,t+1,t+2})$. The backbone of this system is ResNet18, which is tailored to accommodate the unique challenges of this task: 1) the subtle visual indicators of sustained keys in comparison to more prominent elements like hands and fingers; 2) the correlation of pressed keys in each frame, influenced by musical harmony, which suggests certain key combinations are more likely to co-occur; 3) the importance of spatial dependencies in detecting sustained keys, a feature typically unaddressed by conventional CNNs due to their spatial invariance. To tackle these challenges, I've developed a multi-scale feature attention network, inspired by [Re56]. Video2Roll Net, built on the ResNet18 framework, comprises three key modules: feature transformation, feature refinement, and correlation learning. The feature refinement process parallels a feature pyramid network (FPN) [Re57], utilizing a top-down feature propagation approach. However, unlike standard FPNs, Video2Roll Net first modifies and re-calibrates multi-scale features at residual blocks through the feature transform module before advancing them. This enhancement significantly improves the network's ability to discern visual cues at various scales.

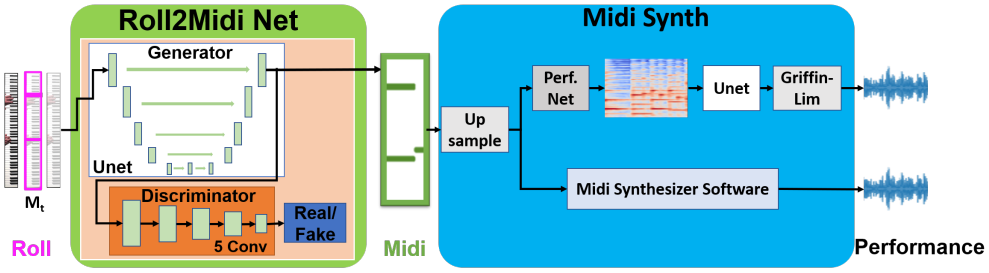

Figure 3.4: Detail schematic of Roll2Midi Net and Midi Synth components of *Audeo* system. (Figure from [KS1])

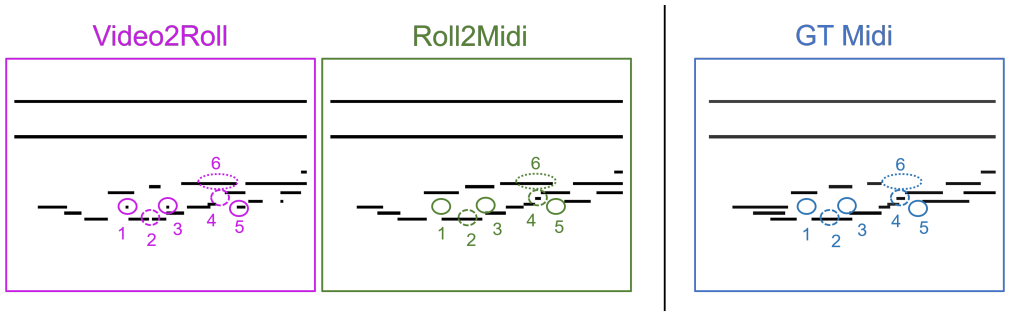

Figure 3.5: Comparison of Roll, Midi and Pseudo GT Midi. Solid ellipses (1,3,5) : elimination of false positives; Dashed ellipses (2,4): elimination of false negatives; Dotted ellipse (6): failure not eliminated. (Figure from [KS1])

Lastly, the correlation learning module employs a self-attention mechanism to understand feature spatial dependencies and semantic relevance. The attention response at any location is informed by the features at other locations. This multi-scale feature attention approach is pivotal in precisely identifying regions corresponding to pressed keys, which is crucial for generating meaningful music (as shown in Fig. 3.3).

Roll2Midi Net: The prediction of the Piano Roll (Roll) $\hat{M}_{:,1:T}$ of Video2Roll Net is imperfect due to several challenges. One notable issue is hand occlusions in video frames, which hinder Video2Roll Net's ability to detect changes in key presses. Additionally, since

$\hat{M}_{:,t}$ is predicted frame by frame, there lacks a temporal correlation in Roll predictions. Another complexity arises from the Pseudo GT Midi, derived from the Onset and Frames framework [Re55] based on the audio stream. A common occurrence is that if a key is sustained for a prolonged period, its frequency magnitude gradually diminishes to zero in the Pseudo GT Midi, marking it as 'off.' However, as our Video2Roll Net relies solely on short-term visual data, all pressed keys remain active, leading to discrepancies between prediction and the actual audio. To address these issues, I have incorporated a generative adversarial network (GAN) [Re21] to refine and enhance the Video2Roll results $\hat{M}_{:,1:T}$, aligning them more closely with the Pseudo GT Midi. This GAN comprises a generator G and a discriminator D. The generator's input is the Roll predictions $\hat{M}_{:,T_1:T_2}$, where each column of $\hat{M}$ represents a probability score obtained from the final fully connected layer of Video2Roll Net, post-application of a sigmoid function. By utilizing probability scores instead of binary thresholds, the generator can adjust these probabilities, fostering a more accurate and robust Pseudo Midi representation. The GAN objective is defined by:

$$\min_G \max_D \mathbb{E}_{M\sim\mathcal{M}}[\log D(M)] + \mathbb{E}_{\hat{M}\sim\hat{\mathcal{M}}}[log(1 - D(G(\hat{M})))]. \tag{3.1}$$

Our generator is structured as a U-Net [Re58] with five levels of depth, while the discriminator is composed of a 5-layer CNN. The inclusion of a discriminator, as opposed to relying solely on a U-Net, enables the model to capture a broader pattern of the Pseudo Midi, deeming a prediction acceptable when it sufficiently resembles 'real' data. Given the substantial variation in Midi across different musical styles, using only a U-Net could lead to overfitting with the training data. To optimize both the generator and the discriminator, I employ the Mean Square Error (MSE). During the inference process, the Roll representation is fed into the generator, resulting in a refined Midi representation $\hat{M}_R = G(\hat{M})$. This Roll2Midi Net significantly enhances the accuracy of the overall predictions. The refined Midi generated is of sufficient quality to be synthesized, producing music that closely approximates the ground truth. Fig. 3.5 shows that Roll2Midi can partially eliminate the false positives and false negatives in the Roll.

Midi Synth: Both the Roll and the Pseudo Midi can be effectively converted into audio using traditional Midi synthesizers. This approach yields clear, robust, and reasonably musical outputs from the predicted Midi. Additionally, classical Midi synthesizers offer versatility, supporting creative applications such as generating music with diverse timbres from a single piano performance video, simply by changing the instrument settings during synthesis. While these results are intriguing, the audio produced from classical Midi synthesizers tends to sound mechanical. This is because the predicted Pseudo Midi is binary and lacks expressive velocities. Estimating expressive velocities at the Midi level requires a Ground Truth (GT) Midi that specifies them. However, the Onsets-and-Frames framework I use for GT generation does not provide sufficiently precise velocity predictions. Therefore, I explore the potential of generating more realistic music using deep synthesizers with Pseudo Midi predictions. To achieve this, I initially pre-train a PerfNet [Re59] with Pseudo GT Midi M. PerfNet is designed to learn a transformation, denoted as H, between M and the corresponding spectrogram S. Utilizing the pre-trained PerfNet, I then forward propagate the Midi $\hat{M}_R$ to produce an initial estimated spectrogram $\hat{S}_R = H(\hat{M}_R)$. It's important to note that despite refinement, a discrepancy between M and $\hat{M}_R$ persists. I found that directly using PerfNet to learn the transformation from $\hat{M}_R$ to S struggles with generalization. This challenge is likely due to the sensitivity of the transformation process from Pseudo Midi to the spectrogram, complicating generalization. To overcome this, I introduce an additional U-Net for refinement at the spectrogram level. This U-Net is formulated as a function U, and its goal is to minimize the L1 distance between $\hat{S}_R$ and S: $L_1(\hat{S}_R, S) = \|U(\hat{S}_R) - S\|$.. I discovered that estimating an initial rough spectrogram and then refining it at the spectrogram level enhances generalization. Finally, to convert the refined spectrogram into an audio waveform, I employ the Griffin-Lim algorithm [Re60].

3.3 Experiments

Datasets. Unlike previous studies conducted in specific laboratory settings, I evaluate ***Audeo*** pipeline using publicly available piano performance videos from YouTube, with a slight

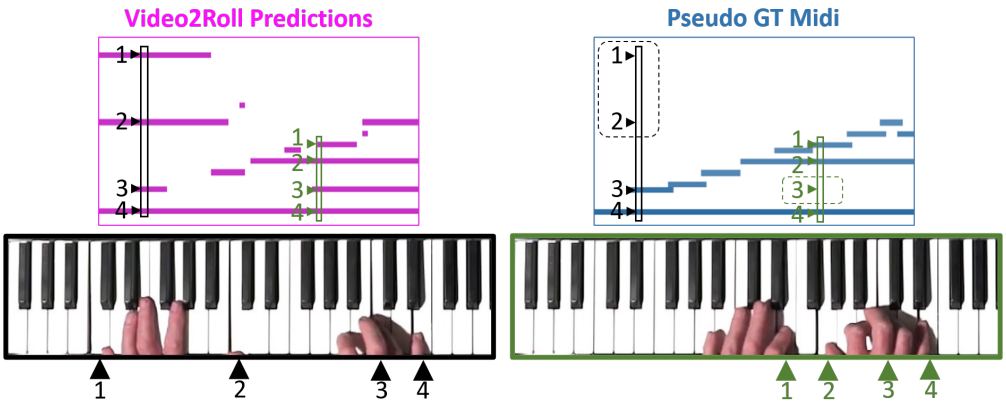

Figure 3.6: Examples of Pseudo GT Midi mismatches with pressed keys. Keys that are active in our predictions and in video frames (black and green) are marked as off in Pseudo GT Midi (dashed).(Figure from [KS1])

preference for top-view recordings that show the entire keyboard. Importantly, uniformity in the instrument and camera setup is unnecessary for our data. Specifically, we utilize videos recorded by Paul Barton[1] with a frame rate of 25fps and an audio sampling rate of 16kHz. The Pseudo GT Midi is derived using the Onsets and Frames framework (OF) [Re55], which, due to its low precision in predicting expressive velocity, results in all Pseudo GT Midi being binary and down-sampled to 25fps. We meticulously crop these videos to display only the full keyboard, removing irrelevant frames, such as logos or black screens. Additionally, we align the video with the Pseudo GT Midi and audio by trimming silent sections up to the first frame, where a key is pressed while retaining all silent frames within each performance. Our evaluation employs two distinct sets:1) ***Pseudo Midi Evaluation Set:*** This set assesses the predictions of our Video2Roll Net and Roll2Midi Net. It includes 24 videos from Bach's Well-Tempered Clavier Book One (WTC B1), totaling 115 minutes of training data. The test set comprises the first three Prelude and Fugue performances from Bach's Well-Tempered Clavier Book Two (WTC B2), amounting to 12.5 minutes. This translates to 172, 404 train-

[1] `https://www.youtube.com/user/PaulBartonPiano`

ing images and 18,788 testing images. It's important to note that the Pseudo GT Midi we use, while imperfect, is essential for our evaluation. We also implement additional audio evaluation protocols detailed below. 2)***Audio Evaluation Set:*** This set focuses solely on audio evaluation. Our objective is to ascertain if the generated music is recognizable by music identification software. This test set includes 35 videos from WTC B2 (comprising 24 Prelude and Fugue pairs and 11 variants), 8 videos from variants of WTC B1, and 9 videos from other composers. In total, this set encompasses 52 videos, amounting to 297 minutes.

Evaluation Metrics. For the ***Midi Evaluation***, I assess the performance of **Video2Roll Net** and **Roll2Midi Net** by calculating and reporting key metrics: precision, the recall, the accuracy, and the F1 score, all evaluated at the frame-level as defined in [Re61]. In order to benchmark against other methods, I reproduce the models proposed in prior research [Re38] and test them using our Pseudo Midi Evaluation set. For the ***Audio Evaluation***, I employ the widely-used music identification App SoundHound[2] to conduct detection tests on the music generated by our system. A detection is considered successful if SoundHound accurately identifies and displays the correct source of the music. Conversely, it's marked as a failure if the app either fails to recognize the music or identifies an incorrect source. The effectiveness of our approach is quantified by the average detection rate at the segment level.

Midi Evaluation: Table 3.1 presents the performance results of *Audeo* in generating the Roll and the Pseudo Midi, in comparison to other methods. The Video2Roll Net stands out for its ability to detect detailed visual cues, leading to superior recall, accuracy, and F1-score relative to previous studies. A critical finding is the importance of minimizing false negatives to ensure the generation of a complete melody without omitting notes. However, the relatively lower precision of Video2Roll Net highlights a common issue in audio-visual mismatch, making some false positives inevitable (See Fig. 3.6). This underscores the complexity of generating music from visual information, a challenge that we anticipate will be addressed more effectively in future developments and the necessity of the Roll2Midi Net

[2]`https://www.soundhound.com/`

Model	Precision	Recall	Accuracy	F1-score
ResNet [Re38]	64.3	54.7	40.4	49.7
ResNet+Aggregation+slope [Re38]	61.5	57.3	41.2	50.8
Video2Roll Net (Our)	61.2	65.6	46.4	56.4
Roll2Midi Net TS=0.4 (Our)	60.0	**77.0**	50.6	**61.5**
Roll2Midi Net TS=0.5 (Our)	**65.1**	69.9	**50.8**	60.4

Table 3.1: Precision, recall, accuracy and F1-score in (%) for pseudo Midi evaluation. If not specified, all results use threshold (TS) = 0.4 after applying the sigmoid function. The bold number indicates the best result. (Table from [KS1])

	Total	Bach WTC B1 Variants	Bach WTC B2 & Other
ResNet+FluidSynth	55.9	74.2	52.9
Roll+FluidSynth	62.6	79.6	59.6
Midi+PerfNet	73.0	80.6	71.6
Midi+FluidSynth	**73.9**	**85.6**	**72.4**
Ground Truth	89.2	92.6	87.7

Table 3.2: Sound Hound music identification rate in (%). (Table from [KS1])

for achieving a cleaner and more robust symbolic representation. The implementation of a generative adversarial network within Roll2Midi Net plays a pivotal role, enabling it to reduce both false negatives and false positives by assessing the realism of the generated Pseudo Midi. Significantly, Roll2Midi Net enhances overall performance substantially, with its F1 score surpassing the best competing model [Re38] by over 10%.

Audio Evaluation on Music Identification: I compare the effectiveness of SoundHound in detecting music samples generated by the ***Audeo*** system against those produced by a ResNet baseline and the original ground truth audio. Additionally, to isolate the ef-

fects of the synthesizer, I synthesize both the Roll and Pseudo Midi obtained from ***Audeo*** using either FluidSynth or PerfNet. The outcomes of the music identification are detained in Table 3.2. The findings reveal that all methods utilizing ***Audeo*** significantly outperform the ResNet baseline, as well as the synthesizing of Pseudo Midi via FluidSynth or PerfNet. This suggests that ***Audeo*** effectively captures the essence of the learned music and exhibits resilience against variations in performance. Interestingly, for test videos featuring music styles not covered in the training set, such as works by Scott Joplin, a noticeable discrepancy remains when compared to the ground truth (72.4 vs. 87.7%). However, the identification results still demonstrate the robustness and versatility of the ***Audeo*** system. Overall, the combination of Midi with FluidSynth yields the highest detection rates, surpassing the ResNet baseline by 18%. It's worth noting that using PerfNet as the synthesizer results in slightly lower detection rates compared to FluidSynth in this particular test. While a deep synthesizer like PerfNet might better capture emotional nuances and naturalness in the spectrogram domain, it also tends to introduce noise, which poses a significant challenge to mitigate.

Chapter 4

MUSIC FROM ARBITRARY HUMAN MOVEMENTS

4.1 Motivation

Rhythmic sounds are omnipresent in our environment. For example, sounds manifested from raindrops hitting surfaces, birds chirping, or machines, all produce distinct sound patterns. When paired with visual scenes, these sounds significantly enhance the viewer's perception by adding layers of semantic context, facilitating communication and drawing attention to specific aspects of the scene, among other benefits. In human activity scenarios, rhythmically aligned music can underscore and amplify the actions being depicted. This synchronization between visual rhythm and musical beat is often achieved through the manual selection of soundtracks in professional video editing.

Drum instruments, known for their pivotal role in establishing the foundational rhythm of music, vary widely in design and mechanics but share the common objective of setting the music's core rhythm. Drums, with a history dating back to around 6000 BC, and even earlier sound-producing instruments based on the principle of striking objects together, form the backbone of musical rhythm [Re62]. The layering of additional instruments over drum patterns introduces secondary rhythms and melodies, culminating in complex, multi-dimensional music. It is a common practice among composers to initiate the composition process by crafting the drum track's rhythm, which then serves as a base for gradually layering other instrumental tracks, enriching the final musical piece.

Motivated by the potential to link rhythmic soundtracks with video content, this work delves into the automatic generation of rhythmic music that mirrors movement patterns of human bodies. Mirroring traditional music composition and improvisation techniques, we initially focus on creating rhythms that closely align with the video's depicted move-

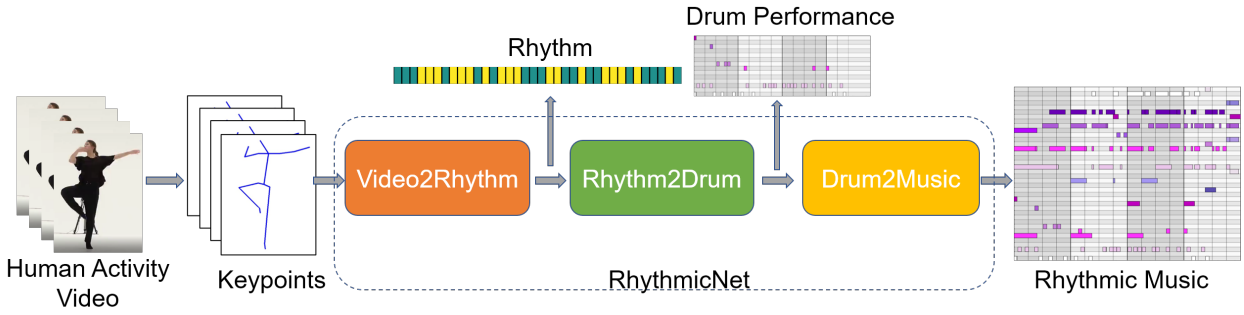

Figure 4.1: System Overview of RhythmicNet. Body keypoints are extracted from human activity video and are processed through the Video2Rhythm stage to generate the rhythm. Afterward, Rhythm2Drum converts the rhythm to drum performance. In the last step, the Drum2Music component adds additional instrument tracks on top of the drum track. (Figure from [KS2])

ments. This foundational rhythm serves as the basis for generating new drum music that complements the body movements. Following the establishment of a rhythm, we emulate the improvisational addition of other instruments (such as piano and guitar) to enrich the musical composition.

In Chapters 3, we observe that musical instruments often correlate strongly with the visual content, echoing the synchronization found in music-dance videos, where body movements naturally align with rhythmic music. To extend this concept, I propose ***RhythmicNet***, a system designed to generate rhythmic music for any human activity depicted in videos. ***RhythmicNet***'s initial phase involves extracting key points from movement videos and employing a spatio-temporal graph convolutional network [Re63] alongside a transformer encoder [Re19] to estimate music beats, capturing the rhythm inherent in human motion. To accommodate the periodic nature of music beats and the dynamic visual changes in human movements, we introduce a supplementary stream to capture rapid movements, termed the style stream. The fusion of these streams defines the movement rhythm, guiding the subsequent music generation phase, Rhythm2Drum. This phase utilizes an encoder-decoder

transformer to produce drum hits and a U-net [Re58] for determining drum velocities and offsets, both crucial for high-quality drum music creation. The final phase, Drum2Music, completes the composition by generating a piano or guitar track based on the drum performance using a transformer-XL [Re64] encoder-decoder architecture. ***RhythmicNet***'s overview is illustrated in Figure 4.1. Experimental validation on large-scale dance and diverse internet videos demonstrates that ***RhythmicNet*** successfully generates music that coherently aligns with human movements in videos.

4.2 Methods

RhythmicNet is structured around three meticulously designed components, each following sequentially to create a harmonious blend of visual movement and musical rhythm. These components are 1) **Video2Rhythm**: This initial phase focuses on associating the rhythm of music with human movements captured in video. It lays the groundwork by analyzing video content to identify and extract the inherent rhythm of the depicted activities, establishing a direct correlation between visual motion and musical tempo. 2) **Rhythm2Drum**: Building upon the rhythm identified in the first stage, this component is tasked with generating a drum track that mirrors the extracted rhythm. It translates the rhythmic patterns into a series of drum beats, creating a foundational drum track that encapsulates the tempo and energy of the human movements. 3) **Drum2Music**: The final stage enriches the drum track by adding layers of musical instruments, such as piano or guitar. This process not only complements the drum beats but also elevates the overall musical piece, adding depth and complexity to the initial rhythmic foundation.

Below, we delve into the specifics of each stage, outlining the technical approaches and innovative methodologies employed to seamlessly integrate rhythmic music with human movements, culminating in a cohesive and dynamic musical experience.

Video2Rhythm. In our approach, we dissect the concept of rhythm into two distinct elements: beats and style. This allows for a comprehensive analysis and synthesis of musical rhythm corresponding to human movements. We introduce an innovative model dedicated to

predicting music beats and employ a kinematic offsets methodology to capture style patterns derived from human activities.

Music Beats Prediction. The core of music beat prediction lies in recognizing beats as binary, periodic signals that follow a consistent tempo. This recognition is facilitated by a specialized music beat prediction network. This network is trained to learn the beat pattern by correlating body keypoints with actual music beats in a supervised manner. To achieve this, we utilize the OpenPose framework [Re10] to extract 2D skeleton keypoints from human body movements within videos. These keypoints undergo a first-order difference calculation to determine the velocity for each sequence in the video. The motion sequences are then represented as a three-dimensional tensor $X \in \mathbb{R}^{V\times T\times 2}$,where V denotes the number of keypoints, T represents the number of frames, and the third dimension corresponds to the 2D coordinates. The challenge of predicting music beats from human body movements is formulated as a temporal binary classification problem. Given the skeleton keypoints X, our goal is to produce an output sequence $Y \in \mathbb{R}^T$ of the same length, where each frame is classified as either a 'beat' ($y = 1$) or 'non-beat' ($y = 0$).

To encapsulate the dynamics of human movements, we utilize a spatio-temporal graph convolutional neural network (ST-GCN) [Re63] for encoding keypoints. This approach conceptualizes the skeleton sequence as an undirected graph $G = (V, E)$, with each node $v_i \in V$ representing a key point on the human body and the edges E delineating the connectivity among these keypoints. Initially, a spatial GCN processes the sequence to extract features from each frame independently. Subsequent application of temporal convolution integrates these features over time, synthesizing a comprehensive representation of motion dynamics. The resulting motion features are denoted as $P = AXW_SW_T \in \mathbb{R}^{V\times T_v\times C_v}$, where X is the input sequence, $A \in \mathbb{R}^{V\times V}$ signifies the adjacency matrix—constructed based on the connectivity of body keypoints—and W_S and W_T are the weight matrices corresponding to spatial and temporal convolutions, respectively. T_v and C_v represent temporal dimensions and feature channels. By averaging the node features, we obtain the final motion features $P \in \mathbb{R}^{T_v\times C_v}$.

Upon deriving the motion feature P, we employ a transformer encoder composed of several multi-head self-attention layers to discern correlations across different frames. To enhance the model's ability to capture the periodic nature of music beats, we incorporate two pivotal components: 1) Relative Position Encoding [Re65]: This technique allows the model to precisely gauge the relative distances between sequence tokens, eschewing traditional positional sinusoids in favor of an encoding that emphasizes relative timing differences—a crucial aspect in music where timing nuances are paramount. 2) Temporal Self-Similarity Matrix (SSM): Proven effective in human action recognition, the SSM aids in regularizing the transformer and identifying periodic movement repetitions [Re66]. It is constructed by calculating pairwise similarities $S_{ij} = f(P_i, P_j)$ between frame-level motion features P_i and P_j, utilizing the negative squared Euclidean distance as the similarity function, $f(a, b) = -||a - b||^2$, and applying softmax over the time dimension. After convolution $\hat{S} = \text{Conv}(S)$, SSM is integrated into each attention head, enhancing the model's attention mechanism implemented as

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T + \hat{S} + R}{\sqrt{D_k}})V,$$

where Q, K, V are the standard query, key, and value, respectively, and R is the ordered relative position encoding for each possible pairwise distance among query pairs and key on each head.

The training of the model leverages weighted binary cross-entropy loss, assigning additional weight to the beat category to mitigate class imbalance. The network outputs a beat activation function, predicting the likelihood of each video frame being a 'beat' frame. To precisely identify beat positions, we apply an algorithm inspired by Hidden Markov Model (HMM) decoding [Re67], ensuring accurate beat detection aligned with the dynamics of human movement. This methodology allows us to directly map the dynamics of human motion to the rhythmic structure of music, facilitating the generation of beats that are in perfect harmony with the visual content.

Style Extraction. While *beats* form the backbone of rhythm through their monotonic, peri-

odic patterns at fixed intervals, rhythm also encompasses a range of a-periodic components. Specifically, the spaces between music beats are often filled with various irregular movements that contribute to the overall rhythm. Unlike beats, these patterns are inconsistent, and systematically extracting them from visual information presents a challenge. To address this, we introduce an additional construct known as the style stream, which captures instances of transitional movements in the human body, such as rapid or sudden movements.

Given the implicit nature of style and the absence of direct data to learn from, we adopt a rule-based approach for predicting these events. The style is conceptualized as a binary stream, marking transitional time points with 1 and non-transitional points with 0. This stream is constructed through a series of steps centered around the spectral analysis of kinematic offsets in motion [Re68].

The initial step involves computing kinematic offsets, which are 1D time series signals representing the average acceleration of the human body over time. To derive these offsets, we calculate the motion's Directogram by dissecting it into various angles, using the formula $F_t(j,t)$ to denote the velocity magnitude of joint j at time t, and formulating the Directogram $D(t,\theta)$ [Re69] as:

$$D(t,\theta) = \sum_j F_t(j,t)\mathbb{1}_\theta(\angle F_t(j,t)), \text{ where } \mathbb{1}_\theta(\phi) = \begin{cases} 1 & |\theta - \phi| \leq 2\pi/N_{\text{bins}} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

An indicator function $\mathbb{1}_\theta(\phi)$ distributes the motion across N_{bins} angular intervals, and the first-order difference of the Directogram is computed to ascertain the acceleration of motion across different angles. The mean acceleration in the positive direction serves as an indicator of motion strength—higher values signify greater motion strength—and this measure is what we refer to as kinematic offsets.

Upon obtaining the kinematic offsets, we proceed to perform a Short-Time Fourier Transform (STFT) on these signals to pinpoint peaks in acceleration change. The most significant transitions in the signal are represented by the highest frequency bin in the STFT analysis (chosen from 8 bins), from which we extract style patterns. Peaks are marked with a 1,

while other points are marked with a 0, and the resulting signal is resampled to match the sampling rate of the music beats.

Rhythm Composition. The culmination of this process is a unified rhythm signal that merges the *beats* and *style* streams. This combined signal aims to reflect the correlation between body movements and the soundtrack's tempo, offering a comprehensive representation of rhythm that incorporates both periodic beats and the nuanced, a-periodic style elements.

Rhythm2Drum. In the **Rhythm2Drum** stage, we translate the rhythm obtained from the preceding phase into tangible drum sounds. This process is inspired by the GrooveVAE framework [Re6], within which each drum track is delineated by three distinct matrices: hits (H), velocities (V), and offsets (O). The Hits (H) is a binary matrix $H \in \mathbb{R}^{N \times T}$ indicates the presence of drum onsets, where N represents the number of drum instruments and T denotes the time steps (corresponding to sixteenth notes). Velocities (V) is a continuous matrix representing the intensity of drum strikes, with values ranging from $[0, 1]$. Offsets (O) is another continuous matrix capturing the timing deviations for each note, with values spanning $[-0.5, 0.5)$. These deviations indicate the temporal distance and direction of each note's timing from the nearest sixteenth note.

With $Y \in \mathbb{R}^{1 \times T}$ as the input rhythm sequence, our goal is to accurately generate the matrices H, V, and O. We approach this modeling challenge in two main steps, leveraging an encoder-decoder transformer [Re19] and a U-net [Re58] for a smooth transition from rhythm to drum sounds. In step 1, The binary rhythm sequence is fed into the transformer encoder, which processes the rhythm information. The decoder then transforms the H matrix into a word sequence based on a defined vocabulary of hit combinations, subsequently reconverting it into a binary matrix. This autoregressive learning approach enables the transformer to produce more authentic and varied drum onsets. The transformer is refined using cross-entropy loss. In step 2, We infuse style patterns—velocity and offsets—into the generated onsets. Given the identical dimensions of H, V, and O, this process is akin to image transformation, where H serves as the 'image' input. Utilizing a U-net [Re58], we transform the onset matrix H to generate the matrices V and O, optimizing the U-net with

Mean Square Error (MSE) loss. The culmination of this process involves converting the constructed matrices H, V, and O into the MIDI format to realize the drum track. This innovative methodology ensures a seamless and nuanced conversion of rhythm sequences into comprehensive drum tracks, embodying the rhythm's essence in a musically coherent and dynamic form.

Drum2Music. In the final stage of our process, we aim to enrich the soundtrack by incorporating additional instruments, building upon the rhythmic foundation provided by the drum track. To achieve this, we introduce an encoder-decoder architecture designed to generate tracks for supplementary instruments conditioned on the previously generated drum track. We focus on piano and guitar as our instruments of choice due to their prominence and versatility in music composition.

For representing multi-track music, we opt for the Remi representation [Re5], which offers a more comprehensive dataset than traditional MIDI-like event formats [Re4] by including details like tempo changes, chords, positions, and bars. This richness allows our model to grasp the complex dependencies between note events across different musical sections.

Both the encoder and decoder in our setup are based on the Transformer-XL model, which enhances the standard transformer by incorporating a recurrence mechanism [Re64]. This addition enables the model to access information from past tokens not limited to the current training segment, thus broadening its historical context for better prediction accuracy.

The encoder processes the drum track input through multiple self-attention layers, with each output E_i being a function of the input x_i and the encoder memory M_i^E, which stores the hidden state sequences from previous recurrent steps for the i-th bar ($E_i = \mathrm{Enc}(x_i, M_i^E)$). Similarly, in the decoder, the prediction for each token $y_{i,j}$ within the i-th bar is a function of previously generated tokens within the same bar $y_{i,t<j}$, the decoder memory M_i^D for the i-th bar, and the encoder output E_i for that bar ($y_{i,j} = \mathrm{Dec}(y_{i,t<j}, M_i^D, E_i)$). The decoder architecture includes layers of causal self-attention, cross-attention to the encoder outputs, and a feed-forward network.

For training, we segment the music into parts defined by the number of bars. The i-th

Models\Metrics	CML_c (%)	CML_t (%)	Cem (%)	F (%)
TCN [Re71]	44.97	45.15	48.14	63.04
TF	16.07	16.24	32.85	46.90
ST-GCN	54.89	55.45	49.23	64.78
ST-GCN+TF	61.89	62.34	55.09	71.93
ST-GCN+TF+SSM	63.20	63.58	57.72	73.07
ST-GCN+TF+RelAttn	68.01	68.31	59.19	74.67
ST-GCN+TF+SSM+RelAttn	**71.43**	**71.94**	**61.59**	**75.79**

Table 4.1: Music beat prediction evaluation. The abbreviation of each component stands for: TF (transformer), ST-GCN (spatio-temporal graph convolutional network), SSM (Self-similarity Matrix), RelAttn (Relative Attention); F (F-score measure), Cem (Cemgil's score), CML_c (Correct metrical level continuous accuracy), CML_t (Correct metrical level total accuracy). Bold font indicates the best value. (Table from [KS2])

bar of drum performance x_i is fed into the transformer-XL encoder at each recurrent step. Utilizing a teacher-forcing strategy, we input the ground truth tokens into the decoder to facilitate the generation of subsequent tokens, minimizing the negative log-likelihood (NLL) between the generated and ground truth tokens to refine the model.

During inference, the encoder receives the drum track for each bar, and the decoder generates the tokens sequentially. We employ a temperature-controlled stochastic top-k sampling method [Re70] to introduce randomness into generating new music tracks, ensuring diversity and creativity in the output. This methodical approach allows us to expand the musical piece with intricate layers of additional instruments seamlessly integrated with the foundational drum track to produce a rich, multi-dimensional musical experience.

4.3 Experiments

Datasets. For the development and evaluation of our **Video2Rhythm** component, we utilize the AIST Dance Video Database [Re72], an extensive repository of high-quality dance videos

Model\Metric (lower better)	NDB	MSE Velocity	MSE Offsets
GrooveVAE [Re6]	46	0.0437	0.0402
TF multi-outputs w.o. hits sequence	44	0.0507	0.0348
TF multi-outputs w. hits sequence	39	0.0493	0.0369
TF w. hits sequence + Unet	**39** (↓15%)	**0.0267** (↓40%)	**0.0169** (↓58%)

Table 4.2: **Rhythm2Drum** performance evaluation. Abbreviations stand for TF (encoder-decoder transformer), Multi-outputs (Predict the hits, velocities, and offsets simultaneously), w./w.o. hits sequence (whether using word tokens to represent the hits). Bold font indicates the best value. (Table from [KS2])

recorded at 60 frames per second. This rich dataset provides a diverse array of movements and rhythms, making it an ideal resource for training and testing our rhythm analysis algorithms.

In the **Rhythm2Drum** stage, our training material comes from the Groove Midi dataset [Re6], which comprises 1150 MIDI files, encompassing over 22,000 measures of varied drumming patterns. This comprehensive collection of drum sequences offers a wide range of rhythmic styles and complexities, serving as an excellent foundation for our drum track generation model.

For the final **Drum2Music** phase, we specifically curate two subsets from the Lakh MIDI dataset [Re73] to train our **Drum2Piano** and **Drum2Guitar** models. These subsets are carefully selected to encompass a broad spectrum of musical pieces that feature piano and guitar, respectively. This targeted approach allows our models to specialize in generating accompanying music tracks that are stylistically coherent and musically rich, tailored to complement the underlying drum tracks effectively.

Video2Rhythm Evaluation. In alignment with established standards for evaluating musical beat tracking [Re74], we assess our model's performance using several key metrics: the F-score measure, Cemgil's score (Cem), and the Correct Metrical Level (CML) continuity, distinguishing between cases where continuity is required and not required ($\text{CML}_{c/t}$). To

Metrics	PC/bar	PI	IOI	PCH ↑	NLH ↑	NLL ↓
Dataset (Piano)	5.48	6.16	0.31	-	-	-
Drum2Piano w.o. memory	7.17	4.63	0.12	0.63	0.52	0.77
Drum2Piano	**6.82**	**5.86**	**0.14**	**0.63**	**0.54**	**0.53**
Dataset (Guitar)	5.33	5.51	0.22	-	-	-
Drum2Guitar w.o. memory	3.54	8.94	0.52	0.56	0.46	0.58
Drum2Guitar	**5.63**	**5.69**	**0.13**	**0.64**	**0.51**	**0.40**

Table 4.3: Drum2Music evaluation. For PC/bar, PI, and IOI values, the closer to the dataset, the better. For PCH and NLH values, the larger, the better. (Table from [KS2])

facilitate a comparative analysis with existing methodologies, we implement a baseline model employing a Temporal Convolutional Network (TCN) specifically for beat prediction [Re71].

The comparative and ablation study results are detailed in Table 4.1. Notably, our premier Video2Rhythm approach, which integrates a Spatio-Temporal Graph Convolutional Network (ST-GCN) with Transformer (TF), Self-Similarity Matrix (SSM), and Relative Attention (RelAttn), demonstrates superior performance across all evaluated metrics, surpassing the baseline TCN model by a significant margin. This is especially evident in the continuity scores, where our method exceeds the baseline's performance by more than 25%. Such a result underscores the exceptional consistency of the beat sequence estimated by our model, highlighting its effectiveness in capturing the rhythmic essence of the music in a more coherent and sustained manner compared to conventional approaches.

Rhythm2Drum Evaluation. To rigorously assess the performance of our **Rhythm2Drum** component, we employ a variety of metrics that cater to different aspects of drum track generation. To gauge the diversity within the generated drum hits, we utilize the Number of Statistically-Different Bins (NDB) metric, a method previously proposed and implemented in other studies [Re75, Re22, Re39]. This metric helps us quantify the variation in our drum samples, a crucial factor for ensuring the richness and authenticity of the generated rhythms.

For a detailed analysis of the velocities and offsets within our drum tracks, we calculate the Mean-Squared Error (MSE) across the test set. This measurement allows us to evaluate the precision of our model in capturing the nuances of drumming dynamics.

Our evaluation includes a comparison with the baseline model, GrooveVAE [Re6], a well-regarded benchmark in the field of automated drum track generation. The comparative results are summarized in Table 4.2, illustrating the effectiveness of our methods.

The findings reveal that our approach, which leverages sequences of drum hits as the foundation for drum track generation, fosters a significantly more diverse collection of samples. This diversity, in turn, enhances the subsequent U-net component responsible for generating velocities and offsets, ultimately leading to the production of more realistic and nuanced drumming sounds. This methodology not only surpasses the baseline in terms of sample variety but also demonstrates the importance of a diverse hit sequence as a precursor to achieving high-quality, lifelike drum tracks.

Drum2Music Evaluation. To assess the quality of the generated piano and guitar tracks, we employ a set of objective metrics that include Pitch Count per Bar (PC/bar), Average Pitch Interval (PI), and Average Inter-Onset Interval (IOI), as delineated in [Re76]. These metrics allow us to systematically evaluate the musical properties of the generated tracks by comparing their statistical profiles against those derived from the test dataset.

Additionally, we incorporate the Pitch Class Histogram (PCH) and Note Length Histogram (NLH) metrics, for which we calculate the Overlapping Area (OA) between the test dataset's statistics and those of the generated music. This calculation is performed for each sample, and we report the average OA to provide a holistic view of the similarity between the generated music and the authentic test data.

Furthermore, we analyze the Negative Log-Likelihood (NLL) loss on the validation set to gauge the model's predictive accuracy. The numerical outcomes of these evaluations are presented in Table 4.3, facilitating a direct comparison between two model variants: one incorporating a recurrence mechanism (with memory) and the other without.

The comparative analysis reveals that for both the Drum2Piano and Drum2Guitar tasks,

	Soundtrack Preference			
	No Soundtrack	Drums Only	Drums + Piano	Drums + Guitar
votes	7.3%	31.2%	32.1%	29.4%

Table 4.4: Soundtrack preference.(Table from [KS2])

Soundtrack match to the video			Soundtrack match to the video (Ablation)		
Random	Shuffle	RhythmicNet	Random + GrooVAE	Video2Rhythm + GrooVAE	Video2Rhythm + Rhythm2Drum
30.8%	27.8%	41.4%	23.3%	33.3%	43.4%

Table 4.5: Soundtracks match to movements in the video. (Table from [KS2])

the models equipped with a recurrent encoder-decoder transformer (i.e., with memory) exhibit lower NLL loss on the validation set. Moreover, the statistical properties of the music generated by these models align more closely with those of the test dataset compared to the versions without memory. This indicates the significant impact of the recurrence mechanism in enhancing the model's ability to generate musically coherent tracks that faithfully mirror the characteristics observed in genuine music samples, underscoring the importance of memory in achieving higher fidelity and authenticity in music generation.

Human Perceptual Evaluation of Soundtrack Music.

Beyond objective evaluations of ***RhythmicNet***'s various components, we conducted human perceptual surveys via Amazon Mechanical Turk to assess the subjective impact of ***RhythmicNet***-generated soundtracks. These surveys aimed to determine how well the soundtracks aligned with video movements and whether they enhanced viewers' overall perception of the videos, compared to a set of soundtrack controls.

In our initial survey, participants (non-experts) were asked to select their preferred video from a set that included one without a soundtrack and three with soundtracks generated by ***RhythmicNet*** (either drums-only or drums accompanied by another instrument). The

results, as outlined in Table 4.4, demonstrated a clear preference for videos accompanied by a soundtrack. Interestingly, preferences for the type of instrumentation in the soundtrack were almost evenly distributed among the three variations offered, with a slight majority favoring tracks that featured Drums+Piano.

The second survey posed the question: "In which video does the sound best match the movements?" Participants were given options featuring soundtracks with Random, Shuffle, and RhythmicNet rhythms. The "Random" option involved a drum track generated by the Rhythm2Drum method with a rhythm having a 50% chance of being ON or OFF at any given time step. The "Shuffle" drum track was also produced using the Rhythm2Drum method, but with the rhythm and order shuffled. The RhythmicNet option represented a drum track created using the Rhythm2Drum method without alterations. Results from this survey, shown in Table 2 4.5 (left), revealed a strong preference for drum tracks generated by our method as the best match for the video movements, with 41.4% selecting RhythmicNet, compared to 30.8% for Random and 27.8% for Shuffle.

An extra survey conducted a perceptual ablation study to examine the influence of the Video2Rhythm and Rhythm2Drum components on soundtrack perception relative to baseline methods. The findings, presented in Table 4.5 (right), indicated that these components significantly enhance the perception of the soundtrack when compared to baseline approaches, affirming their effective contribution to the overall auditory experience of the videos.

These human perceptual surveys underscore the effectiveness of ***RhythmicNet*** in producing soundtracks that not only align with video movements but also substantially improve the viewers' enjoyment and engagement with the content, highlighting the nuanced understanding and application of rhythm in enhancing multimedia experiences.

Chapter 5

MUSIC FROM ARBITRARY VIDEOS

5.1 Motivation

Building on the foundation laid in previous chapters for video-to-music generation, which primarily focused on capturing the audio-visual correspondence between specific video representations and the symbolic representation of music, we recognize a notable limitation: the input video types are confined to particular visual scenarios, thus hindering generalization to arbitrary video inputs such as cat videos or image slideshows. Addressing this limitation, in this chapter, I discuss an approach that expands visual-driven music generation to accommodate any type of video content. This objective presents a formidable challenge, as it necessitates moving beyond the intrinsic physical correspondence, such as instrumental play or human motion rhythm.

Indeed, videos can be effectively paired with music that complements and enriches the visual content, even in the absence of direct physical correspondence. Achieving semantic mapping between such diverse video content and music requires extensive paired data. However, the symbolic representation of music, a cornerstone in prior successful methodologies, is not readily applicable in this broader context, further complications of the task of generating high-quality music audio.

In this chapter, we introduce a novel strategy to learn general audio-visual correspondence directly from paired video frames and music waveform data. Our approach endeavors to bridge the gap between coarser video representations and the detailed audio domain by identifying a mutually aligned low-resolution representation space. This methodology facilitates adaptation to a broad spectrum of video inputs and also harnesses the vast reservoir of parallel music and video data available online for scaling purposes.

The work is titled ***V2Meow*** and represents a novel music audio waveform generator that is conditioned on a wide array of video inputs. Inspired by the breakthroughs of MusicLM [Re77], ***V2Meow*** adopts a multi-stage autoregressive language modeling framework. Accepting video frames as input, ***V2Meow*** allows for style modulation via textual prompts, integrating video and text into a cohesive input stream processed by a Transformer equipped with feature-specific adaptors. Through the explicit modeling of domain-specific audio-visual correspondences, ***V2Meow*** allows for zero-shot transfer capabilities as shown in evaluations of the model for AIST++ dance videos [Re78].

Through both quantitative and qualitative analyses of music-video correspondence, complemented by an ablation study, we elucidate the key factors that influence the quality of generated content. Our findings, supported by numerical data and human-centric studies, confirm that V2Meow achieves a closer alignment with human music preferences compared to MIDI-based benchmarks. Furthermore, incorporating video input significantly enhances ***V2Meow***'s ability to learn visually relevant features, surpassing the outcomes of text-only inputs. An overview of V2Meow's architecture and capabilities is provided in Figure 5.1, showcasing its potential to revolutionize the field of visual-driven music generation.

5.2 Methods

Audio Representations. For the generation of audio waveforms, we adopt a methodology inspired by AudioLM [Re79], leveraging semantic and acoustic tokens derived from two distinct pre-trained self-supervised models. The semantic tokens are sourced from a pre-trained w2v-BERT model [Re80], a self-supervised architecture that utilizes masked language modeling (MLM) with contrastive loss. This model adeptly captures a wide array of audio characteristics, from local elements like melody to broader aspects such as harmony and rhythm, by analyzing both short-term and long-term structural dependencies within the audio.

To extract semantic tokens, we initially procure embeddings from an intermediate layer of the w2v-BERT model [Re80]. Subsequently, we employ the k-means clustering algo-

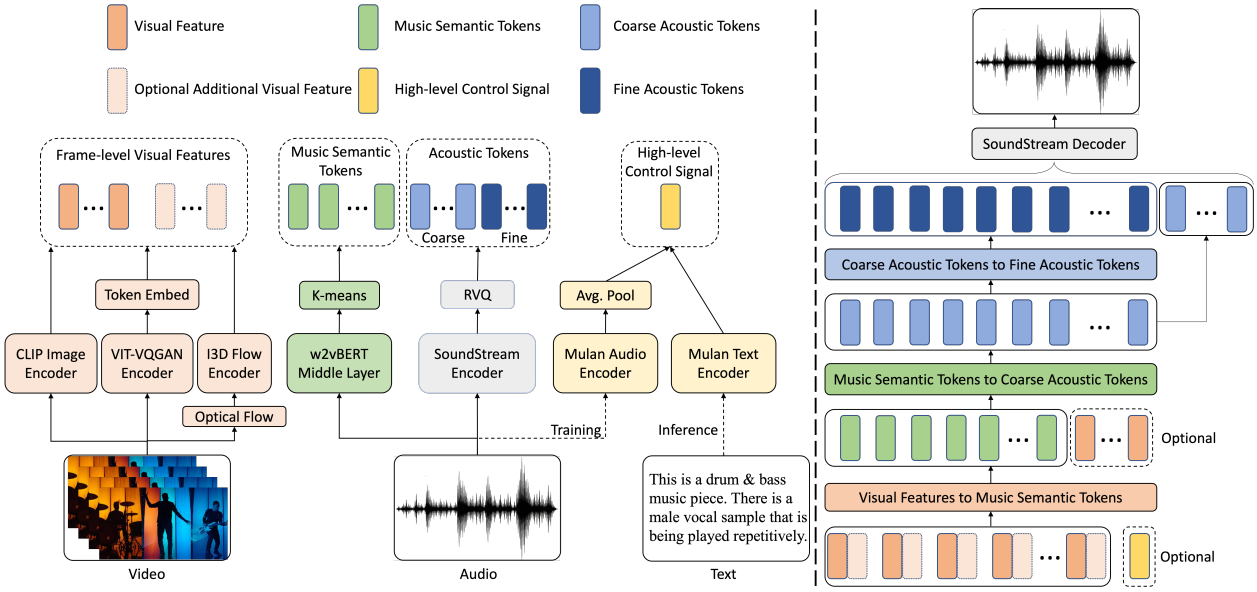

Figure 5.1: ***V2Meow*** Overview: (left) Feature extraction pipeline for video, audio, and text representations. (right) Overview of multi-stage video to music modeling.

rithm, with a predetermined number of clusters K_s on these embeddings. The indices of the centroids obtained through this process serve as our semantic tokens, yielding a series of semantic tokens $\{S_t : t = 1, ..., T_s\}$ for each audio waveform, where T_s represents the total count of tokens. Although the relatively coarse resolution of these semantic tokens facilitates the modeling of long-term dependencies—thereby enhancing the coherence of generated content over extended periods—the direct reconstruction of audio from solely these tokens often results in suboptimal quality.

To surmount this limitation and achieve high-quality music audio generation, we further incorporate acoustic tokens extracted from the pre-trained SoundStream model [Re8]. SoundStream, functioning as a universal neural audio codec, efficiently compresses and subsequently reconstructs arbitrary audio at low bit rates without compromising quality. The model employs a convolutional encoder to embed the input waveform, which is then discretized through residual vector quantization (RVQ), allowing each quantizer (with a vocab-

ulary size of K_a) to learn the quantization of the embedding concurrently during the training phase.

Given the pivotal role of coarser levels in ensuring high-fidelity audio reconstruction—as evidenced by the findings in AudioLM—we initially establish a mapping from the semantic tokens of music to coarser acoustic tokens. This strategy is followed by a subsequent phase where we refine the mapping process, transitioning from coarse acoustic tokens to their more detailed counterparts. This two-tiered approach enables the precise modeling of audio nuances, ensuring the generation of music audio that not only exhibits high quality but also maintains semantic integrity with the input video content.

Visual Representations. Given a video comprised of T frames, denoted as $v_t \in \mathbb{R}^{H\times W\times 3} : t = 1, ..., T$, our goal is to distill relevant visual features using existing pre-trained visual models. The quest for the most informative visual representation for music generation leads us to explore a variety of models, spanning pure visual, multimodal, and quantized domains, given the ambiguity surrounding which type of visual representation could most effectively inform the music generation process. Specifically, we investigated the following categories of visual features:

Pure visual representations. Acknowledging the commonality between visual changes in a video and musical rhythm, we employ the Inflated 3D (I3D) model [Re81], renowned for its ability to capture optical flow—a critical component for motion analysis. For our purposes, the visual flow embeddings are represented as $\{f_t \in \mathbb{R}^{D_f} : t = 1, ..., T\}$, extracted from an I3D model pre-trained on Kinetics [Re81], where D_f indicates the dimensionality of the I3D features.

Multimodal embeddings. We also consider the use of CLIP, a text-image joint embedding model, anticipating its broad applicability and robustness in capturing video semantics. The CLIP embeddings are denoted as $\{c_t \in \mathbb{R}^{D_c} : t = 1, ..., T\}$, with D_c indicating the dimensionality of the CLIP embedding.

Visual tokens. In alignment with the discrete token-based representations employed for semantic and acoustic aspects of music, we investigate the application of analogous dis-

crete tokens for visual inputs. Utilizing the ViT-VQGAN [Re82] model—a leading self-supervised Vision Transformer (ViT) that performs image quantization—each video frame $v_t \in \mathbb{R}^{H\times W\times 3}$ is encoded into a matrix of discretized latent codes sized $H/D_q \times W/D_q$, where D_q represents the dimension of non-overlapping image patches corresponding to a single token. Consequently, a video consisting of T frames is articulated as a set of tokens $\{Q_t \in \mathbb{Z}^{H/D_q\times W/D_q} : t = 1, ..., T\}$.

Control Signal Representations. Recognizing that musical preferences for a video can vary significantly among individuals, we've designed our system to accommodate optional user inputs in the form of music-related text descriptions alongside visual inputs. This feature empowers users to influence the overarching theme or mood of the generated music. However, sourcing music-video-text triplet data from natural settings poses a substantial challenge. To address this, we turn to the MuLan model, a music-text joint embedding model trained on paired music and text data through contrastive loss, enabling us to bridge the gap between music and textual descriptions effectively.

For each video within our training dataset, ***V2Meow*** systematically extracts MuLan embeddings [Re83] from all audio segments $m_j \in \mathbb{R}^{D_m} : j = 1, ..., J\}$, where each segment spans ten seconds, and J represents the total count of such segments within a video. The dimensionality of the MuLan audio embedding is set at $D_m = 128$. To simplify and consolidate these embeddings, we compute their average to represent each video at a singular, unified level. During the inference phase, should a music-related text description be provided, we obtain a corresponding MuLan embedding to serve as a conditioning factor for our ***V2Meow*** model. This approach not only enriches the model's capacity to generate music that aligns with the user's preferences but also showcases the flexibility and adaptability of ***V2Meow*** in producing music that complements the video content, guided by high-level descriptive inputs. Through leveraging the power of music-text joint embeddings, we ensure a more personalized and nuanced music generation process, catering to the diverse tastes and expectations of users.

Model Pipeline. Sequence-to-sequence modeling tasks in our context unfold across three

pivotal stages, each contributing uniquely to the process of generating music that authentically reflects the accompanying video content.

Stage 1. Visual Features to Music Semantic Tokens. At this foundational stage, ***V2Meow*** embarks on learning the intricate mapping from visual inputs to music semantic tokens. Employing an encoder-decoder Transformer architecture [Re19], the encoder processes visual features, while the decoder, operating autoregressively, predicts the music semantic tokens. This stage is crucial for establishing a bridge between the visual and audio modalities, facilitating the semantic transition from visual to audio information. Although it doesn't yield high-quality, fine-grained audio, this phase is instrumental in aligning the two modalities, laying the groundwork for semantic understanding.

Stage 2. Music Semantic Tokens to Coarse Acoustic Tokens. The objective here is to refine music semantic tokens into acoustic tokens suitable for high-quality synthesis. Adapting the AudioLM framework, we bifurcate this stage into coarse and fine acoustic modeling processes. Two distinct training strategies are explored: i) Following AudioLM, a decoder-only Transformer is trained to transition music semantic tokens to coarse acoustic tokens, leveraging vast quantities of music data for robust pre-training without necessitating visual inputs. ii) We also investigate the potential benefits of visual conditioning in this stage by employing an encoder-decoder Transformer model that combines visual features and music semantic tokens to predict coarse acoustic tokens. While this method theoretically promises enhanced performance, the limited availability of music-video pairs compared to music-only data poses challenges to its efficacy.

Stage 3. Coarse to Fine Acoustic Tokens and Audio Decoding. Having established coarse acoustic tokens, we proceed to refine these tokens from the first N_c levels of SoundStream's Residual Vector Quantization (RVQ) to the subsequent N_f levels, facilitating a transition from coarse to fine acoustic detail. Ultimately, these comprehensive levels of tokens are processed through the SoundStream Decoder to reconstruct the audio waveform.

Adding Control. To embed a control signal within V2Meow during training, we introduce the MuLan audio embedding as an ancillary input to the Transformer encoder alongside visual

features right from the first stage, ensuring both inputs share a uniform feature dimensionality. During inference, we pivot to incorporating the MuLan text embedding with visual features to guide the generation of semantic tokens, offering a nuanced layer of control over the resulting music based on user-provided textual descriptions. This structured approach not only enriches the model's ability to generate music that resonates with the visual content but also empowers users to steer the music generation process according to their preferences, enhancing the overall creativity and flexibility of V2Meow.

5.3 Experiments

Training Datasets. Following [Re84], we filtered a publicly available video dataset [Re85] to 110k videos with the label Music Videos and referred to it as MV100K. The training and validation datasets were split into an 80:20 ratio. We trained the Stage 1 model and Stage 2 model on 5k hours of music videos. A version of the Stage 2 model is trained on audio-only data for the ablation study. For computing semantic and acoustic audio tokens, we adopt the SoundStream tokenizer and w2v-BERT tokenizer, both of which are pre-trained on 46k hours of music-only audio data sampled at a 16kHz sampling rate.

Evaluation Datasets. We evaluate our methods on three different datasets. For the task of video conditional music generation, we use the test partition of the MV100K. We select 13 genres of music videos to comprise a genre-balanced subset with a total number of 4076 videos. For the task of video and text conditional music generation, we use the latest MusicCaps dataset [Re77], which is a subset of AudioSet [Re86]. The MusicCaps has about 5.5k human annotated text captions, music, and video pairs. With the text caption, we can verify whether the generated music could be controllable and whether its performance is comparable with text-to-music generation models like MUBERT [Re87] and Riffusion [Re88]. For both tasks, we generate ten-second audio for each video clip. For the dance-to-music generation task, we evaluate temporal alignment on 20 dance videos in the test split of AIST++ [Re78]. The evaluation is in zero-shot fashion without any fine-tuning on the AIST++ train split, and only video frames are used for modeling while no motion

data is involved. The reported metrics represent averages over 20 10-second audio segments and 86 2-second audio segments, with 5 inference examples per segment.

Evaluation Metrics. To objectively assess the fidelity, semantic relevance, and rhythmic alignment of the generated music samples, we adopt a suite of quantitative metrics, drawing inspiration from the approaches used in MusicLM [Re77] and [Re89, Re90]. The objective metrics include 1) Fréchet Audio Distance (FAD): We measure audio quality using FAD, utilizing two publicly available audio embedding models: TRILL[1] [Re91] (trained on speech data) and VGGish[2] [Re92] (trained on a public audio event dataset). These models allow us to evaluate distinct aspects of audio fidelity. 2) KL Divergence (KLD) and MuLan Cycle Consistency (MCC): To assess the semantic relevance of the generated music to the reference audio or text, we employ KLD [Re93, Re94] and MCC [Re77]. The LEAF classifier [Re95], applied to multi-label classification on AudioSet, enables us to compare predicted class probabilities between the original and generated audio, assessing conceptual similarity. For video-to-music tasks, MCC calculates the average cosine similarity between the MuLan audio embeddings of generated and reference audio. In video-and-text-to-music tasks, we use MuLan text embeddings as the reference to evaluate adherence to the text description. 3) Rhythmic Alignment: To quantify rhythmic alignment, we use Beats Coverage Scores (BCS) and Beats Hit Scores (BHS) [Re89, Re90], which measure the alignment of synthesized music with ground-truth music in terms of rhythm. Adjusted BCS and BHS metrics from [Re96] are employed, along with an additional F1 score calculation for a comprehensive evaluation.

Acknowledging the subjective nature of video and music matching, we conducted a human study to evaluate visual relevance and music preferences. We selected 89 video examples from the MV100K test set and 76 from the MusicCaps test set, surveying approximately 200 participants. Each participant reviewed a pair of videos—identical in visual content but differing in background music—with each video pair evaluated by three individuals, totaling 3500 ratings. 1) Visual Relevance: Participants compared music generated by baseline

[1] tfhub.dev/google/nonsemantic-speech-benchmark/trill/3

[2] tfhub.dev/google/vggish/1

models (CMT, Riffusion, or MUBERT) against five V2Meow variants, answering: "Which music do you think goes best with the video?" The focus was solely on the music-video match, disregarding sound quality. 2) Music Preference: This aspect of the study aimed to discern whether the generated music aligns with human perceptual preferences, asking raters to select their preferred piece of music while ignoring the video content. Here, sound quality was again not a consideration, focusing instead on personal music preferences.

Through this combination of objective metrics and human-centered evaluations, we aim to provide a holistic assessment of our model's ability to generate music that not only exhibits high fidelity and semantic relevance but also resonates with human preferences and appropriately complements video content.

Video Conditional Music Generation Results. Quantitative evaluation results, detailed in Table 5.1, provide insightful observations about the performance of our model across the MV100K dataset when conditioned on various visual embeddings. Notably, the Clip embedding achieves the highest MuLan Cycle Consistency (MCC) score, suggesting its effectiveness in capturing the semantic alignment between video and music. Conversely, the I3D flow embedding excels in Fréchet Audio Distance (FAD) metrics, indicating its prowess in reflecting the audio quality aspects of the video-music relationship.

This disparity in performance highlights the nuanced roles different visual features play in navigating the video-music aligned subspace. Remarkably, a combined approach utilizing both Clip and I3D Flow embeddings not only secures the top MCC score among all models but also shows a notable improvement in FAD VGGish metrics, outshining models relying on either visual input exclusively. Although VIT-VQGAN tokens alone do not lead in individual metrics, their integration with I3D Flow embedding enhances overall performance, underscoring the value of multi-modal inputs in capturing a broader spectrum of video-music correlations.

Video and Text Conditional Music Generation Results. In the evaluation using the MusicCaps dataset, our approach—significantly bolstered by incorporating video frames as an additional control—showcases a remarkable improvement in visual relevance by 20-30%

Method	Visual	Text	Semantic / Semantic + Acoustic Modeling			
			FAD TRILL ↓	FAD VGG ↓	KL Div. ↓	MCC ↑
Eval Dataset: MV100K						
CMT	✓	✗	N/A	N/A	N/A	N/A
Random Shuffle	✗	✗	-	-	0.67	0.268
V2Meow-CLIP	✓	✗	0.236/0.158	6.094/2.779	0.63/0.54	0.312/0.372
V2Meow-I3D	✓	✗	0.236/**0.151**	6.278/2.328	0.77/0.65	0.279/0.296
V2Meow-VIT	✓	✗	0.240/0.174	6.097/1.988	0.73/0.62	0.276/0.294
V2Meow-VIT+I3D	✓	✗	0.236/0.178	5.801/**1.945**	0.68/0.57	0.298/0.327
V2Meow-CLIP+I3D	✓	✗	0.235/0.165	6.126/2.003	0.64/**0.49**	0.343/**0.419**
Eval Dataset: MusicCaps						
CMT	✓	✗	N/A	N/A	N/A	N/A
Riffusion	✗	✓	0.760	13.4	1.19	0.34
MUBERT	✗	✓	0.450	9.6	1.58	0.32
V2Meow-CLIP	✓	✓	0.379/**0.328**	5.198/4.628	1.31/1.19	0.364/0.377
V2Meow-I3D	✓	✓	0.389/0.331	5.190/**4.623**	1.26/1.22	0.377/0.371
V2Meow-VIT	✓	✓	0.377/0.366	4.970/5.039	1.34/1.23	0.380/0.392
V2Meow-VIT+I3D	✓	✓	0.381/0.359	5.094/4.819	1.34/1.21	0.379/0.389
V2Meow-CLIP+I3D	✓	✓	0.391/0.349	5.385/4.948	1.27/**1.19**	0.369/**0.394**

Table 5.1: Quantitative evaluations on MV100K and MusicCaps for different models. For FAD and KL Divergence, lower is better. For MCC, higher is better. Bold font indicates the best value. The five V2Meow variants are named based on the video features used as input. Semantic Modeling indicates video conditioning is used only for semantic modeling, while Semantic + Acoustic Modeling indicates video conditioning is used for both semantic and acoustic modeling.

over Riffusion and MUBERT, which rely solely on textual input for conditioning. Notably, our method also secures lower Fréchet Audio Distance (FAD) scores and higher MuLan Cy-

Model (Length)	Beat Coverage	Beat Hit	F1
GT	100	100	100
V2Meow CLIP+I3D (10s)	100.0 (0.00)	84.4 (25.1)	91.5
V2Meow CLIP+I3D (6s)	99.3 (8.64)	84.7 (25.7)	91.4
V2Meow CLIP+I3D (2s)	90.0 (30.0)	84.8 (32.1)	87.3
CDCD Step-Intra (6s)	87.9	83.2	85.5
D2M-GAN (2s)	88.2	84.7	86.4
CMT (2s)	85.5	83.5	84.5

Table 5.2: Zero-shot evaluation results on AIST++. For CMT and V2Meow, only video frames are used as input, while CDCD Step-Intra and D2M-GAN require additional motion annotation as inputs. For each video input, we randomly generate 10 music samples and report the average score and standard deviation.

cle Consistency (MCC) scores. In this context, the MCC score represents the congruence between the generated music audio and the accompanying text, underscoring the effectiveness of video frames in enhancing audio quality and textual fidelity. This achievement is particularly impressive given the challenge of working with a comparatively modest dataset of 5,000 hours of music videos.

The strategic combination of Clip and I3D Flow embeddings emerges as a standout, yielding the highest scores in KL Divergence (KLD) and MCC, indicating superior performance in audio quality and text alignment. Meanwhile, the integration of VIT-VQGAN tokens with I3D Flow embeddings is distinguished by its exceptional contribution to visual relevance. Across diverse configurations, V2Meow consistently surpasses baselines across key metrics, including audio quality and adherence to text.

Dance to Music Generation Results. In the specialized domain of dance-to-music generation, we extend our comparative analysis to include V2Meow and baseline models such as

D2M-GAN [Re89], CDCD Step-Intra [Re90], and CMT [Re40], utilizing the AIST++ test split [Re78]. This comparison is particularly designed to scrutinize V2Meow's proficiency in interpreting complex dance movements. Remarkably, zero-shot evaluations conducted on dance videos from the AIST++ test split indicate that V2Meow achieves performances on par with those of dedicated dance-to-music generation models, as evidenced by metrics like beat coverage and beat hit. This assessment is notably executed in a zero-shot manner, eschewing any fine-tuning on the AIST++ training split and relying solely on video frames for modeling, without recourse to human-annotated motion data.

These findings underscore the versatility and robustness of our proposed framework, demonstrating its capability to accurately navigate videos characterized by significant rhythm variations and intricate dance movements, even at a sampling rate as low as 1fps. It's crucial to highlight that while D2M-GAN and CDCD Step-Intra benefit from fine-tuning on the AIST++ training split—requiring a higher sampling rate and the incorporation of additional motion annotations—our model, alongside CMT, operates exclusively on video frame input.

This comparison not only showcases V2Meow's advanced understanding of dance dynamics but also emphasizes its efficiency and effectiveness in generating music that aligns with the visual rhythm of dance, all without the need for extensive data preprocessing or specialized training. Through this approach, V2Meow demonstrates a significant advancement in the field of audio-visual synthesis, offering a streamlined and accessible solution for generating music that complements and enhances the visual experience of dance.

Subjective Evaluation Results. In our evaluation, we focused on two listening tasks across genre-balanced subsets extracted from the MV100K and MusicCaps datasets, aiming to provide a comprehensive numerical assessment. The findings, as detailed in Table 5.3 (left column), reveal a definitive preference for the music clips produced by our proposed models over those generated by the MIDI-based video-to-music generation baseline, CMT. This preference is attributed to our models' ability to generate music that more accurately matches the visual content, a clear testament to their effectiveness in capturing the nuanced interplay between audio and visual elements.

Eval Dataset	MV100K		MusicCaps	
Model\Metrics	Visual Relevance	Music Quality	Visual Relevance	Music Quality
CMT [Re40]	20.6%	30.0%	19.7%	20.7%
Riffusion [Re88]	N/A	N/A	38.6%	41.2%
MUBERT [Re87]	N/A	N/A	43.3%	49.3%
V2Meow-CLIP	78.2%	67.6%	63.6%	58.5%
V2Meow-I3D	74.3%	65.8%	68.8%	68.0%
V2Meow-VIT	81.4%	71.5%	66.9%	60.3%
V2Meow-VIT+I3D	83.8%	76.8%	71.8%	67.1%
V2Meow-CLIP+I3D	79.2%	68.2%	67.4%	65.8%

Table 5.3: Human perceptual evaluation on visual relevance and music quality metrics.

Moreover, in terms of music preference, our approaches stand out for their capacity to produce music characterized by finer details and more complex patterns. This contrasts with the MIDI baseline's tendency to generate music with simpler patterns, highlighting the advanced capabilities of our models in delivering a richer musical experience. The results presented in the right column of Table 5.3 further reinforce our models' superiority. Compared to text-to-music generation models like MUBERT and Riffusion, our generated music demonstrates a closer alignment with the video content. This enhanced alignment is attributable to the additional video conditioning employed in our models, underscoring the importance of integrating video inputs for achieving more contextually relevant and engaging music generation.

Chapter 6

SELF-SUPERVISED LEARNING TO SEPARATE HUMAN BODY MOVEMENTS

6.1 Motivation

Human body movements hold a profound connection with audio, as gestures and motions often coincide with various sounds and rhythms. From the graceful sway of a conductor's baton to the energetic tapping of a drummer's foot, these movements serve as visual cues that mirror the auditory experience. In dancing, the synchronization of movement with music creates a harmonious blend of sight and sound, where each step and gesture contributes to rhythmic expressions. Moreover, in everyday interactions, actions such as clapping, snapping fingers, or even footsteps generate distinct audio signals that accompany and enhance our visual perception. This intricate relationship between human movement and audio underscores the interconnections of sensory experiences and provides fertile ground for exploring innovative approaches to audio-visual understanding and generation.

Hence, the main aim of this chapter is to describe an approach that I have developed for unsupervised skeleton-based action recognition. Robust action recognition, particularly in the context of human actions, stands as a crucial capability for advancing audio-visual learning in various applications of computer vision and artificial intelligence. While existing methods have demonstrated ability in identifying basic actions within videos, they typically rely on extensive supervision with large datasets containing annotated action labels. Gathering and annotating such datasets poses significant challenges, especially for recognizing diverse types of actions across various applications. Moreover, annotation itself presents difficulties, as it often relies on subjective interpretation by annotators to assign meaningful labels to given sequences. Thus, our study delves into exploring unsupervised methods for

action recognition, aiming to overcome the limitations associated with data annotation and supervision.

For human action recognition, the time series of body joints (skeleton) tracked over time have proven to be effective descriptors of actions. In this study, we specifically focus on 3D skeleton time sequences and propose an unsupervised system called **PREDICT & CLUSTER** (P&C). This system operates by training an encoder-decoder network to simultaneously predict and cluster skeleton sequences, thereby learning an effective hidden feature representation of actions. By replacing the traditional classification supervised task with a non-classification unsupervised task, our approach aims to predict or regenerate the given sequence, guiding the hidden states to capture key action features. In the encoder-decoder architecture, the prediction task involves the encoder processing the action sequence, while the decoder continues or generates the sequence. Both the encoder and decoder are recurrent neural networks (RNNs) with hidden variables for each time sample. The final hidden state of the encoder represents the action feature, with the decoder training strategy significantly influencing its effectiveness. Two main decoder training strategies are proposed. The first strategy is a conditional strategy, where the output of the previous time-step of the decoder is used as the input for the current time-step. With such a strategy, the output of the decoder is expected to be continuous. In contrast, the unconditional strategy assigns a zero input to each time step of the decoder. Previous research suggested that unconditional training of the decoder leads to better prediction performance by effectively weakening the decoder, thereby forcing the encoder to learn a more informative representation.

In our system, we extend upon such strategies to bolster the encoder representation, thereby improving the clustering and organization of actions in the feature space. We introduce two novel decoder training strategies, Fixed Weights, and Fixed States, aimed at further penalizing the decoder. By implementing these strategies, we compel the encoder to refine its feature representation of the processed sequences. Essentially, both strategies render the decoder as a 'weak decoder', meaning it is not effectively optimized but instead serves to propagate gradients to the encoder for further optimization of its final state. Through the

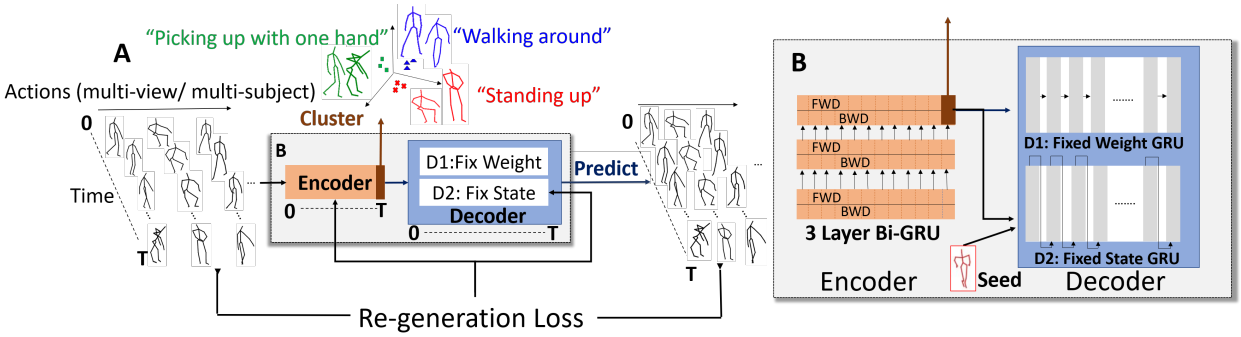

Figure 6.1: PREDICT & CLUSTER system summary. **A:** System overview. **B:** Encoder-Decoder architecture.(© 2024 IEEE)

combination of these strategies, we observe that the network can acquire a robust representation, leading to significantly enhanced performance compared to unsupervised approaches trained without them. To validate the effectiveness and generality of our methodology, we conduct t evaluations on three extensive skeleton-based and RGB+D action recognition datasets. Our results demonstrate that our P&C unsupervised system achieves high accuracy performance, surpassing previous methods in action recognition tasks.

6.2 Methods

Utilizing body-keypoints sequences captured during diverse movements as inputs, I introduce a self-supervised framework designed to correlate these sequences with corresponding actions without relying on any predefined labels. Our system is built upon an encoder-decoder recurrent neural network architecture, wherein the encoder learns to extract a discriminative feature representation within its hidden states by engaging in a prediction task. Through this self-supervised training paradigm, we demonstrate that both the decoder and the encoder autonomously organize their hidden states into a feature space. This feature space exhibits a clustering effect, wherein similar movements are grouped together within the same cluster while distinct movements are separated into distant clusters. An overview of the ***Predict***

& Cluster system is depicted in Fig 6.1.

A pivotal aspect we leverage in our system is the recent discovery that feeding input sequences through recurrent neural networks (RNNs) results in the self-organization of these sequences into clusters within the network's hidden states. These clusters effectively represent features in an embedding of the hidden states, offering a promising avenue for unsupervised multi-dimensional sequence clustering, such as body keypoints sequences [KS8]. This self-organization phenomenon is inherent to all RNN architectures, extending even to random RNNs initialized with random weights and held fixed, i.e., no training is conducted. When sequences of body keypoints corresponding to different actions are fed into random RNNs, the features within the hidden state space act as effective filters. While this approach holds promise, the achieved recognition accuracy may not be optimal. To address this, we implement an encoder-decoder system, named ***Predict & Cluster*** (P&C), wherein the encoder processes input sequences and transmits the last hidden state to the decoder. The decoder then reconstructs the input sequences. Additionally, we exploit the random network configuration (which requires no training) to determine the optimal hyper-parameters for the network to undergo training. The components of P&C are outlined in detail below.

Motion prediction. Central to our unsupervised methodology is an encoder-decoder RNN, often referred to as Seq2Seq. These network models have demonstrated effectiveness in predicting the future evolution of multi-dimensional time-series features, including temporal skeleton data of diverse actions. In typical applications, the network flow is unidirectional: The encoder handles an initial sequence of activity and conveys the last state to the decoder, which then generates the subsequent evolution based on this state. We extend this network structure for our approach, as depicted in the system overview in Fig. 6.1.

We introduce a bi-directional flow to enhance the network's ability to capture long-term dependencies in action sequences. In this configuration, the encoder comprises a multi-layered bi-directional Gated Recurrent Unit (GRU), with its input being the entire sequence of body keypoints associated with an action. We denote the hidden states of the last layer of the encoder in forward and backward directions at time t as $\overrightarrow{E_t}$ and $\overleftarrow{E_t}$ respectively and

represent the final state of the encoder as their concatenation $E_T = \{\overrightarrow{E_T}, \overleftarrow{E_T}\}$. The decoder is a uni-directional GRU with hidden states at time t denoted as D_t. The final state of the encoder is fed into the decoder as its initial state, i.e., $D_0 = E_T$. In such a setup, the decoder generates a sequence based on E_T initialization. In a typical prediction task, the generated sequence will be compared with forward evolution of the same sequence (prediction loss). In our system, since our goal is to perform action recognition, the decoder is required to re-generate the whole input sequence (re-generation loss). Specifically, for the decoder outputs $\hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T\}$ the re-generation loss function is the error between X and $\hat{X}$. In particular, we use mean square error (MSE) $L = \frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2$ or mean absolute error (MAE) $L = \frac{1}{T}\sum_{t=1}^{T}|x_t - \hat{x}_t|$ as plausible losses.

Hyper-parameter search. In any deep learning system, hyper-parameters play a crucial role in determining network performance and often require tuning to achieve an optimal configuration. Leveraging the self-organization capability of randomly initialized RNNs, we utilize the network's ability to process sequences and assess its performance before training begins. Specifically, we measure the encoder's capacity by passing the skeleton sequence through it and evaluating recognition performance based on the final encoder state. This efficient hyperparameter search approach drastically reduces total training time by identifying an optimal network configuration conducive to training.

Training. After setting the optimal hyperparameters for the encoder, training is conducted on the decoder's outputs to predict or regenerate the action sequences fed into the encoder. Training for prediction typically follows one of two approaches: (i) *unconditional* training, where zeros are fed into the decoder at each time step or (ii) *conditional* training, where an initial input is provided at the first time step, and subsequent steps use the predicted output of the preceding step as their input. Based on these training strategies, we propose two decoder configurations *(i) Fixed Weights decoder (FW)* and *(ii) Fixed States decoder (FS)*. These configurations serve to weaken the decoder, compelling it to regenerate sequences based on information from the encoder's hidden representation, thus enhancing the encoder's clustering performance.

1.Fixed Weights decoder (**FW**): In this configuration, the input to the decoder is unconditional. The decoder is not tasked with learning useful information for prediction; instead, it relies solely on the state provided by the encoder. Consequently, the weights of the decoder can be set as random, and its role becomes that of a recurrent propagator of the sequences. In training for the re-generation loss, this setup is expected to compel the encoder to learn latent features and represent them through the final state passed to the decoder. This straightforward approach proves to be computationally efficient since only the encoder is trained, and our results demonstrate promising performance, particularly when combined with KNN action classification.

2.Fixed States decoder (**FS**): In this configuration, the external input into the decoder is conditional, with each time-step's external input being the output of the previous time-step. However, the internal input, typically the hidden state from the previous step, is replaced by the final state of the encoder E_T. Specifically, within the RNN cell

$$
\begin{aligned}
&h_t = \sigma(W_x x_t + W_h g_t + b_h), g_t = h_{t-1} \rightarrow E_T, \\
&y_t = \sigma(W_y h_t + b_y), \\
&x_{t+1} = y_t,
\end{aligned}
$$

with x_t the external input, y_t the output and h_t the hidden state at time-step t, h_{t-1} terms are substituted with E_T. Additionally, a residual connection between the external input and output is introduced, a technique previously found beneficial in human motion prediction. The final output and next input will be $\hat{y}_t = y_t + x_t$ and $\hat{x}_{t+1} = \hat{y}_t$, respectively. This configuration compels the network to rely on E_T instead of the hidden state from the previous time step, mitigating the issue of gradient vanishing. Consequently, during back-propagation at each time-step, there exists a defined gradient back to the final encoder state.

Feature level auto-encoder. Once the prediction network is trained, we extract the final encoder state E_T as the feature vector corresponding to each action sequence. Given that the feature vector is high-dimensional, we employ a feature-level auto-encoder to capture the essential low-dimensional components of this high-dimensional feature, enabling its use in

classification tasks. This auto-encoder, denoted as f follows an encoder-decoder architecture with parameters θ such that

$$\hat{E_T} = f_\theta(E_T) \approx E_T.$$

The encoder and the decoder are multi-layer FC networks with non-linear tanh activation and we implement the following loss $l_{aec} = \left| E_T - \hat{E_T} \right|$.

K-nearest neighbors classifier. To evaluate our method's performance on the action recognition task, we employ the K-nearest neighbors (KNN) classifier on the middle layer of the auto-encoder feature vector. Here, we apply the KNN classifier (with $k = 1$) to the features extracted from the trained network across all sequences in the training set to assign classes. Subsequently, we utilize cosine similarity as the distance metric to carry out recognition, effectively assigning each tested sequence to a class. Importantly, the KNN classifier operates without the need for learning additional weights for action classification.

6.3 Experiments

Datasets. We utilize three distinct datasets to train, evaluate, and compare our P&C system with existing approaches. These datasets encompass varying numbers of classes, types of actions, and body keypoints captured from diverse viewpoints and subjects. Body keypoints in these datasets are captured via depth cameras and may include additional data such as RGB videos and depth information. Various types of action recognition methods have been applied to these datasets, including supervised skeleton approaches and unsupervised RGB+D methods. We provide a detailed summary of these approaches and their performance on the datasets in Table 6.1. Importantly, to the best of our knowledge, our work is the first fully unsupervised skeleton-based approach applied to these extensive action recognition tests. The datasets that we have applied our P&C system to are (i) NW-UCLA, (ii) UWA3D, and (iii) NTU RGB+D. The datasets include 3D body key points of $10, 30$, and 60 action classes, respectively. We briefly describe them below.

North-Western UCLA (NW-UCLA) dataset [Re97] is captured by Kinect v1 and contains 1494 videos of 10 actions. These actions are performed by 10 subjects repeated 1 to

6 times. There are three views of each action, and for each subject, 20 joints are being recorded. We follow [Re98] and [Re97] to use the first two views (V1, V2) for training and the last views (V3) to test cross-view action recognition.

UWA3D Multiview Activity II (UWA3D) dataset [Re99] contains 30 human actions performed 4 times by 10 subjects. 15 joints are being recorded, and each action is observed from four views: frontal, left and right sides, and top. The dataset is challenging due to many views and the resulting self-occlusions from considering only parts of them. In addition, there is a high similarity among actions, e.g., the two actions "drinking" and "phone answering" have many keypoints being near identical & idle and in the dynamic keypoints there are subtle differences.

NTU RGB+D dataset [Re100] is a large scale dataset for 3D human activity analysis. This dataset consists of 56880 video samples captured from 40 different human subjects using Microsoft Kinect v2. NTU RGB+D(60) contains 60 action classes. We use the 3D skeleton data for our experiments such that each time-sample contains 25 joints. We test our P&C method on both cross-view and cross-subject protocols.

Evaluation: In all experiments, we use the K-nearest neighbors classifier with $k = 1$ to compute the action recognition accuracy and evaluate the performance of our P&C method. We test different variants of P&C architectures and report a subset of these in the paper: *baseline* random initialized encoder with no training (**P&C-Rand**), full system with FS decoder and feature-level auto-encoder (**P&C-FS-AEC**) and full system with FW decoder and feature-level auto-encoder (**P&C-FW-AEC**).

Comparison: We compare the performance of our P&C method with prior related supervised and unsupervised methods applied to (left-to-right): NW-UCLA, UWA3D, NTU RGB+D datasets, see Table 6.1. In particular, we compare action recognition accuracy with approaches based on supervised skeleton data (blue), unsupervised RGB+D data (purple) and unsupervised skeleton data (red). For comparison with unsupervised skeleton methods, we implement and reproduce the LongTerm GAN model (LongT GAN) as introduced in [Re107] and list its performance.

Method	NW-UCLA (%)
Supervised Skeleton	
HOPC[Re99]	74.2
Actionlet Ens [Re101]	76.0
HBRNN-L[Re102]	78.5
VA-RNN-Aug[Re103]	**90.7**
AGC-LSTM[Re104]	**93.3**
Unsupervised RGB+D	
Luo et al.[Re105]	**50.7**
Li et al.[Re106]	**62.5**
Unsupervised Skeleton	
P&C Rand **(Our)**	72.0
LongT GAN [Re107]	74.3
P&C FS-AEC **(Our)**	**83.8**
P&C FW-AEC **(Our)**	**84.9**

Method	UWA3D	
	V3 (%)	V4 (%)
Supervised Skeleton		
HOJ3D[Re108]	15.3	28.2
2-layer P-LSTM[Re109]	27.6	24.3
IndRNN (6 layers)[Re109]	30.7	47.2
IndRNN (4 layers)[Re109]	34.3	54.8
ST-GCN[Re109]	36.4	26.2
Actionlet Ens[Re101]	45.0	40.4
LARP[Re110]	49.4	42.8
HOPC[Re99]	**52.7**	**51.8**
VA-RNN-Aug[Re103]	**70.9**	**73.2**
Unsupervised Skeleton		
P&C Rand **(Our)**	48.5	51.5
LongT GAN [Re107]	53.4	59.9
P&C FS-AEC **(Our)**	**59.5**	**63.1**
P&C FW-AEC **(Our)**	**59.9**	**63.1**

Method	NTU RGB-D 60	
	C-View (%)	C-Subject (%)
Supervised Skeleton		
HOPC[Re99]	52.8	50.1
HBRNN[Re102]	64.0	59.1
2L P-LSTM[Re100]	70.3	62.9
ST-LSTM[Re111]	**77.7**	**69.2**
VA-RNN-Aug[Re103]	**87.6**	**79.4**
Unsupervised RGB+D		
Shuffle & learn[Re112]	40.9	46.2
Luo et al.[Re105]	**53.2**	**61.4**
Li et al.[Re106]	**63.9**	**68.1**
Unsupervised Skeleton		
LongT GAN [Re107]	48.1	39.1
P&C Rand **(Our)**	56.4	39.6
P&C FS-AEC **(Our)**	**76.3**	**50.6**
P&C FW-AEC **(Our)**	**76.1**	**50.7**

Table 6.1: Comparison of action recognition performance of our P&C system with state-of-the-art approaches of *Supervised Skeleton* (blue) and *Unsupervised RGB+D* (purple); *Unsupervised Skeleton* (red) types.(© 2024 IEEE)

For NW-UCLA, P&C outperforms previous unsupervised methods (both RGB+D and skeleton-based). Our method even outperforms the first three supervised methods listed in Table 6.1-left. UWA3D is considered a challenging test for many deep learning approaches since the number of sequences is small, while it includes a large number of classes (30). Indeed, action recognition performance of many supervised skeleton approaches is low ($< 50\%$). For such datasets, it appears that the unsupervised approach could be more favorable, i.e., even P&C Rand reaches a performance of $\approx 50\%$. LongT GAN achieves slightly higher performance than P&C Rand, however, not as high as P&C FS/FW-AEC which perform with $\approx 60\%$. Only a single supervised skeleton method, VA-RNN-Aug, is able to perform better than our unsupervised approach, see Table 6.1-middle. On the large-scale NTU-RGB+D

dataset, our method performs extremely well on the cross-view test. It outperforms prior unsupervised methods (both RGB+D and skeleton-based) and is on-par with ST-LSTM (second best supervised skeleton method), see Table 6.1-right. On the cross-subject test we obtain performance that is higher (including P&C Rand) than the prior unsupervised skeleton approach, however, our accuracy does not outperform unsupervised RGB+D approaches. We believe that the reason stems from skeleton-based approaches not performing well in general on cross-subject tests since additional aspects such as subject parameters, e.g., skeleton geometry and invariant normalization from subject to subject, need to be taken into account.

In summary, for all three datasets, we used a single architecture, and it was able to outperform the prior unsupervised skeleton method, LongT-GAN [Re107], most supervised skeleton methods and unsupervised RGB+D methods on cross-view tests and some supervised skeleton and unsupervised RGB+D on large scale cross subject test.

Chapter 7

INSTRUMENTAL MUSIC FROM MUSICIAN BODY MOVEMENTS

7.1 Motivation

A multi-instrumentalist is a musician who plays two or more musical instruments and easily transitions from one instrument to another. For example, the famous multi-instrumentalist, Prince, played all of the 27 instruments featured in the album 'For You'. Such talent is expressed in the ability to disentangle the uniqueness of each instrument along with maneuvering the similarities across instruments.

Associations of music with an instrument appear to be tangled, evidenced by perceptual experiments suggest that non-professionals would not be able to tell which instrument is playing by just listening to a piece of instrumental music. Such a complexity exists in a computational setting as well, where disentanglement of music characteristics, including instrument type, pitch, rhythm, dynamics, and timbre from audio, is not straightforward. In a situation where the audio signal is ambiguous, the visual information, i.e., video of the musician playing, greatly simplifies such associations where visual cues along with audio complete each other.

The approaches in Chapter 3, 4 and 5 leverage various pre-selected intermediate representations to serve as a bridge to connect vision and music, introducing multi-stage generative modeling. In this Chapter, I describe our study of generating instrument music from videos using a single generative model. Through the training process, I also aim to learn deep learning features that can interpret the timbre attribute of instruments. Since an large number of instruments could be available worldwide, I propose to generate multi-instrumental music from videos without labeling the instruments. To achieve that, I introduce a novel approach

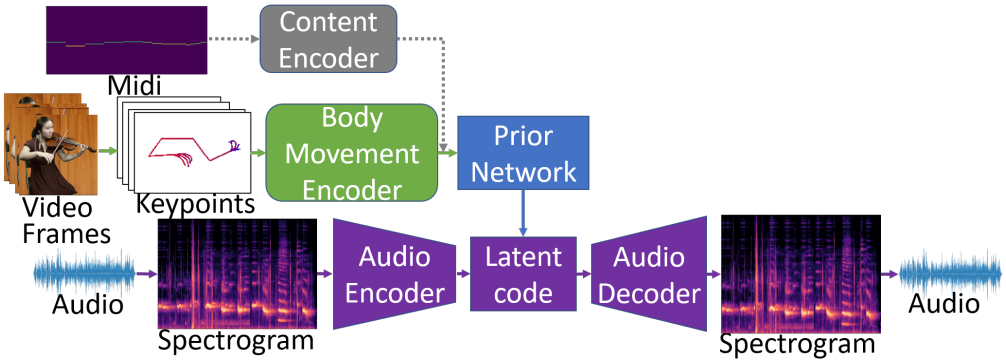

Figure 7.1: ***MI Net*** system overview. VQ-VAE (bottom) is used to reconstruct audio sequences of various instruments and infers a latent representation of music (latent code). VQ-VAE is conditioned by a prior network (middle) that encodes body movements w/wo MIDI content. Further, given an input of Video Frames of the musician playing the instrument, Multi-instrumentalist Net (***MI Net***) generates the music for that instrument. (Figure from [KS3])

named 'Multi-Instrumentalist Net' (***MI Net***) that first learns a discrete latent representation of the music of various instruments and leverages human movement of the instrument playing as the condition to drive the music generation. The pipeline is a novel adaptation of a Vector Quantized Variational Autoencoder (VQ-VAE) [Re7] that can encode and generate log-spectrogram audio representations. We train the pipeline with an autoregressive prior conditioned on the musician's body keypoints movements encoded by a recurrent neural network. Joint training of the prior with the body movements encoder disentangles the music into latent features indicating the musical components and the instrumental features. The latent space generates distributions disentangled into clusters of distinct instruments from which new music can be generated. Furthermore, the VQ-VAE architecture supports detailed music generation with additional content conditioning via Midi to generate the exact content of the music played by each instrument in the video. It thus supports novel mixing applications on the instrument level.

7.2 Methods

Encoding of Audio Representations. As discussed in 2.1, we train a neural audio codec architecture to first encode the magnitude of the log-spectrogram into the latent space and obtain the audio representations for conditional audio modeling. We introduce a multi-band residual 1D convolutional Vector Quantized Variational Autoencoder (VQ-VAE with MBR), as shown in Fig. 7.1. VQ-VAE [Re7] encoder outputs a discrete latent representation, and the decoder decodes this representation and reconstructs the input. Direct application of original VQ-VAE to spectrogram is not optimal since the log-spectrogram is of high dimension due to the high resolution of the frequency bins. Therefore, we introduce novel components capable of processing spectrograms within VQ-VAE. In particular, we propose to include a multi-band residual learning method on the audio encoder and the decoder to capture the spectral features of musical overtones better. Such a block was shown effective in Midi to music synthesis [Re59]. The multi-band residual (MBR) block splits the input into a specific number of frequency bands and then feeds each band individually to identical sub-blocks consisting of the following layers: 1D-convolution, ReLU, and 1D-convolution. The output of all sub-blocks is then concatenated along the frequency dimension, and a residual connection sums up the output with the input of the block. In the audio encoder, we progressively divide the spectrogram into more bands in the earlier layers and into fewer bands in the latter layers. The decoder then decodes the latent representation from fewer bands to more bands in a symmetric way. Since both the encoder and the decoder are fully 1D convolutional, the system supports any input length. Our VQ-VAE model incorporates two additional terms in its objective to align the vector space of the code with the output of the encoder. (i) The embedding loss is applied to the codebook variable and e_k and brings the selected codebook $\mathbf{e}$ closer to the output of the encoder $E(S)$. (ii) The commitment loss is applied to the encoder weights, which aims to keep the encoder output as close as possible to the chosen codebook vector to prevent it from fluctuating from one code vector to another. As proposed in [Re7], we use the exponential moving average updates for the codebook to

replace the embedding loss. The resulting loss is $\mathcal{L} = \|S - D(e)\|_2^2 + \beta\|E(S) - sg[e]\|_2^2$, where the first term is the reconstruction loss and the second term, β, is a hyper-parameter that depends on the scale of the reconstruction loss and sg stands for the stop gradient operator defined as the identity at forward computation time and has zero partial derivatives.

Encoding Visual Representations and Learning a Prior over the Audio Latent Code. Given a sequence of 2D human pose key-points $P = \{p_1, p_2, ..., p_T\} \in \mathbb{R}^{J \times T}$, where J is the number of joints and T is the total number of time steps, we first encode the key-points into a latent representation. If optimal, the latent space should differentiate movements of playing different instruments and self-organize itself into separate clusters as shown in my previous unsupervised skeleton-based action recognition approach (Section 6) [KS8, KS6]. To achieve this, we use a bi-GRU as the body movement encoder E_b. Given the input sequence P, the last hidden state of the encoder $h = E_b(P)$ is taken as the global representation of the musician's movement. To associate body movements with audio features, we jointly train the body key points encoder with the prior latent space of the audio features. Superimposed with the latent audio features, the body movement features differentiate the type of instrument performance and other characteristics of the performed music. This allows us to use body movements to generate new music for the corresponding instrument.

The prior distribution over the discrete latent audio features $p(Z)$ is a joint distribution of categorical variables across time and can be learned autoregressively. When training our system, the prior is kept constant and uniform. After training, we fit an autoregressive distribution over Z to generate new samples via ancestral sampling. We use the encoder of transformer structure [Re113] over the discrete latent space. We concatenate the last hidden state of the body keypoints encoder h to every time step of the discrete latent representation. This forces the prior of audio features to align and correlate to body motion features when autoregressively learning to predict the next latent code. Subsequently, the concatenated features are passed through multi-head self-attention and feed-forward layers. The outputs are passed to a softmax layer to predict the probability of the next latent code over the codebook.

Content Conditioning. In addition to the unconditional generation of music, we could generate the exact music content for each instrument in the video by using the Midi matrix $M \in N \times T$ as the content signal. Since instruments considered in our experiments are monophonic, there is a single note at each time step. We first convert the 2D matrix into a binary matrix by ignoring the expressive dynamics (i.e., the loudness of music). We then transform the binary matrix to a 1D sequence containing the activated note's index at each time step. We use a transformer-based encoder-decoder architecture to achieve content conditioning [Re19]. The content encoder is the transformer encoder that inputs the Midi information. We then concatenate the encoded body movement representation to the embedded Midi at each time step and pass them to two self-attention and feed-forward layers. The content encoder output will go through a fully connected layer to become the conditioned signal C. The transformer decoder includes a cross-attention layer after the self-attention layer to compute the attention between the conditioned signal and the latent code representation. Considering the output of the self-attention module $A \in \mathbb{R}^{T_a \times C}$ and the Midi conditional signal $M \in \mathbb{R}^{T_m \times C}$, where C is the feature dimensions, the cross-attention is defined as: Cross Attention(A, M) = softmax$(\frac{AM^T}{\sqrt{D_k}})M$. Here, the feature dimension of A and M are designed to be the same. During sampling, we provide both Midi and body movements of each instrument's performance to the content and body movement encoder, respectively. The prior network autoregressively generates discrete latent representations via ancestral sampling.

7.3 Experiments

Dataset. We evaluate the ***MI Net*** on **URMP** dataset [Re114], a high-quality multi-instrument video dataset recorded in a studio. It includes 13 instruments and provides the musical score in Midi format, which we use for evaluation. We further evaluate the MI Net on **Solos** dataset [Re115], a recent dataset of YouTube videos of musicians' recitals. It contains the same 13 instruments as the URMP dataset. Solos also contain pre-processed skeleton key points extracted via Openpose; however, it doesn't include Midi files due to the 'in the

wild' nature of the videos.

Comparison with Other Models. For comparison, we implement two baselines of current systems: (i) *RNN-based Seq2Seq Network:* An encoder-decoder recurrent neural network. The encoder inputs body key points, and the decoder generates the expected spectrogram. (ii) *Graph-Transformer Network:* A Graph-Transformer network similar to the architecture of Foley Music [Re39]. It is based on Spatio-temporal Graph Convolution Network (ST-GCN) [Re63], which encodes the body key points to pose features, then fed into the Transformer decoder where each block contains self-attention, cross-attention, and feed-forward modules. Instead of predicting the Midi events, we directly generate the log-spectrograms.

Number of Statistically-Different Bins (NDB). We adopt the metric proposed in [Re75] and used in [Re22, Re39] to measure the diversity of the generated examples. NDB is reported as the number of cells where the number of training examples is statistically significantly different from the number of generated examples by a two-sample Binomial test. For each model, we generate 1600 samples from the testing set and perform the comparison. The NDB results are shown in Table 7.1 and indicate how well the model learns from the training set. The largest non-optimal NDB score is 50. The lower NDB score the better it is. For the URMP dataset, the MI Net outperforms other methods by a large margin. Both the RNN-based Seq2Seq and Graph-Transformer do not generate music with sufficient quality for the unsupervised setup. Thereby, the generated samples of MI Net without conditioning on content have a distribution closer to the training data, resulting in a lower NDB score than the reference itself. For the Solos dataset, the NDB result of our method is better than others, however, not as realistic as for URMP. One of the limitations is that the body motions of 'in the wild' videos have large variations.

Classification with Body Motion. To evaluate whether the encoded pose features are separated to generate exclusive music for specific instruments, we extract the body movement encoder's final hidden state and fit the K-Nearest-Neighbors classifier ($K = 1$) using the cosine similarity metric. For comparison, we use the encoder final state for RNN-based Seq2Seq and take the mean of both temporal and joint dimensions of the outputs of ST-

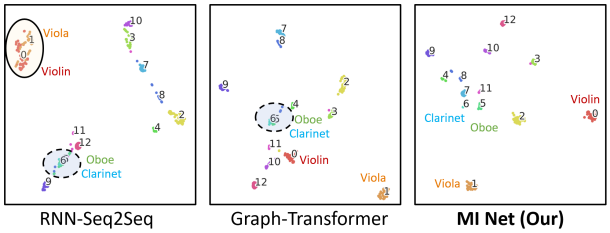

Figure 7.2: T-SNE plots of the encoded body movements representation in URMP test set. the number next to clusters indicates instruments. Dotted circle: Mixing of samples belonging to Oboe and Clarinet instruments. Solid circle: Mixing of samples belonging to Violin and Viola instruments. (Figure from [KS3])

Model	URMP	Solos
RNN-based Seq2Seq	48	44
Graph-Transformer	45	41
MI Net (Our)	**33**	**36**
Content Cond. MI Net (Our)	**35**	-
Testing Set Data	36	31

Table 7.1: NDB results. **Lower is better**. (Table from [KS3])

Model	URMP	Solos
RNN-based Seq2Seq	82.6	37.8
Graph-Transformer	89.2	38.3
MI Net (Our)	**98.7**	**61.5**

Table 7.2: KNN Classification Accuracy in %. (Table from [KS3])

GCN for Graph-Transformer to perform the classification. The results are shown in Table 7.2. Notably, for the URMP dataset, our method associates the body movements with the instruments at a high accuracy, evidenced by the score in Table 7.2 and t-SNE plots of the latent representations in Fig.7.2. These plots show that the models that we compare against do not distinguish well the instruments. Furthermore, these models appear to separate the instruments according to the difference in pose only and thus cannot generate music across instruments. This is particularly challenging for instruments with similar poses (e.g., Viola v.s. Violin, Oboe v.s. Clarinet). In comparison, our method obtains a latent space that self-organizes visual and audio events by instruments.

Chapter 8

OBJECT IMPACT SOUND FROM VIDEOS

8.1 Motivation

In this chapter, I further describe the generation of impact sounds from videos showcasing object interactions. This task, akin to synthesizing instrumental sounds, aims to achieve generation through a single generative model. Visual cues often enable the natural inference of object materials, prompting us to anticipate specific sound types. As outlined earlier, traditional film production heavily depends on skilled Foley artists who create a plethora of sound samples beforehand and manually edit them to align with visual content. While this method delivers a satisfying auditory experience in cinemas, it is labor-intensive and difficult to scale for generating sound effects for diverse and complex physical interactions.

Recent advancements in automatic impact sound synthesis have been categorized into two primary approaches. The first, physics-based modal synthesis [Re32], simulates sounds resulting from various object interactions. These synthesized sounds can capture the nuances of different interactions and the objects' geometric properties. However, these methods necessitate a complex setup for physics simulation and a time-consuming process to determine physics parameters for sound synthesis, making them impractical for complex scenes. Conversely, the proliferation of impact sound videos has made training deep learning models for sound synthesis a viable alternative [Re35, Re37]. Despite the promise shown by these models in audio-visual applications, many rely on end-to-end training of black-box models without incorporating crucial physics knowledge. Such omission is critical in impact sound synthesis, where minor changes in impact location can significantly alter the sound produced, leading to models that often generate averaged or smoothed audio representations filled with artifacts, resulting in unauthentic sounds.

The primary challenge for learning-based methods is the subtle correspondence between the visual and audio domains, as impact sounds are highly sensitive to underlying physics. Generating high-fidelity impact sounds from videos alone is challenging without incorporating physics insights. Inspired by physics-based methods that use physics mode parameters for sound representation and synthesis, I introduce a physics prior encompassing adequate physics details to guide deep generative models in synthesizing impact sounds from videos. Given the impracticality of conducting physics simulations on raw video data to obtain precise parameters, we explored estimating and predicting physics priors from video sounds. These priors significantly enhance the quality of the synthesized impact sounds. Our system comprises two stages: the first involves encoding sound physics knowledge through physics priors, utilizing signal processing for physical parameter estimation and neural networks for learning residual parameters reflecting the sound environment (shown in Fig. 8.1). The second stage involves a DDPM model, conditioned on visual inputs and physics priors, to generate the spectrogram of impact sounds (shown in Fig. 8.2). Since physics priors derived from audio samples are unavailable during inference, we propose a novel inference pipeline. This pipeline uses test video features to retrieve a physics latent feature from the training set, serving as a guide for synthesizing impact sounds on unseen videos, thereby allowing for the generation of novel impact sounds through the diffusion model, even when reapplying physics knowledge from the training dataset.

8.2 Methods

Reconstruct Physics Priors From Sound. Our objective is to reconstruct physical properties from sound. This process is divided into two key modules: 1) First, we have the physics parameters estimation module, which is responsible for extracting modal parameters from the audio waveform. This involves identifying the specific characteristics of the sound that correlate with physical interactions, such as vibrations and collisions. 2) Second, the residual parameters prediction module focuses on encoding environmental information, including background noise and reverberation, through the use of neural networks. This module aims

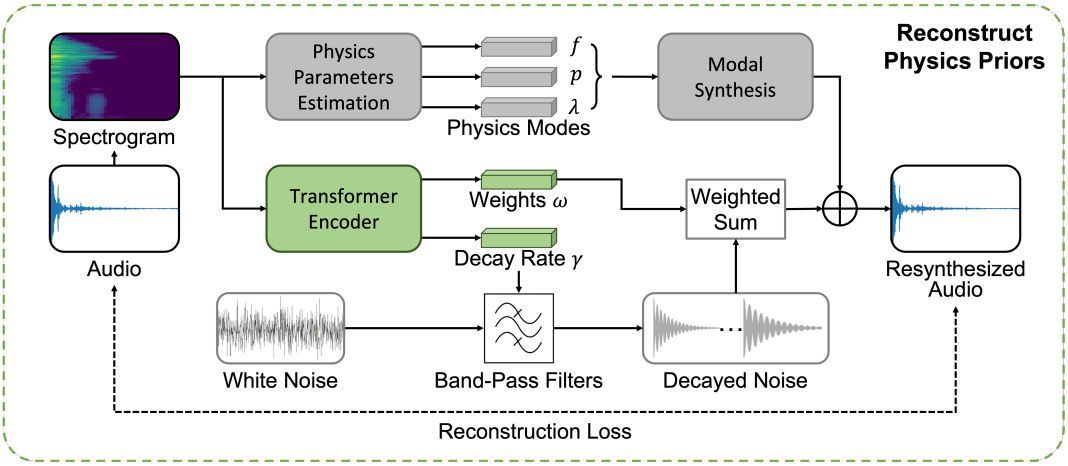

Figure 8.1: Reconstruction of physics priors by two components: 1) We estimate a set of physics parameters (frequency, power, and decay rate) via signal processing techniques. 2) We predict residual parameters representing the environment by a transformer encoder. A reconstruction loss is applied to optimize all trainable modules.(© 2024 IEEE)

to capture the nuances of the sound's context, which are not directly related to the physical properties of the objects involved but significantly affect the overall auditory experience. These modules work together to provide a comprehensive analysis of sound, bridging the gap between auditory signals and their physical origins.

Physics Parameters Estimation. Traditional linear modal synthesis is a cornerstone technique for physics-based sound synthesis [Re33], where the displacement x of a system is derived from a linear equation described as follows:

$$M\ddot{x} + C\dot{x} + Kx = F, \tag{8.1}$$

where F represents the force, M represents the mass, C represents the damping, and K represents the stiffness.

This approach allows for the resolution of the generalized eigenvalue problem $KU =$

ΛMU, subsequently decomposing it into a more manageable form:

$$\ddot{q} + (\alpha I + \beta\Lambda)\dot{q} + \Lambda q = U^T F \tag{8.2}$$

where Λ represents the diagonal matrix that contains eigenvalues of the system, U represents the eigenvectors which can transform x into the bases of decoupled deformation q by matrix multiplication $x = Uq$.

Once the system is decoupled, a set of modes is obtained, each of which can be succinctly described as damped sinusoidal waves. The i-th mode can be expressed by:

$$q_i = p_i e^{-\lambda_i t} \sin(2\pi f_i t + \theta_i) \tag{8.3}$$

where f_i is the frequency of the mode, λ_i is the decaying rate, p_i is the excited power, and θ_i is the initial phase. It is also common to represent q_i under the decibel scale and we have

$$q_i = 10^{(p_i - \lambda_i t)/20} \sin(2\pi f_i t + \theta_i). \tag{8.4}$$

The frequency, power, and decay rate can define the physics parameter feature ϕ of mode i: $\phi = (f_i, p_i, \lambda_i)$ and we neglect θ_i since we assume the object is initially at rest and struck at $t = 0$ and therefore it is usually treated as zero in the estimation process [Re33].

To analyze a given recorded audio waveform $s \in \mathbb{R}^T$, we commence by estimating physics parameters, including a collection of damped sinusoids characterized by constant frequencies, powers, and decay rates. The process begins with the computation of the log-spectrogram magnitude $S \in \mathbb{R}^{D \times N}$ of the audio using the short-time Fourier transform (STFT), where D is the number of frequency bins and N is the number of frames. To ensure we capture an adequate number of physics parameters, we align the number of modes with the frequency bins' count. For each frequency bin, we pinpoint the peak frequency f utilizing the magnitude result from the fast Fourier transform (FFT) over the entire audio segment. The magnitude at the spectrogram's first frame is then designated as the initial power p. Subsequently, we calculate the decay time λ for each mode based on the temporal bin at which it descends to silence (-80dB). Upon completing these steps, we ascertain the physics

parameters $\{(f_i, p_i, \lambda_i)\}_{i=1}^{D}$ for D modes, enabling us to re-synthesize the audio waveform using equation 8.4. This methodological approach not only facilitates a detailed analysis of the sound's physical properties but also allows for the precise recreation of audio waveforms based on the underlying physics parameters, offering a robust tool for sound synthesis and manipulation.

Residual Parameters Prediction. While the estimated modes proficiently capture the essence of impact sounds resulting from physical interactions between objects, audio recorded in natural settings encompasses complex residual components. These include background noise and reverberation, which are inherently dependent on the acoustic environment and play a pivotal role in creating a realistic and immersive auditory experience. To address this, we introduce a learning-based strategy for modeling such residual parameters, aiming to approximate the sound environment component using exponentially decaying filtered noise.

Our method begins with the generation of a Gaussian white noise $\mathcal{N}(0, 1)$ signal, which is subsequently partitioned into M frequency bands using a band-pass filter (BPF). For each band m, the residual component is defined by the following:

$$R_m = 10^{(-\gamma t)/20}\mathrm{BPF}(\mathcal{N}(0, 1))_m \tag{8.5}$$

The accumulated residual components R is a weighted sum of subband residual components

$$R = \sum_{m=1}^{M} w_m R_m, \tag{8.6}$$

where w_m is the weight coefficient of band m residual component.

We then process the log-spectrogram $S \in \mathbb{R}^{D\times N}$ of the input audio through a transformer-based encoder, which encodes each frame of the S. The resulting features are averaged, and through the application of two linear projections, we estimate the decay time $\gamma \in \mathbb{R}^M$ and weights $w \in \mathbb{R}^M$ for each band. To refine our model and reduce discrepancies, we employ a multi-resolution STFT loss $L_{\text{mr-stft}}(\hat{s} + R, s)$ to minimize the error between $\hat{s} + R$ and s. This loss function has demonstrated efficacy in accurately modeling audio signals within the time

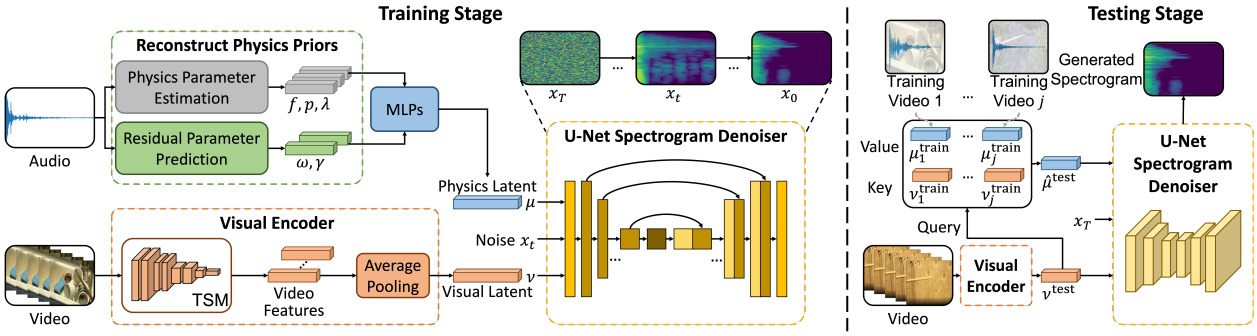

Figure 8.2: Overview of the physics-driven diffusion model for impact sound synthesis from videos. (left) During training, we reconstruct physics priors from audio samples and encode them into a physics latent. Besides, we use a visual encoder to extract visual latent from the video input. We apply these two latents as conditional inputs to the U-Net spectrogram denoiser. (right) During testing, we extract the visual latent from the test video and use it to query a physics latent from the key-value pairs of visual and physics latents in the training set. Finally, the physics and visual latents are used as conditional inputs to the denoiser and the denoiser iteratively generates the spectrogram.(© 2024 IEEE)

domain [Re116], ensuring that our approach can effectively capture the nuanced dynamics of the sound environment.

By simultaneously estimating physics parameters and predicting residual parameters, we derive a comprehensive set of physics priors. These priors serve as an essential condition for guiding our impact sound synthesis model, enabling it to produce high-fidelity sounds from video inputs. This holistic approach not only enhances the realism of synthesized audio but also enriches the perceptual experience, bridging the gap between visual cues and corresponding auditory outputs.

Physics-Driven Diffusion Models. Leveraging physics priors and video inputs, we introduce a conditional Denoising Diffusion Probabilistic Model (DDPM) tailored for the synthesis of impact sounds. This innovative model employs a reverse diffusion process, meticulously

steering the noise distribution towards a spectrogram distribution that resonates with the provided physics priors and visual content. To achieve this, we intricately encode both the physics and residual parameters into a cohesive latent feature embedding using multi-layer perceptrons (MLPs). This process culminates in the creation of a physics latent vector, represented by μ. For the video inputs, we process a series of RGB frames through a Temporal-Shift Module (TSM) [Re117], an efficient mechanism for extracting dynamic visual features. These features are subsequently aggregated using average pooling to formulate a singular visual latent representation, denoted by ν. This dual-input strategy allows our DDPM to generate highly accurate and contextually relevant impact sounds by integrating detailed physical properties with visual dynamics, ensuring a synchronized and immersive auditory experience that closely mirrors the real-world phenomena depicted in the videos.

Our physics-driven diffusion model for sound synthesis, illustrated in Fig. 8.2, centers around a diffusion forward process. This process incrementally introduces Gaussian noise $\mathcal{N}(0, I)$ across time steps $t = 0, ..., T$ into a spectrogram x, with each step having a specified variance scale β. We employ a scheduler to adjust the variance scale at each time step, resulting in a sequence of scales $\beta_1, \beta_2, ..., \beta_T$ [Re118]. The spectrogram at diffusion time step t is denoted as x_t. Given the spectrogram at the previous time step x_{t-1}, along with the physics latent μ, and the visual latent ν, the explicit diffusion process for the spectrogram at time step t is expressed as $q(x_t|x_{t-1}, \mu, \nu)$. Since the complete diffusion process that transitions x_0 to x_T, conditioned on the physics latent vector μ and visual latent vector ν, operates as a Markov process, it can be decomposed into a series of multiplicative steps $\prod_{t=1}^{T} q(x_t|x_{t-1})$.

For synthesizing a spectrogram, the reverse process is crucial, aiming to reconstruct a spectrogram from Gaussian noise. This reverse process is defined by the conditional distribution $p_\theta(x_{0:T-1}|x_T, \mu, \nu)$, and leveraging the Markov chain property, it can be broken down into multiple transitional stages:

$$p_\theta(x_0, ..., x_{T-1}|x_T, \mu, \nu) = \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, \mu, \nu). \tag{8.7}$$

Given the diffusion time-step with physics latent and visual latent conditions, a spectrogram is recovered from the latent variables by applying the reverse transitions $p_\theta(x_{t-1}|x_t, \mu, \nu)$. Considering the spectrogram distribution as $q(x_0|\mu, \nu)$, we aim to maximize the log-likelihood of the spectrogram by learning a model distribution $p_\theta(x_0|\mu, \nu)$ obtained from the reverse process to approximate $q(x_0|\mu, \nu)$. Since it is common that $p_\theta(x_0|\mu, \nu)$ is computationally intractable, we follow the parameterization trick in [Re27, Re118] to calculate the variational lower bound of the log-likelihood. Specifically, the training objective of the diffusion model is L1 loss function between the noise $\epsilon \sim \mathcal{N}(0, I)$ and the diffusion model output f_θ described as follows:

$$\min_\theta ||\epsilon - f_\theta(h(x_0, \epsilon), t, \mu, \nu)||_1, \tag{8.8}$$

where $h(x_0, \epsilon) = \sqrt{\hat{\beta}_t} x_0 + \sqrt{1 - \hat{\beta}_t}\epsilon$, and $\hat{\beta}_t = \prod_{\bar{t}=1}^{t} 1 - \beta_{\bar{t}}$. Through this meticulously designed process, our model achieves the synthesis of spectrograms, effectively reverting the noise-added spectrogram back to its original state, guided by the integration of both physics-driven and visual context cues, ensuring a highly accurate and context-aware sound synthesis.

Training and Inference. During the training phase, we utilize physics priors extracted from the audio waveform as an ancillary condition to assist the model in learning the correlation between video inputs and impact sounds. However, in the inference phase, the absence of ground truth sound clips means we cannot directly extract physics priors for new video inputs as done during training. To navigate this challenge, we introduce a novel inference pipeline that retains the advantages of physics priors without requiring direct extraction from the audio.

This method involves creating key-value pairs for visual and physics latents within our training dataset. In the inference stage, when a new video input is presented, we extract its visual latent vector, denoted as ν^{test}. This vector acts as a query feature, which we use to search for the closest match in our training dataset by calculating the Euclidean distance between ν^{test} and each training video latent $\{\nu_j^{\text{train}}\}_{j=1}^{J}$. The closest training video latent

serves as the key to retrieve its corresponding physics latent μ_j^{train} which we then use as the test physics latent $\hat{\mu}^{\text{test}}$.

Armed with both the visual latent ν^{test} and the inferred physics latent $\hat{\mu}^{\text{test}}$, our model embarks on reversing the noise-infused spectrogram. It does so by first predicting the added noise at each forward iteration to produce the model output $f_\theta(x_t, t, \hat{\mu}^{\text{test}}, \nu^{\text{test}})$ and subsequently removing this noise. Through iterative sampling across all time steps, we arrive at the final spectrogram distribution $p_\theta(x_0|\hat{\mu}^{\text{test}}, \nu^{\text{test}})$.

It's important to note that although we employ physics latents from the training set, our model is capable of generating novel sounds. This innovation is made possible because the diffusion model also incorporates additional visual features from the new video inputs, ensuring that the synthesized sounds are both unique and contextually aligned with the visual content.

8.3 Experiments

Dataset. To assess the performance of our physics-driven diffusion models and compare them with other methodologies, we employ the Greatest Hits dataset [Re34]. This dataset features recordings of individuals interacting with various physical objects, primarily through actions such as hitting and scratching materials using a drumstick. Each action within the dataset has been meticulously annotated by human annotators, who have provided material labels for the objects involved and the precise timestamps of the impact sounds. This level of detailed annotation allows for a comprehensive evaluation of our models' ability to accurately synthesize and replicate the nuanced sounds associated with different materials and interactions, facilitating a direct comparison with existing approaches in the field.

Evaluation Metrics. To rigorously assess the fidelity and relevance of the generated samples from our physics-driven diffusion models, we employ four distinct metrics designed for automatic evaluation. This evaluation framework includes training an impact sound object material classifier, leveraging the labels from the Greatest Hits Dataset. We utilize a ResNet-50, a convolutional neural network architecture, and train it using spectrograms as input.

The chosen metrics are as follows: 1) Fréchet Inception Distance (FID): This metric evaluates the quality of generated impact sound spectrograms by measuring the distance between the distribution of synthesized spectrograms and those within the test set. To construct these distributions, we extract features from the layers preceding the impact sound classification layer in our model. 2) Kernel Inception Distance (KID): KID is computed using maximum mean discrepancy (MMD) to evaluate the similarity between the synthesized and real impact sound features. The MMD is calculated across numerous subsets, allowing us to obtain both the mean and the standard deviation of the KID, providing a nuanced understanding of the distributional similarity. 3) KL Divergence (KLD): Unlike FID and KID, which assess the distribution of collections of samples, KLD measures the distance between the output distributions of synthesized and actual ground truth features on an individual basis. This metric offers a more granular view of the model's performance in replicating specific sound characteristics. 4) Recognition Accuracy: This metric tests the quality of the generated impact sound samples by evaluating their ability to 'fool' the trained classifier. A high recognition accuracy indicates that the synthesized sounds are indistinguishable from real impact sounds to the classifier, showcasing the model's effectiveness in generating realistic audio.

Together, these metrics provide a comprehensive and multi-faceted evaluation of our model's performance, comparing it against other approaches and assessing its capability to produce high-quality, relevant audio samples. **Results.** The results of our quantitative evaluation are presented in the accompanying table 8.1, highlighting that our proposed physics-driven diffusion model surpasses all competing methods across every evaluated metric. A key observation from these results is the inadequacy of using solely video features as a condition for the spectrogram denoiser in generating high-fidelity sounds. Although the inclusion of class labels as additional information does result in some improvement, there remains a significant disparity in performance when compared to our physics-driven approach. This disparity underscores the critical role that physics priors play in enhancing the quality and realism of synthesized sounds. The physics-driven model leverages a deeper understanding of the physical interactions at play, enabling it to produce audio that more accurately

Model\Metric	FID ↓	KID (mean, std)↓	KL Div. ↓	Recog. Acc (%) ↑
ConvNet-based	43.50	0.053, 0.013	4.65	51.69
Transformer-based	34.35	0.036, 0.015	3.13	62.86
Video Diffusion	54.57	0.054, 0.014	2.77	69.94
Video + Class label Diffusion	31.82	0.026, 0.021	2.38	72.02
Video + MFCC Diffusion	40.21	0.037, 0.010	2.84	67.87
Video + Spec Diffusion	28.77	0.016, 0.009	2.55	70.46
Video + Physics Diffusion (Ours)	**26.20**	**0.010, 0.008**	**2.04**	**74.09**

Table 8.1: Quantitative evaluations for different models. For FID, KID, and KL Divergence, lower is better. For recognition accuracy, higher is better. The bold font indicates the best value.(© 2024 IEEE)

reflects the nuances of real-world impact sounds. This advantage is not merely incremental but substantial, illustrating the value of incorporating physics-based insights into the sound synthesis process for achieving superior results.

Chapter 9

SPATIAL AUDIO REPRESENTATION FOR INDOOR SCENES

9.1 Motivation

The focus of previous chapters centered on audio content generation but it did not take into account the environment's profound influence on the sound. This is an important aspect since the world is home to over a billion buildings, each with its own unique architecture, interior design, and intended purpose. While vision is paramount for forming an overall impression and navigation through interiors, auditory perception is crucial for a fully immersive experience. Daily interactions within an environment, such as conversation, playing music, watching TV, or locating pets by sound, rely heavily on our auditory capabilities. The quality of sound and its harmonization with the scene is essential for completing the audio-visual perception, underlining the significant role scene acoustics play in determining the suitability of space for specific activities. For instance, an IMAX theater equipped with the latest surround sound system attracts moviegoers, on the other hand quiet classrooms are reserved for educational purposes, and coffee shops create a vibrant yet ambient atmosphere for working.

Given the importance of spatial audio in interior scenes, computational modeling of spatial audio aspects is crucial for rendering scenes with spatial sound. This task, however, is complex and has been a focal point of acoustics research for decades [Re119]. Typically, the spatial sound emitted by a source and its interaction within a space can be characterized by an impulse response, contingent upon the positions of the sound source and the listener [Re120]. In real-world settings, impulse responses are measured by emitting a sine sweep from a loudspeaker and recording it at the listener's location with a microphone [Re121]. Alternatively, computational geometry-based sound propagation techniques can simulate impulse responses

for both real and virtual scenes [Re122, Re123, Re124, Re125]. Nonetheless, rendering impulse responses throughout a continuous space is both time-consuming and computationally intensive, limiting immersive and interactive spatial sound rendering.

Traditional encoding methods for impulse responses rely on a limited set of perceptual parameters to reproduce reverberations [Re126, Re127], often resulting in scene-specific custom solutions that may not translate well to novel environments. Conversely, recent advances in neural networks, particularly in implicit, continuous representations, have shown potential in computer graphics and could offer a new approach to acoustic field representation. Given that sound propagation is governed by the acoustic wave equation, with solutions representing a continuous field of impulse responses, encoding the acoustic field in a smooth, continuous representation could overcome the limitations of discrete encoding and interpolation during rendering.

Inspired by interactive sound propagation techniques that use a precomputed acoustic transfer operator independent of specific emitter and listener positions [Re128, Re129], we introduce Implicit Neural Representations for Audio Scenes (***INRAS***). This lightweight, efficient neural network model can generate high-fidelity spatial impulse responses at any emitter-listener position. ***INRAS*** comprises two primary stages: the first stage divides the audio scene's features into three parallel modules—Scatter, Bounce, and Gather—each generating independent features for the emitter, scene geometry, and listener. This disentanglement allows for the versatile representation of various scenes with minimal parameter adjustments. In the second stage, the listener module integrates these features to produce directional and binaural impulse responses. An overview of ***INRAS*** is provided in Figure 9.2, illustrating its comprehensive approach to modeling spatial audio for enhanced immersion in audio-visual experiences.

9.2 Methods

Problem Setup. ***INRAS*** utilizes advanced deep neural networks to articulate a continuous implicit function that correlates the coordinates within a 3D scene to their respective time-

domain directional and binaural impulse responses in the sound field. To formalize, for any given 3D scene D, sound emitter locations are denoted as $s \in \mathbb{R}^3$, listener locations as $l \in \mathbb{R}^3$, and the listener head orientation as $\theta \in \mathbb{R}^2$. Thus, for every combination $\forall(s, l, \theta) \in \mathbb{R}^8$ within the scene, there exists a corresponding set of binaural impulse responses $h \in \mathbb{R}^{2\times T}$ where T represents the temporal length. We conceptualize a continuous function $f(s, l, \theta) \rightarrow h$ parameterized by a deep neural network, to map each set of s, l, θ to the appropriate impulse response h.

A pivotal insight for facilitating the training process is recognizing that while the scene's geometry is a determinant factor for the impulse responses, it remains unchanged regardless of variations in the positions of the emitter and listener. This constant, geometry-based information can thus be universally applied across any emitter and listener positions within the scene, an idea previously explored in interactive sound propagation via acoustic radiance transfer [Re128, Re129]. The training strategy for our neural network model capitalizes on this by learning scene-dependent features that are reusable and associating them with specific emitter and listener configurations. This methodology enables the model to discern that variations in impulse responses across different emitter-listener locations are inherently tied to the scene's geometry.

Inspired by this concept, we have developed a two-stage model. The initial stage is dedicated to decomposing the audio scene's features to isolate independent geometric features of the scene and to link the emitter and listener within this context. The subsequent stage integrates these features to accurately render the binaural impulse responses. In the sections that follow, we will delve into the interactive acoustic radiance transfer mechanism before detailing our model's structure and functionality comprehensively. This approach not only enhances the precision of spatial audio rendering but also enriches the immersive quality of audio-visual experiences by leveraging the intricate relationship between sound propagation and scene geometry.

Background on Interactive Acoustic Radiance Transfer. Acoustic radiance transfer represents a foundational methodology for simulating sound propagation within complex

room configurations. This classical approach is delineated in three primary steps, as illustrated in Figure 9.1. 1) Scattering from the Emitter to Bounce Points: Initially, the perimeter of the scene is segmented into N distinct bounce points. Sound energy is then disseminated from the sound emitter to each of these points, effectively scattering the initial acoustic energy throughout the scene. 2) Emission and Propagation from Bounce Points: At this stage, sound energy radiates omnidirectionally from each bounce point, traversing the scene until it either attenuates or encounters another bounce point, where the process is repeated. The trajectory and intensity of this sound energy are cataloged in an echogram for each bounce point, capturing the energy-time relationship and providing a detailed map of sound propagation. 3) Energy Collection by the Listener: In the final step, the listener accumulates the acoustic energy emanating from all bounce points, synthesizing the composite sound field based on the cumulative contributions from the entire scene.

Interactive adaptations of this model incorporate a precomputed linear acoustic transfer operator [Re128, Re129] to accurately describe the movement of sound radiance among the bounce points arrayed across the scene's surfaces. Conceptually, this operator embodies the scene-specific characteristics that remain constant across various emitter and listener positions, facilitating an efficient recalibration of the impulse response for different spatial configurations by adjusting for propagation delays relative to the distance from the bounce points.

Inspired by this systematic and modular approach to sound propagation, we propose a neural network model that mirrors this structure with similarly decoupled components. Our model is designed to capture and reuse scene geometry information effectively across any emitter and listener pairing. By segregating the functions of scattering, bouncing, and gathering into discrete modules within the network, we aim to leverage the static nature of scene geometry to dynamically generate accurate binaural impulse responses. This methodology not only honors the principles of acoustic radiance transfer but also introduces a flexible and computationally efficient framework for spatial audio rendering in virtual environments.

Implicit Neural Representation for Audio Scenes. ***INRAS*** is structured around

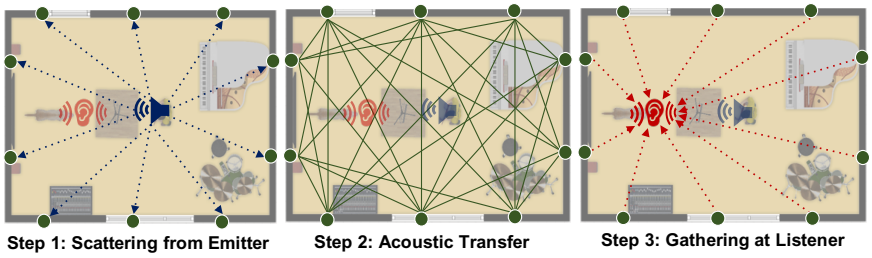

Figure 9.1: Acoustic radiance transfer steps overview. (Figure from [KS4])

two core components: (a) decomposition of audio scene features, and (b) prediction of spatial binaural impulse responses. Within component (a), the model employs three parallel modules designed to capture distinct aspects of sound propagation: 1) the Scatter Module focuses on learning features that link the sound emitter to various bounce points within the scene. It mirrors the process of scattering initial radiance from the emitter across the scene in traditional acoustic radiance transfer. 2) The Bounce Module is tasked with understanding scene-dependent features that remain constant across all combinations of emitter and listener positions, effectively capturing the essence of the scene's acoustic properties. 3) The Gather Module learns how to associate the listener with the bounce points, completing the path of sound propagation from its source to the recipient. In component (b), the features derived from these three modules are integrated to produce directional and binaural impulse responses, thereby synthesizing the sound as it would be perceived by a listener within the scene. An overview of the system's structure is depicted in Figure 9.2.

Scatter Module. Analogous to the initial step in acoustic radiance transfer where radiance scatters from the emitter to bounce points, the Scatter module's functionality hinges on the relative distances between the emitter and each bounce point. The scene's surface is discretized into N uniform bounce points with 3D locations $\{b_i\}_{i=1}^{N} \in \mathbb{R}^3$. We calculate the relative distances from the emitter position s to all bounce points $\{d_{b_i}^{s}\}_{i=1}^{N}$, utilizing these distances as inputs to underscore the emitter's awareness of the scene's geometry. This approach aids in learning continuous features for varying emitter positions. Sinusoidal encoding

transforms the input $\{d_{b_i}^s\}_{i=1}^N$ into a higher-dimensional space, drawing inspiration from techniques used in graphical implicit neural representationsn [Re17]. The learned function F_Θ, parameterized by a fully connected network, outputs feature $I = F_\Theta(\{d_{b_i}^s\}_{i=1}^N) \in \mathbb{R}^{N\times D}$, where D represents the dimension of the feature space. Our experimentation indicates that between 40 to 60 bounce points adequately represent a scene's structure. Further investigation into the selection and impact of bounce points on model performance is detailed in our ablation studies within the Results section, offering insights into the optimization of this process for enhanced spatial audio rendering.

Bounce Module. The Bounce Module is crafted to produce features that capture the geometry of static scenes, which are invariant across various emitter and listener positions. This module's aim is to distill the essence of scene-dependent features into a form that can be universally applied, regardless of the specific locations of the sound source and receiver within the environment.

To achieve this, we employ a learning function U_Φ, realized through a multi-layer perceptron (MLP) equipped with residual connections. This architectural choice facilitates the deep network's ability to learn complex relationships without succumbing to the vanishing gradient problem, thereby enhancing its capacity to model intricate scene geometries effectively.

The function takes as its input the positions of all bounce points within the scene, denoted as $\{b_i\}_{i=1}^N \in \mathbb{R}^3$ and processes these data points to generate a set of features $Q = U_\Phi(\{b_i\}_{i=1}^N) \in \mathbb{R}^{N\times D}$. Here, D represents the dimensionality of the feature space, encapsulating the geometric characteristics of the scene that influence sound propagation.

Through this module, we abstract the static physical attributes of the scene into a high-dimensional feature space, enabling the model to understand and utilize the inherent acoustic properties of the environment. These features serve as a foundational component in the computation of spatial audio, ensuring that the synthesized binaural impulse responses accurately reflect the unique acoustic signature of the scene.

Gather Module. The Gather Module operates in tandem with the principles underlying the

Scatter module, with its primary objective being to forge a connection between the listener and the bounce points scattered throughout the scene. This module focuses on the listener's perspective, calculating the relative distances from the listener's position l to all bounce points within the scene: $\{d^l_{b_i}\}^N_{i=1}$.

To process these distances and extract meaningful acoustic features, we employ sinusoidal encoding—a technique that effectively captures the nuances of spatial relationships in a high-dimensional space. This encoding serves as the foundation for learning a function G_Ψ, which is parameterized by a fully connected neural network. The network is designed to transform the encoded relative distances into a sophisticated output feature $O = G_\Psi(\{d^l_{b_i}\}^N_{i=1}) \in \mathbb{R}^{N\times D}$.

By leveraging this approach, the Gather Module adeptly synthesizes the spatial and acoustic information relevant to the listener's position in relation to the scene's geometry. This results in the generation of a feature set that encapsulates the listener's acoustic environment, effectively modeling how sound propagates to and is perceived by the listener from various points within the scene.

Spatial-Time Feature Composition. The Feature Decomposition approach, encompassing the Scatter, Bounce, and Gather modules, initially does not account for temporal dynamics, focusing purely on spatial relationships within the audio scene. However, the inclusion of temporal factors is crucial for a comprehensive representation of sound propagation, despite the potential to complicate and slow down the training process.

Drawing inspiration from the concept of acoustic operator decomposition used in interactive sound propagation [Re129], we can model the energy-time echogram for a specific bounce point $b_i(t)$ through a series of time-domain basis functions $\{\tau^k(t)\}^K_{k=1}$ using a linear combination: $b_i(t) = \sum^K_{k=1} \alpha_k \tau^k(t)$, where α represents the coefficients in the basis space. This approach allows for a compact and efficient representation of temporal dynamics.

Embracing this methodology, we develop a function P_τ, realized via a fully connected network, to derive a set of time-domain basis functions. These functions are designed to be universally applicable across all spatial features, offering a streamlined method to incorporate time into our model without significantly impacting the training efficiency. To achieve this,

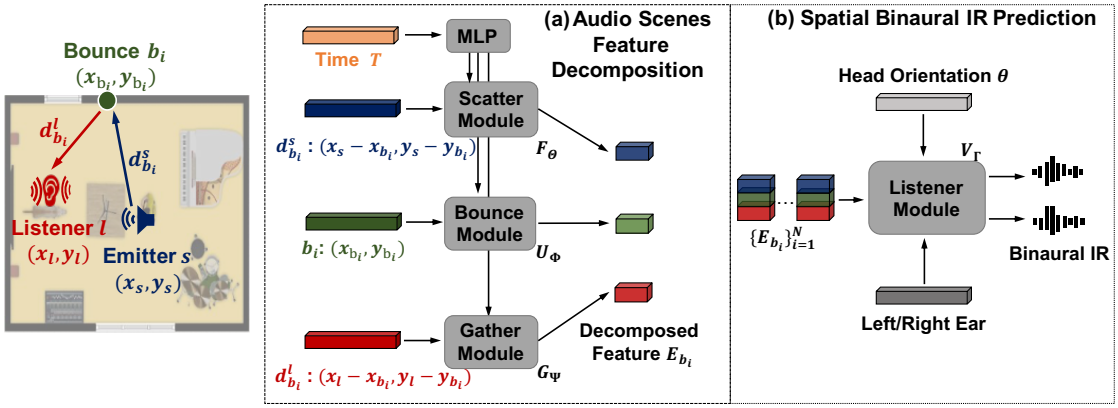

Figure 9.2: System overview of ***INRAS***. (a) Audio Scenes Feature Decomposition: inputs to the scatter/gather module are the relative distances between the emitter/listener locations and bounce points. The bounce module takes all bounce points to generate scene-dependent features. (b) Spatial Binaural IR Prediction: in this stage, the decomposed features are stacked and fed to the Listener module which generates the spatial binaural impulse responses. (Figure from [KS4])

we encode the time samples $\{t_j\}_{j=1}^T$ using sinusoidal encoding, a technique that facilitates the model's ability to capture temporal variations. The resultant output is denoted as $M = P_\tau(\{t_j\}_{j=1}^T) \in \mathbb{R}^{T \times D}$.

By conducting fast matrix multiplication, we integrate the temporal basis functions with the spatial features from each module, generating spatial-time features $\hat{I} = MI^\top$, $\hat{Q} = MQ^\top$ and $\hat{O} = MO^\top$. This process not only enriches the model's representation with essential temporal information but also maintains computational efficiency, enabling the nuanced modeling of sound propagation over time while preserving the agility of the training procedure. Through this sophisticated spatial-time feature composition, we ensure a dynamic and detailed acoustic modeling that captures both the spatial intricacies and temporal fluctuations of sound within a scene.

Listener Module. In the phase dedicated to predicting spatial binaural impulse responses,

the listener module undertakes the crucial task of fusing features to encapsulate the complex interplay between spatial and temporal elements in relation to the listener's position and orientation. This is achieved by concatenating the spatial-time features $\hat{I}, \hat{Q}, \hat{O}$ to form a comprehensive feature set $E = \{\hat{I}, \hat{Q}, \hat{O}\} \in \mathbb{R}^{T\times 3N}$, where each subset $\{E_{b_i}\}_{i=1}^{N} \in \mathbb{R}^{T\times 3}$ epresents the amalgamated features corresponding to the listener (l) and emitter (s) positions relative to each bounce point b_i.

The aggregated feature set E is then introduced to the Listener module, which further integrates the listener's head orientation (θ), encoded by a learnable embedding matrix. This process ensures that the final model output is sensitive not only to the spatial and temporal aspects of sound propagation but also to the listener's orientation within the scene, a critical factor in rendering realistic binaural audio.

Employing a Multi-Layer Perceptron (MLP) within the Listener module, we successfully generate binaural impulse responses in the time domain, denoted as $h = V_\Gamma(E, \theta)$. This step marks the culmination of our modeling efforts, where the complex interdependencies of spatial features, temporal dynamics, and listener orientation are seamlessly integrated to produce high-fidelity binaural impulse responses.

Training and Rendering. All components and modules of INRAS are trained jointly. We use a combination of mean square error loss $L_{\text{mse}} = \|h - \hat{h}\|_2^2$ and multi-resolution STFT loss $L_{\text{mr\_stft}}$ which has been shown effective in modeling audio signals in the time domain [Re116]. The multi-resolution STFT loss first converts the impulse response into frequency-time domain $H = \text{STFT}(h)$ and computes the spectral convergence loss $L_{\text{sc}} = \frac{\||H|-|\hat{H}|\|_2}{\||H|\|_2}$, the magnitude loss $L_{\text{mag}} = \||H|-|\hat{H}|\|_1$ and the phase loss $L_{\text{phase}} = \|\phi(H)-\phi(\hat{H})\|$, our total loss can be summarized as follow:

$$L_{\text{mr\_stft}} = L_{\text{sc}} + L_{\text{mag}} + L_{\text{phase}}, L_{\text{total\_loss}} = L_{\text{mse}} + L_{\text{mr\_stft}} \tag{9.1}$$

Once we obtain the impulse response h, we can render sounds perceived at the listener location by convolving the impulse response with a sound source y. The final sound is denoted as $\hat{y} = h \circledast y$.

9.3 Experiments

Datasets. To evaluate our method, we use the *Soundspaces* dataset, which is comprised of densely sampled pairs of impulse responses meticulously generated through geometric sound propagation techniques [Re130]. This dataset ensures a standardized environment across all scenes, with uniform ceiling heights and the provision of binaural impulse responses captured at four distinct head orientations ($0, 90, 180, 270$ degrees). To ensure a fair and consistent comparison with prior studies, we adjust all impulse responses to a sampling rate of 22050 Hz. Our evaluation focuses on the same six scenes that have been used in previous research, encompassing a diverse range of architectural layouts: two multi-room configurations, two rooms featuring non-rectangular walls, and two standard single rooms with rectangular walls. For each of these carefully selected scenes, we allocate 90% of the data for training purposes, reserving the remaining 10% for testing.

Baseline Methods. Our evaluation framework involves a comprehensive comparison of our method, ***INRAS***, against both contemporary learning-based models and established classical approaches in the field of spatial audio rendering. Specifically, we benchmark ***INRAS*** against Neural Acoustic Fields (NAF) [Re131], representing the forefront of learning-based techniques for acoustic scene modeling. Given that ***INRAS*** introduces a novel approach to encode and compress the acoustic field of a scene, it's also pertinent to assess its performance relative to standard audio encoding methods. To this end, we compare ***INRAS*** with two widely recognized audio encoding techniques: Advanced Audio Coding (AAC) and Xiph Opus. We apply both linear and nearest neighbor interpolation methods to the acoustic fields encoded by AAC and Opus.

Results. Quantitative evaluation results, as presented in Table 9.1, showcase that ***INRAS*** surpasses both traditional audio coding methods and contemporary learning-based approaches across all evaluated metrics. Notably, ***INRAS*** demonstrates a remarkable improvement in C50 and EDT errors over the Neural Acoustic Fields (NAF) method by 43% and 39%, respectively. This indicates a significant enhancement in the accuracy of early re-

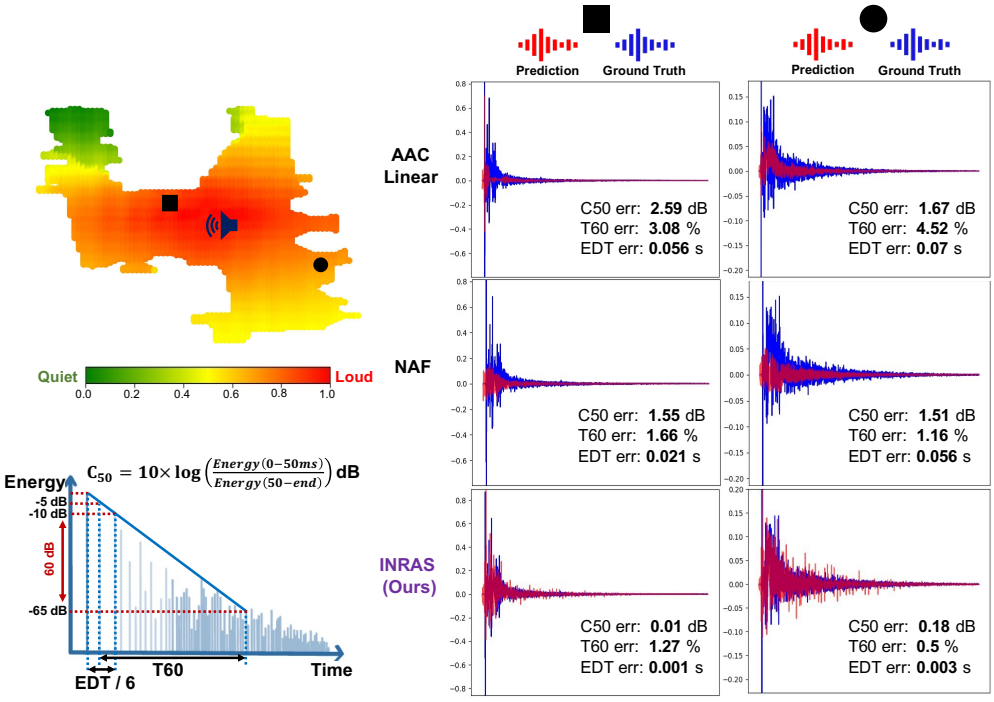

Figure 9.3: Rendered Impulse Responses Waveform Visualization. Top left: the speaker indicates the emitter location. We show examples (right) of rendered waveforms at two listener locations (black square and black circle) rendering by three methods: AAC-Linear, NAF, and ***INRAS*** (blue: GT; red: Prediction). Bottom left: Metrics upon which we evaluate Impulse Response are illustrated. (Figure from [KS4])

flection components within ***INRAS***-rendered impulse responses, bringing them much closer to the ground truth.

Figure 9.3 provides a detailed comparison of impulse response waveforms rendered by AAC-linear, NAF, and the ***INRAS*** method for two illustrative examples. The figure's top left quadrant features a loudness map of impulse responses generated by ***INRAS***, with color coding to represent amplitude levels. The comparative analysis in the two right columns highlights the AAC-linear waveforms' substantial deviations from the ground truth. Although NAF manages to replicate the exponential decay pattern characteristic of reverberation, it falls short in capturing the crucial early reflection components of the impulse

Model\Metric	C50 error (dB) ↓	T60 error (%) ↓	EDT error (sec) ↓	Parameters (Million) ↓	Storage (MB) ↓	Speed (ms) ↓
Opus-nearest	3.58	10.10	0.115	-	181.37	-
Opus-linear	3.13	8.64	0.097	-	181.37	-
AAC-nearest	1.67	9.35	0.059	-	346.74	-
AAC-linear	1.68	7.88	0.057	-	346.74	-
NAF	1.06	3.18	0.031	2.23	8.55	37.86
INRAS (Ours)	**0.6**	**3.14**	**0.019**	**0.67**	**2.56**	**9.47**

Table 9.1: Quantitative evaluation of impulse response quality, storage requirements, and inference speed. Results are indicated on average of six single scene models. (Table from [KS4])

responses, which are pivotal for sound clarity. Conversely, ***INRAS*** adeptly renders both the early reflections and late reverberation phases of the impulse responses, showcasing its comprehensive modeling capabilities.

Moreover, the efficiency of the ***INRAS*** model is underscored by its relatively modest requirement of approximately 0.65 million trainable parameters, translating to less than 3MB of storage and an inference speed of 4ms. This highlights ***INRAS***'s status as a significantly more lightweight and efficient model, offering a potent combination of high fidelity in spatial audio rendering and operational efficiency. Through these evaluations, ***INRAS*** emerges as a leading solution in the field, adept at delivering precise and immersive auditory experiences while maintaining a lean computational footprint.

Multi-condition Training on Multiple Scenes. The strategic decomposition of audio scene features within INRAS enables the training of a singular model capable of accurately representing multiple scenes. This versatility was explored by training INRAS on a diverse set of scenes, each featuring distinct layout characteristics: a multi-room layout, a room with non-rectangular walls, and a room with traditional rectangular walls, as depicted in Figure 9.4. This exploration confirmed INRAS's ability to learn continuous implicit neural representations for varied scene architectures.

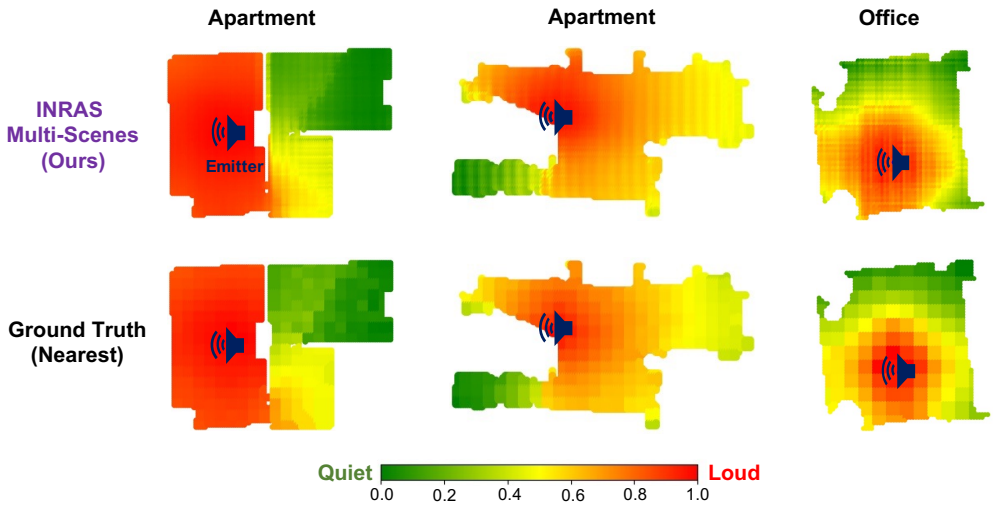

Figure 9.4: Loudness map visualization comparing INRAS multi-scenes rendering of three scenes (Top) vs. Ground truth using nearest neighbors (Bottom). (Figure from [KS4])

Figure 9.4 and Table 9.2 showcase the loudness maps for the three scenes as learned by this unified INRAS model, along with quantitative results demonstrating the model's performance across these multi-scene configurations. Beyond assessing impulse response quality through traditional acoustic parameters, we extended our evaluation to the quality of the final audio signal produced by convolving the impulse response with a sound source. This included calculations of the Signal-to-Noise Ratio (SNR) and the Peak Signal to Noise Ratio (PSNR), metrics critical to gauging the auditory signal's fidelity.

The findings, detailed in Table 9.2, reveal that INRAS when subjected to multi-condition training, maintains not only high-quality outcomes but also achieves superior overall accuracy compared to averaging the performance of other models individually tailored to each scene. Remarkably, to accommodate multi-scene representation, the INRAS model required only a marginal increase in trainable parameters by 0.1M, thus maintaining the storage requirements under 3MB. This efficiency contrasts with other approaches, where the storage needs are typically scaled linearly with the number of scenes represented.

Model\Metric	Multi-scenes	SNR (dB) ↑	PSNR (dB) ↑	C50 error (dB) ↓	T60 error (%) ↓	Storage (MB) ↓
Opus-nearest	✗	3.18	13.35	3.6	10.1	544.11
Opus-linear	✗	3.57	13.45	3.23	8.7	544.11
AAC-nearest	✗	6.48	17.84	1.51	9.64	1040.31
AAC-linear	✗	7.52	18.7	1.57	8.05	1040.31
NAF	✗	-1.54	11.25	1.05	**3.01**	25.65
INRAS (Ours)	✓	**8.06**	**18.80**	**0.68**	4.09	**2.99**

Table 9.2: Quantitative evaluation of INRAS after multi-condition training on three scene layouts. Results for other methods are computed as an average of the three scenes. (Table from [KS4])

Chapter 10

FROM VISION TO AUDIO AND BEYOND

10.1 Motivation

In the preceding chapters, vision-to-audio generation across different levels of audio-visual correspondence was considered through the development of deep learning methods to seamlessly map visual representations to their auditory counterparts through multi-stage or single-stage generative models. While these models have successfully generated high-quality audio and some of them could produce meaningful deep learning features interpreting certain audio-visual characteristics, they do not address broader audio-visual tasks which typically necessitate learning general audio-visual representations. Humans, when faced with a silent video, possess the innate ability to mentally conjure the corresponding sounds, drawing upon our knowledge to associate and creatively imagine the sound. Recent advancements in artificial intelligence have introduced foundational models in vision and language capable of supporting both content association and generation [Re132, Re133, Re134]. Yet, the potential for similar models in the audio-visual domain remains largely untapped.

This work ventures into developing such a model through a unified audio-visual framework designed to learn general audio-visual representations and enable vision-to-audio generation within a unified single system. Constructing a general-purpose audio-visual model presents significant challenges, notably due to the high-dimensional and time-dependent nature of raw video and audio signals, which complicate the understanding of their joint occurrences and demand extensive computational resources for training. Existing deterministic models based on contrastive learning and masked autoencoders [Re53, Re54] are ill-suited for generative tasks, and sophisticated models are often required to produce high-quality audio through latent space modeling [Re94, Re135, Re43] and multi-stage processes [Re136, Re137, Re77].

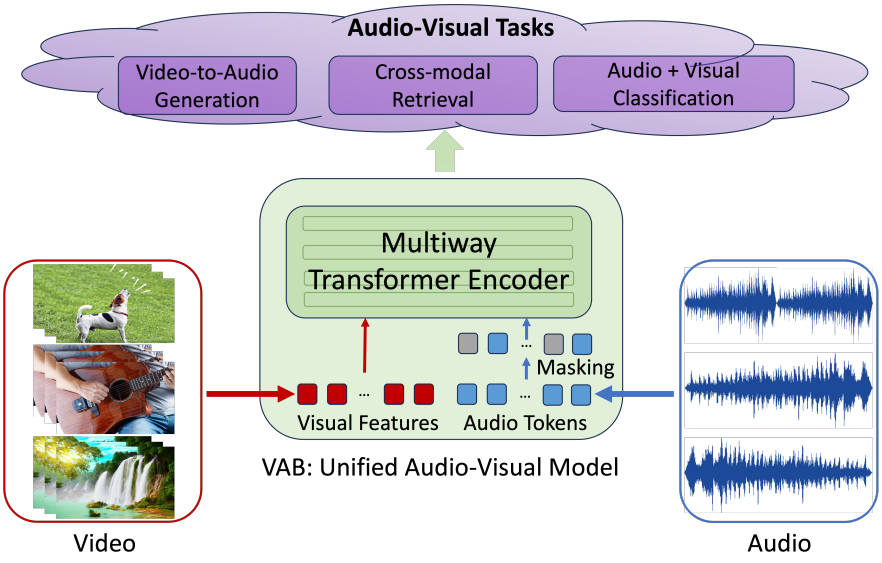

Figure 10.1: VAB is a unified audio-visual model capable of supporting various audio-visual tasks within a single framework.

Addressing these challenges, we introduce 'Vision to Audio and Beyond' (VAB), an innovative framework centered around a pre-training task that predicts masked audio from visual inputs, facilitating both audio-visual representation learning and audio generation. This pre-training occurs within the latent space, leveraging discrete audio tokens derived from a publicly available pre-trained neural audio codec and frame-level visual features extracted by a self-supervised pre-trained image encoder. The VAB model, utilizing an encoder-only multi-way transformer, predicts masked audio tokens from visual features with a variable masking scheme. After pre-training, VAB can act as a uni-modal or multi-modal encoder, ready for fine-tuning in cross-modal retrieval and classification tasks, and supports zero-shot visual-conditioned audio generation with efficient parallel decoding.

Our comprehensive experiments across various downstream tasks—ranging from vision-to-audio generation and audio-visual event classification to cross-modal retrieval and audio-only classification—demonstrate the efficacy and advantages of our approach. The VAB

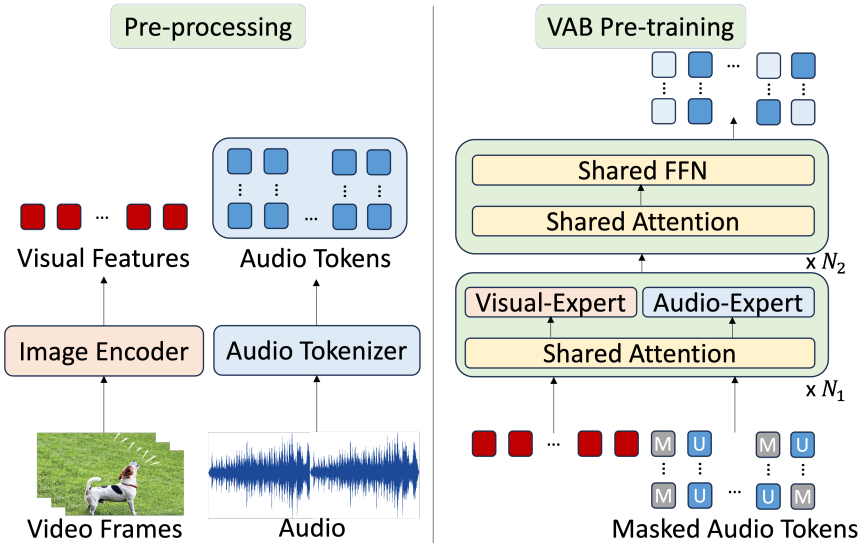

Figure 10.2: Pre-processing (left) and masked audio token prediction pre-training (right) of VAB framework.

model not only efficiently generates high-fidelity audio from silent videos, achieving a significant speedup over autoregressive approaches, but also delivers competitive performance in classification and retrieval tasks, underscoring its potential to redefine the landscape of audio-visual synthesis and understanding.

10.2 Method

We introduce "Vision to Audio and Beyond" (VAB), a comprehensive audio-visual framework designed for generating high-quality audio from silent videos and capturing semantic audio-visual representations for a range of applications. Initially, we detail the pre-processing phase, where raw audio and visual signals are transformed into latent spaces using a pre-trained audio neural codec and an image encoder. With the framework utilizing audio tokens and frame-level visual features as inputs, we delve into the VAB components, focusing on a self-supervised pre-training task aimed at predicting masked audio tokens based on visual features. This crucial stage allows VAB to build its representational capacity and master

audio generation from video. Following this, we discuss the application of the pre-trained VAB model to vision-to-audio generation and its fine-tuning for various downstream tasks, showcasing the framework's versatility and effectiveness.

10.2.1 Transform Audio and Video into Latent Spaces

Two fundamental motivations underpin our decision to convert audio and visual data into latent spaces. The first is grounded in existing literature [Re94, Re135, Re43], which suggests that conducting generative modeling within latent spaces not only accelerates training convergence but also enhances the quality of the generated audio samples. The second motivation addresses the computational challenges associated with processing raw video frames. For example, processing just ten seconds of video at a frame rate of 1fps using a standard Vision Transformer (ViT) [Re138] results in 1960 patches, a volume that remains daunting despite attempts to alleviate it, such as the use of tubelets.

To tackle this issue, we've employed a frozen pre-trained image encoder like CLIP [Re139] to distill frame-level features. This approach significantly streamlines the visual sequence length and leverages the rich semantic insights gained from image-text pre-training. Interestingly, we found that these robust image-text features are highly adaptable and effectively capture the intricate relationships between audio and video. This strategy not only reduces the computational burden but also minimizes the potential information loss that might occur when compressing audio tokens, striking a balance between efficiency and fidelity in representing audio-visual connections.

To convert audio waveforms into discrete tokens, we investigate two pre-trained neural codec variants: DAC [Re140] and Encodec [Re9]. These codecs are distinguished by their self-supervised training on audio reconstruction tasks, which does not require any labeled data. As a result, the audio tokens generated by DAC and Encodec represent highly compressed versions of the original audio waveforms. Utilizing K residual vector quantization (RVQ), both codecs transform a 1D audio waveform $A_w \in \mathbb{R}^{T_a}$ sampled at 16kHz into a series of audio tokens $A \in \mathbb{N}^{K \times S}$, where K denotes the number of residual codebooks, and $S = T_a/d_a$

with $d_a = 320$ being the downsampling factor for both codecs. Consequently, a 10-second audio clip is encoded into $A \in \mathbb{N}^{K \times 500}$ tokens.

DAC employs $K = 12$ codebooks, creating a hierarchical structure where lower-level codes encapsulate coarse acoustic features and higher-level codes detail finer acoustic nuances. In contrast, Encodec utilizes only $K = 4$ codebooks, resulting in a comparatively lower quality of audio reconstruction than DAC. Our research encompasses a detailed examination of both DAC and Encodec tokens. During the pre-training phase of our Vision to Audio and Beyond (VAB) framework, we utilized the first four levels of audio tokens $A_c = A^{0:4,500}$ from both codecs. For DAC's additional eight levels, we adopted a strategy akin to Vampnet [Re137], implementing a coarse-to-fine model specifically tailored for audio generation, thereby enriching our study with a versatile approach to audio tokenization and its implications for generative audio modeling.

For the image encoding process, we utilize the CLIP image encoder to derive frame-level features at a capture rate of 1 frame per second. This approach yields a concise representation of 10 seconds of video features, denoted as $V \in \mathbb{R}^{10 \times d}$, where d represents the dimensionality of CLIP image features. It is crucial to note that both the audio tokens and the frame-level visual features are extracted in advance of the VAB pre-training phase. This preparatory step significantly enhances our ability to efficiently model both temporal dynamics and audio-visual relationships during the pre-training process.

10.2.2 Conditional Masked Audio Token Prediction

Masking: For our VAB pre-training task, we focus on predicting masked audio tokens conditioned on visual features, using the given audio tokens A_c and visual features V. The process begins with randomly masking a portion of the audio tokens, employing a variable masking strategy. This involves selecting the masking ratio M_r from a truncated Gaussian distribution centered at 0.55, with a standard deviation of 0.25, and truncated between 0.5 and 1. Such variability ensures a diverse and appropriate range of masked audio tokens, critical for fostering the learning of both representations and generative capabilities.

To implement this, we incorporate the existing codebook embeddings from neural codecs and introduce a new $[MASK]$ token for the masked portions. These masked audio tokens $A_m \in \mathbb{N}^{4,500}$ undergo embedding, and the resultant embeddings $A_{\text{emb}} \in \mathbb{R}^{500,d_{\text{emb}}}$ are combined, with d_{emb} representing the embedding dimension. To synchronize the dimensions, visual features V are linearly projected to match d_{emb}, creating visual embeddings V_{emb}. Additionally, modality-specific embeddings, E_v for visual and E_a for audio, are added to their respective embeddings.

Finally, these enriched visual and audio embeddings are concatenated to generate the complete input sequence $x = [V'\text{emb}, A'\text{emb}]$, where $V'\text{emb} = V\text{emb} + E_v$ and $A'\text{emb} = A\text{emb} + E_a$. This setup forms the backbone of our approach, optimizing the model to understand and predict the complex interplay between audio and visual information effectively.

Multiway Transformer Encoder: The VAB framework adopts an architecture reminiscent of the Multiway Transformer Encoder [Re141, Re132], incorporating features from recent advancements in transformer technology. Each transformer layer within VAB is equipped with bidirectional multi-head attention mechanisms that cater to both audio and visual embeddings. This setup is further enhanced by layer normalization, feed-forward networks, and residual connections. In a VAB model composed of N layers, the architecture differentiates between modality-specific and shared processing layers. For the initial N_1 layers, modality-specific feed-forward networks are deployed, tailoring the learning process to the unique characteristics of either audio or visual information. The subsequent $N_2 = N - N_1$ layers utilize shared feed-forward networks, facilitating cross-modal integration and learning. This design choice enables the shared bidirectional self-attention mechanism to foster associative learning between audio and visual embeddings, enriching the model's ability to correlate these two modalities. The early, modality-specific layers act as dedicated experts for processing either audio or visual data, allowing them to function as standalone encoders for tasks focused on a single modality. Meanwhile, the later, joint layers are optimized for the more complex vision-to-audio generation task, leveraging the combined strengths of both modalities. Finally, VAB employs multiple linear projection heads tasked with pre-

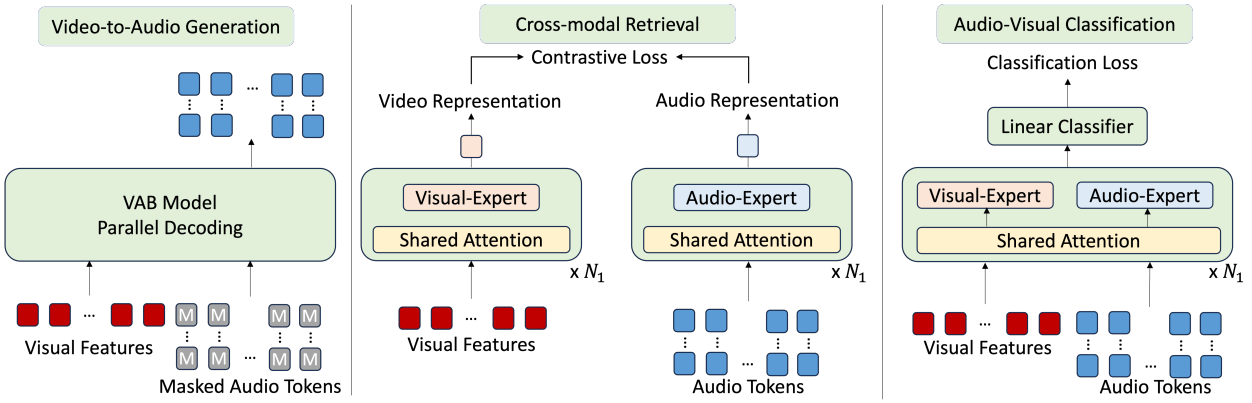

Figure 10.3: After the pre-training phase, VAB allows for zero-shot video-to-audio generation (left). Moreover, it can undergo representation adaptation fine-tuning to facilitate cross-modal retrieval through contrastive loss (middle) and accommodate classification tasks by incorporation of a linear classifier (right).

dicting each level of the masked audio tokens. This intricate structure ensures that VAB can adeptly handle a wide spectrum of tasks, from simple modality-specific encoding to the intricate process of generating audio from visual inputs.

Let A_u be the set of all unmasked audio tokens and VAB model parameters θ. The objective of the pre-training is to minimize the negative log-likelihood

$$l_{\text{vab}} = -\sum_{\forall a \in A_m} \log p(a|A_u, V, \theta) \tag{10.1}$$

The overview of VAB pre-processing and pre-training is shown in Figure 10.2.

10.2.3 Zero-Shot Video-to-Audio Generation

Following the completion of VAB's pre-training, we employ an efficient iterative decoding technique akin to prior masked generative modeling methods to generate audio tokens [Re142, Re143, Re137]. Initially, all audio tokens are set to $[MASK]$ and paired with visual features V, mirroring the pre-training setup. During each decoding iteration $t \in [0, t_T]$, we predict

the token for the current masked audio token, resulting in $\hat{a}_t$. To evaluate the prediction's confidence score z, we incorporate the prediction log-probabilities and add temperature-annealed Gumbel noise:

$$z(\hat{a}_t) = \log(p(\hat{a}_t)) + \alpha \cdot g_t. \tag{10.2}$$

where g_t is a sample from Gumbel noise, distributed independently and identically, and α is the temperature parameter, which decreases linearly to 0 across the sampling iterations. The sampled tokens are then ranked by their confidence scores, and we select the k tokens with the lowest confidence for remasking in the next iteration. The quantity of tokens to be remasked, $k = \gamma(\frac{t}{t_T})N$, varies dynamically, with N being the total number of tokens, and γ adhering to a cosine schedule. This approach ensures a gradual increase in the number of tokens being replaced as iterations advance, starting with fewer replacements in early iterations and escalating in later ones.

Moreover, to enhance the model's adherence to visual content, classifier-free guidance is implemented [Re144]. This nuanced strategy allows for a more controlled and visually coherent audio token generation, optimizing the audio output for more accurate representation of the visual stimuli.

The audio tokens obtained from sampling are set to be transformed into audio waveforms using either DAC or Encodec codecs. However, it's observed that for DAC, the audio quality derived from decoding solely the coarse levels of audio tokens is not satisfactory [Re77, Re137]. To address this, we develop a supplementary coarse-to-fine model that enhances the fine-level audio tokens A_f based on the given coarse tokens A_c. This model adopts a bidirectional transformer encoder architecture similar to VAB, but differs by not incorporating modality-specific feed-forward networks and by utilizing audio tokens from all levels, rather than just the coarse levels, as inputs.

The training of this model centers on the masked audio token prediction task, focusing exclusively on masking and predicting the fine-level tokens. The training objective for the

model θ_{c2f} aims to minimize the following function:

$$l_{\text{c2f}} = -\sum_{\forall a \in A_{f,m}} \log p(a|A_{f,u}, A_c, V, \theta_{\text{c2f}}). \tag{10.3}$$

For the inference phase, the model uses the generated coarse audio tokens and visual features as conditioning inputs, employing iterative decoding to refine the generation process. Contrary to the sequential nature of autoregressive generation, our method enables simultaneous token generation, which significantly cuts down the number of inference steps required. This not only streamlines the generation process but, in certain instances, also improves upon or matches the quality of audio produced by traditional autoregressive methods, thus offering a more efficient and effective approach to audio waveform generation.

10.2.4 Adaptation of VAB for Retrieval

Following the completion of VAB's pre-training, the model's learned representations are primed for adaptation to an array of audio-visual tasks through fine-tuning. We thus embark on fine-tuning the first N_1 layers, which are modality-specific, employing contrastive loss to better align the audio and visual modalities, particularly for retrieval tasks. Although integrating masked prediction and contrastive loss during pre-training may appear intuitive, our preliminary experiments indicated that such a dual-focused approach hindered the success of both tasks, leading to convergence issues.

Moreover, we discovered that beginning contrastive training from scratch on the initial N_1 layers demanded a significantly longer training period to achieve comparable results to those initialized from the VAB model pre-trained with masked audio token prediction. Likewise, initiating fine-tuning for masked audio token prediction from a model pre-trained with contrastive loss did not facilitate the masked prediction task's convergence. These findings informed our decision to sequence the training phases, positioning contrastive training as a post-masked prediction pre-training fine-tuning task.

Specifically, we conduct two separate forward passes to extract audio and visual output features from the corresponding tokens and frames, respectively, without applying masking

to audio tokens at this juncture. These output features are then subject to average pooling and normalization. The fine-tuning phase employs the standard contrastive loss L_c:

$$l_c = -\frac{1}{N}\sum_{i=1}^{N}\log\left[\frac{\exp s_{i,i}/\tau}{\sum_j \exp s_{i,j}/\tau}\right] \tag{10.4}$$

where $s_{i,j} = ||a_i^c||^T||v_j^c||$, a^c and v^c are audio and video representations, and τ is the learnable temperature value initialized with 0.05.

10.2.5 Adaptation of VAB for Classification

The latent representations derived from our VAB model offer the flexibility to be tailored for a range of additional tasks, such as classification. For tasks focused on a single modality, we direct either audio tokens or visual features to the initial N_1 layers of the VAB model, which are specific to each modality. In the case of joint audio-visual classification, we combine both audio tokens and visual features, following the same procedure as in the pre-training phase of VAB, without masking any audio tokens. The output features from these N_1 layers are then subjected to average pooling, and a linear classifier is appended for the fine-tuning process across all classification tasks.

While the VAB model fine-tuned post pre-training demonstrates commendable performance in classification tasks, initiating fine-tuning from a model that underwent contrastive fine-tuning exhibits superior results. This enhancement likely arises from the shared attributes between retrieval and classification tasks, notably the non-requirement of audio token masking, facilitating a smoother transition and knowledge transfer. This benefit is especially pronounced in tasks involving audio-only and combined audio-visual classification, underscoring the versatility and adaptability of the contrastive fine-tuned VAB model as a foundation for classification tasks.

10.3 Experiments

Implementation. For the pre-training of VAB, we utilized AudioSet [Re86] and VGGSound [Re145], two extensive audio-visual datasets derived from YouTube. From AudioSet,

we were able to compile approximately 1.57 million videos out of the original 2 million, due to some videos being unavailable. For VGGSound, we employed its training set, which consists of around 177,000 videos. Audio tokens were extracted using the pre-trained Encodec and DAC models at a 16kHz sampling rate. Video frames were processed using the eva-CLIP image encoder, as featured in BLIP [Re146, Re147], to obtain CLIP embeddings at a frame rate of 1fps. Both audio tokens and CLIP embeddings underwent preprocessing and were stored ahead of the VAB model training.

Our experiments explored two variations of the model: VAB-Encodec/VAB-DAC, which served as the standard for reporting and comparisons across tasks, and VAB-DAC-Test, utilized for ablation studies and detailed examinations. The VAB-Encodec and VAB-DAC configurations consist of 24 layers each, with dimensions of 1024 and 16 attention heads, amounting to a total of 403M parameters. The initial 12 layers are allocated for modality-specific processing. The VAB-DAC-Test model is structured with 768-dimensional transformer layers, featuring 12 attention heads across a total of 20 layers, with the first 12 designated for modality-specific expertise. For the VAB's pre-training, an AdamW optimizer was employed, targeting a learning rate of 2e-4, complemented by a cosine scheduler. The entire pre-training phase and the fine-tuning for contrastive loss were executed on a single A100 (80G) GPU. Subsequent classification tasks leveraged an A100 (40G) GPU, illustrating the computational efficiency and scalability of our approach in handling extensive audio-visual data and sophisticated model architectures.

Zero-Shot Video-to-Audio Generation. We conducted a comprehensive evaluation of video-to-audio generation on the VGGSound test set, benchmarking our VAB-Encodec and VAB-DAC models against established baselines such as SpecVQGAN [Re41], IM2WAV [Re42], Diff-Foley [Re44], and FoleyGen [Re43]. This evaluation encompassed generating 10 seconds of audio for each entry within the VGGSound test set, totaling 15,546 samples. For the VAB models, we utilized 16 decoding steps and applied a classifier-free guidance scale of 5. For the DAC coarse-to-fine model specifically, we increased the decoding steps to 36.

To objectively assess the quality of the generated audio, we employed two metrics: Fréchet

Methods	FAD ↓	KLD ↓	Speed (s) ↓
SpecVQGAN	6.63	3.78	7.2
IM2WAV	6.32	**2.54**	289.5
Diff-Foley	6.40	3.15	4.4
FoleyGen	**2.59**	2.89	6.9
VAB-DAC (Ours)	3.24	2.84	1.3
VAB-Encodec (Ours)	2.69	2.58	**0.4**

Table 10.1: Quantitative evaluations for video-to-audio generation on the VGGSound test set. Values in bold indicate the best value.

Audio Distance (FAD) [Re148] and Kullback–Leibler Distance (KLD). FAD serves as a collection-based metric, gauging the similarity between generated and ground truth audio features, as characterized by the VGGish Network [Re92] trained on AudioSet. KLD, on the other hand, measures the divergence between the label distributions of each generated audio sample and its corresponding ground truth, using predictions from a pre-trained PaSST model [Re149]. FAD correlates with human auditory quality perception, while KLD offers insight into the accuracy of the audio categories represented.

Additionally, we measured the inference speed of each model, defined as the average time required to synthesize a 10-second audio clip from a video on an A6000 GPU. We used open-sourced pre-trained models for SpecVQGAN, IM2WAV, and Diff-Foley to generate their respective samples. For FoleyGen, we meticulously replicated the model as described in its publication.

Our findings, detailed in the table 10.1, underscore the advantage of utilizing state-of-the-art discrete audio tokens with multiple codebooks, as seen in FoleyGen and VAB, which outperform previous methods in audio quality. With its innovative masked audio token prediction and iterative decoding strategy, VAB significantly enhances inference speed, surpassing all preceding models by a considerable margin. Although VAB's FAD and KLD

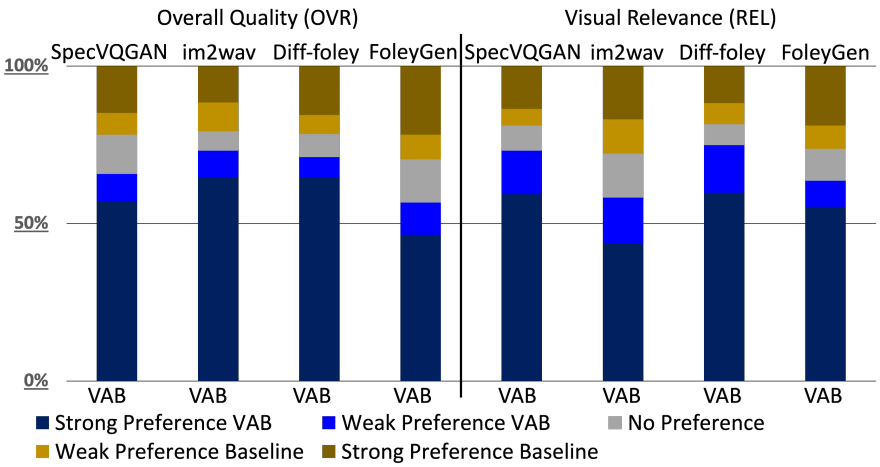

Figure 10.4: Human evaluations for video-to-audio generation. Left: overall quality of the video. Right: visual relevance to the audio.

scores marginally trail behind the leading approaches, it demonstrates a consistently high performance across both metrics, establishing a well-rounded balance between quality and efficiency in audio generation.

To further explore the perceptual nuances between our generated audio samples and those from baseline models, we implemented a subjective evaluation through a carefully designed survey. This survey aimed to appraise two key perceptual attributes of the generated audio: Overall Quality (OVR) and Visual Relevance (REL). To bolster the credibility of our results, we meticulously selected 150 samples from a class-balanced subset of the VGGSound test set for evaluation with each method involved in the study.

Participants in the study engaged in an A-vs-B comparative rating task, where they were presented with pairs of audio samples: one produced by a baseline model and the other by our leading VAB-Encodec model. Participants were asked to choose between five response options, indicating either a strong or weak preference for sample A or B, or stating no preference. The outcomes of this evaluation, illustrated in figure 10.4, reveal a clear preference for the VAB-Encodec model across both assessed criteria. This preference was maintained even

	AudioSet Eval Subset			VGGSound Eval Subset			MSR-VTT (Zero-shot)		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Video → Audio									
CAV-MAE [Re53]	18.8	39.5	50.1	14.8	34.2	44.0	13.3	29.0	40.5
VAB-DAC (Ours)	35.5	61.8	72.4	30.8	59.6	70.8	**13.8**	30.6	40.1
VAB-Encodec (Ours)	**39.5**	**65.4**	**74.6**	**33.5**	**63.3**	**74.3**	14.2	**31.1**	**42.0**
Audio → Video									
CAV-MAE [Re53]	15.1	34.0	43.0	12.8	30.4	40.3	7.6	19.8	30.2
VAB-DAC (Ours)	37.0	61.8	70.8	33.1	**62.7**	**73.8**	**12.0**	**27.3**	**36.2**
VAB-Encodec (Ours)	**37.5**	**64.0**	**73.7**	**34.9**	**62.7**	**73.1**	9.6	23.3	32.9

Table 10.2: Cross-modal retrieval results on AudioSet, VGGSound, and MSR-VTT. Values in bold highlight the best performance.

against baseline methods that might have achieved superior FAD or KLD scores, underscoring the perceptual appeal and efficacy of the VAB-Encodec model in generating audio that resonates more effectively with human listeners.

Audio-Visual Retrieval. In this evaluation, we focus on analyzing the effectiveness of the VAB model's learned representations in both audio-to-visual and visual-to-audio retrieval tasks. Initially, we fine-tune the VAB model applying a contrastive loss, leveraging the identical dataset used during its pre-training phase. To gauge the retrieval efficacy, we utilize the methodology outlined in CAV-MAE [Re53] for conducting retrieval assessments with audio-visual samples derived from the AudioSet and VGGSound evaluation sets. Additionally, we broaden our assessment scope to encompass zero-shot retrieval performance on the MSR-VTT test set [Re150].

The evaluation process involves passing audio tokens and visual features through the VAB model in separate forward passes. We then aggregate the encoder outputs via mean pooling and normalize them to obtain distinct audio and visual representations. Based on

the cosine similarity between these representations, we calculate the retrieval recall metrics at ranks 1, 5, and 10 (R@1, R@5, R@10).

The comparative results, detailed in the table 10.2, highlight the VAB model's superior performance on AudioSet and VGGSound compared to CAV-MAE, showcasing a significant improvement (up to twice the performance enhancement on both AudioSet and VGGSound). Additionally, our findings indicate that audio-to-video retrieval consistently outperforms video-to-audio retrieval across all evaluated datasets, underscoring the VAB model's proficiency in bridging audio and visual modalities effectively for retrieval tasks.

Audio-Visual Event Classification. In this study, we assess the performance of the

	VGGSound (Acc.)↑			AS-2M (mAP)↑			AS-20K (mAP)↑		
Method	V+A	V	A	V+A	V	A	V+A	V	A
Audio-visual Models									
G-Blend [Re151]	-	-	-	41.8	18.8	32.4	37.8	22.1	29.1
Perceiver [Re152]	-	-	-	44.2	25.8	38.4	-	-	-
Attn AV [Re153]	-	-	-	44.2	25.7	38.4	-	-	-
CAV-MAE [Re53]	65.5	47.0	59.5	51.2	26.2	46.6	42.0	19.8	37.7
MBT [Re47]	64.1	51.2	52.3	49.6	31.3	41.5	43.9	27.7	31.3
MAViL [Re54]	**67.1**	50.9	**60.8**	**53.3**	30.3	**48.7**	**44.9**	24.8	**41.8**
VAB-DAC (Ours)	63.9	**55.4**	48.2	47.0	33.3	36.2	38.9	28.3	28.8
VAB-Encodec (Ours)	65.2	55.1	51.3	47.7	**33.5**	38.6	38.7	**29.0**	29.0

Table 10.3: Comparison to previous audio-visual models on VGGSound, AS-2M, AS-20K in audio-visual (V+A), video-only (V) and audio-only (A) classification tasks. Values in bold represent the best performance.

VAB model's representations in the task of audio-visual event classification. Leveraging the contrastively fine-tuned VAB model, we fine-tune it further on three diverse datasets: AudioSet-20K, AudioSet-2M, and VGGSound. During this stage of fine-tuning for classifi-

cation, we maintain the initial N_1 layers and append a linear classification head. This setup allows us to fine-tune our model using audio-only (A), video-only (V), and combined audio-visual (V+A) data, offering a comprehensive evaluation of the model's capability in both single-modal and multi-modal representation learning.

The classification results, as outlined in the table 10.3, demonstrate that both VAB-DAC and VAB-Encodec models deliver robust performances across the A, V, and V+A tasks, with VAB-Encodec showing a marginal edge. Remarkably, the V classification scenario outshines previous approaches across all datasets, underscoring the strength of utilizing frame-level CLIP embeddings for visual feature representation. However, a noticeable disparity is observed in the A (audio-only) classification performance compared to leading methods. This divergence can be attributed to the inherent lossy nature of audio token representation, which, in the absence of visual cues, poses greater challenges for accurate audio categorization. This phenomenon resonates with insights from self-supervised learning research in the image domain involving quantized tokens [Re143]. Despite this, the pre-training phase of VAB considerably enhances the model's effectiveness over training from scratch with visual features and audio tokens, affirming the value of VAB's approach in leveraging advanced representation learning for audio-visual classification tasks.

Audio-only Classification. To gauge the versatility and applicability of our VAB model's audio representations beyond its initial training scope, we extend its application to tasks that are strictly speech or audio-centric. Drawing inspiration from MAViL [Re54], we carry out experimental evaluations on the Environmental Sound Classification (ESC-50) [Re154] and Speech Commands (SPC-v1) [Re155] datasets, focusing solely on fine-tuning the audio component of the VAB model.

The outcomes, detailed in the table 10.4, showcase that VAB delivers performances that are on par with, or even exceed, those of recently developed supervised and self-supervised models. This achievement highlights VAB's remarkable capability for generalization and transfer learning, demonstrating its effectiveness in adapting from an audio-visual self-supervised pre-training environment to tackling purely audio-based tasks.

Method	ESC-50	SPC-1
AST [Re156]	88.7	95.5
SS-AST [Re157]	88.8	96.0
Aud-MAE [Re158]	94.1	96.9
MAViL [Re54]	**94.4**	**97.4**
VAB-DAC (Ours)	89.2	95.1
VAB-Encodec (Ours)	91.4	96.1

Table 10.4: ESC-50 and SPC-1 classification accuracy

Chapter 11

CONCLUSION

In this thesis, I have considered vision-to-audio generation, approaching the topic through various lenses and scales and proposing a suite of methodologies to obtain outcomes for this intricate task. My thesis work commenced with a thorough review of fundamental concepts and related works in Chapter 2, which established the critical foundation necessary for understanding the rest of my thesis.

As described in this thesis, I followed a bottom-up approach to identify the building blocks of learning audio and vision correspondence and audio generation. In particular, in Chapter 3 a system specifically designed for generating music from top-down views of piano performances is showcasing the initial step towards bridging visual inputs with musical outputs. In Chapter 4, we extend our study to the generation of music from arbitrary human movements widening the applicability of video-to-music generation techniques. Chapter 5 presents a generalized video-to-music pipeline, offering a comprehensive framework for audio generation from visual inputs, signifying a pivotal advancement in the field. In these studies, the common thread is using a preset representation.

In further studies, I addressed how such representation can be learned implicitly. In particular, in Chapter 6, a self-supervised learning model that adeptly correlates temporal series data, such as sequences of human body poses, without relying on manual annotations is described. This chapter illuminates the potential of self-supervised techniques in capturing the dynamics of human movement. Building on the insights from Chapter 6 as a building block, Chapter 7 integrates it in the application of instrument music generation from musicians' body movements, further expanding our methodological repertoire for audio generation from visual cues. Chapter 8 introduces a novel approach for synthesizing object impact sounds

from video content, broadening the audio generation landscape to encompass a wider array of sound types. Chapter 9 presents an innovative spatial audio representation specifically tailored for indoor scenes, enhancing the authenticity and immersion of audio experiences within virtual environments.

The studies described above identified building blocks of learning vision-to-audio generation and audio-visual representation, and upon these, in Chapter 10, I propose an audio-visual foundational framework that is based on and extends these building blocks to both generate diverse types of audio from general videos and also learn audio-visual representations for a multitude of tasks extending beyond generation alone.

Throughout this thesis, the collective contributions reflect a deeper representation of the audio-visual correspondence and novel approaches to the vision-to-audio generation domain. These contributions have marked significant strides toward higher precision, generic, and more interpretable audio-visual experiences.

KUN SU'S PUBLICATIONS

[KS1] Kun Su, Xiulong Liu, and Eli Shlizerman. Audeo: Audio generation for a silent performance video. *Advances in Neural Information Processing Systems*, 33:3325–3337, 2020.

[KS2] Kun Su, Xiulong Liu, and Eli Shlizerman. How does it sound? *Advances in Neural Information Processing Systems*, 34:29258–29273, 2021.

[KS3] Kun Su, Xiulong Liu, and Eli Shlizerman. Multi-instrumentalist net: Unsupervised generation of music from body movements. *arXiv preprint arXiv:2012.03478*, 2020.

[KS4] Kun Su, Mingfei Chen, and Eli Shlizerman. Inras: Implicit neural representation for audio scenes. *Advances in Neural Information Processing Systems*, 35:8144–8158, 2022.

[KS5] Kun Su, Judith Yue Li, Qingqing Huang, Dima Kuzmin, Joonseok Lee, Chris Donahue, Fei Sha, Aren Jansen, Yu Wang, Mauro Verzetti, et al. V2meow: Meowing to the visual beat via music generation. *arXiv preprint arXiv:2305.06594*, 2023.

[KS6] Kun Su, Xiulong Liu, and Eli Shlizerman. Predict & cluster: Unsupervised skeleton based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9631–9640, 2020.

[KS7] Kun Su, Kaizhi Qian, Eli Shlizerman, Antonio Torralba, and Chuang Gan. Physics-driven diffusion models for impact sound synthesis from videos. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[KS8] Kun Su and Eli Shlizerman. Clustering and recognition of spatiotemporal features through interpretable embedding of sequence to sequence recurrent neural networks. *Frontiers in artificial intelligence*, 3:70, 2020.

BIBLIOGRAPHY

[Re1] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013.

[Re2] Lawrence R Rabiner, Ronald W Schafer, et al. Introduction to digital speech processing. *Foundations and Trends® in Signal Processing*, 1(1–2):1–194, 2007.

[Re3] Robert A Moog. Midi: Musical instrument digital interface. *Journal of the Audio Engineering Society*, 34(5):394–404, 1986.

[Re4] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure. In *International Conference on Learning Representations*, 2018.

[Re5] Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1180–1188, 2020.

[Re6] Jon Gillick, Adam Roberts, Jesse Engel, Douglas Eck, and David Bamman. Learning to groove with inverse sequence transformations. In *International Conference on Machine Learning (ICML)*, 2019.

[Re7] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

[Re8] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2021.

[Re9] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

[Re10] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

[Re11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Re12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[Re13] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[Re14] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 203–213, 2020.

[Re15] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[Re16] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

[Re17] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[Re18] 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.

[Re19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[Re20] 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. *arXiv preprint arXiv:1612.07837*, 2016.

[Re21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Re22] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. *arXiv preprint arXiv:1902.08710*, 2019.

[Re23] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[Re24] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

[Re25] Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models. *arXiv preprint arXiv:2103.16091*, 2021.

[Re26] Junhyeok Lee and Seungu Han. Nu-wave: A diffusion probabilistic model for neural audio upsampling. *arXiv preprint arXiv:2104.02321*, 2021.

[Re27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[Re28] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[Re29] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[Re30] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47–1, 2022.

[Re31] Lilian Weng. What are diffusion models? *lilianweng.github.io*, Jul 2021.

[Re32] Kees Van Den Doel, Paul G Kry, and Dinesh K Pai. Foleyautomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the*

28th annual conference on Computer graphics and interactive techniques, pages 537–544, 2001.

[Re33] Zhimin Ren, Hengchin Yeh, and Ming C Lin. Example-guided physically based modal sound synthesis. *ACM Transactions on Graphics (TOG)*, 32(1):1–16, 2013.

[Re34] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2405–2413, 2016.

[Re35] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara L Berg. Visual to sound: Generating natural sound for videos in the wild. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[Re36] Kan Chen, Chuanxi Zhang, Chen Fang, Zhaowen Wang, Trung Bui, and Ram Nevatia. Visually indicated sound generation by perceptually optimized classification. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[Re37] Peihao Chen, Yang Zhang, Mingkui Tan, Hongdong Xiao, Deng Huang, and Chuang Gan. Generating visually aligned sound from videos. *IEEE Transactions on Image Processing*, 29:8292–8302, 2020.

[Re38] A Sophia Koepke, Olivia Wiles, Yael Moses, and Andrew Zisserman. Sight to sound: An end-to-end approach for visual piano transcription. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1838–1842. IEEE, 2020.

[Re39] Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba. Foley music: Learning to generate music from videos. *ECCV*, 2020.

[Re40] Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. Video background music generation with controllable music transformer. In *Proc. of the ACM International Conference on Multimedia*, 2021.

[Re41] Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation, 2021.

[Re42] Roy Sheffer and Yossi Adi. I hear your true colors: Image guided audio generation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[Re43] Xinhao Mei, Varun Nagaraja, Gael Le Lan, Zhaoheng Ni, Ernie Chang, Yangyang Shi, and Vikas Chandra. Foleygen: Visually-guided audio generation. *arXiv preprint arXiv:2309.10537*, 2023.

[Re44] Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *arXiv preprint arXiv:2306.17203*, 2023.

[Re45] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[Re46] Yelin Kim, Honglak Lee, and Emily Mower Provost. Deep learning for robust feature generation in audiovisual emotion recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 3687–3691. IEEE, 2013.

[Re47] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. *Advances in Neural Information Processing Systems*, 34:14200–14213, 2021.

[Re48] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. *Advances in neural information processing systems*, 29, 2016.

[Re49] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proceedings of the IEEE international conference on computer vision*, pages 609–617, 2017.

[Re50] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *Proceedings of the European conference on computer vision (ECCV)*, pages 435–451, 2018.

[Re51] Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Active contrastive learning of audio-visual video representations. *arXiv preprint arXiv:2009.09805*, 2020.

[Re52] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12486, 2021.

[Re53] Yuan Gong, Andrew Rouditchenko, Alexander H Liu, David Harwath, Leonid Karlinsky, Hilde Kuehne, and James Glass. Contrastive audio-visual masked autoencoder. *arXiv preprint arXiv:2210.07839*, 2022.

[Re54] Po-Yao Huang, Vasu Sharma, Hu Xu, Chaitanya Ryali, Haoqi Fan, Yanghao Li, Shang-Wen Li, Gargi Ghosh, Jitendra Malik, and Christoph Feichtenhofer. Mavil: Masked audio-video learners. *arXiv preprint arXiv:2212.08071*, 2022.

[Re55] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153*, 2017.

[Re56] Zheng Yan, Weiwei Liu, Shiping Wen, and Yin Yang. Multi-label image classification by feature attention network. *IEEE Access*, 7:98005–98013, 2019.

[Re57] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[Re58] 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, 2015.

[Re59] Bryan Wang and Yi-Hsuan Yang. Performancenet: Score-to-audio music generation with multi-band convolutional residual network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1174–1181, 2019.

[Re60] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[Re61] Mert Bay, Andreas F Ehmann, and J Stephen Downie. Evaluation of multiple-f0 estimation and tracking systems. In *ISMIR*, pages 315–320, 2009.

[Re62] James Blades. *Percussion instruments and their history*. Bold Strummer Limited, 1992.

[Re63] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455*, 2018.

[Re64] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[Re65] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

[Re66] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Counting out time: Class agnostic video repetition counting in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10387–10396, 2020.

[Re67] Florian Krebs, Sebastian Böck, and Gerhard Widmer. An efficient state-space model for joint tempo and meter tracking. In *ISMIR*, pages 72–78, 2015.

[Re68] Peter Grosche, Meinard Müller, and Frank Kurth. Cyclic tempogram—a mid-level tempo representation for musicsignals. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5522–5525. IEEE, 2010.

[Re69] Abe Davis and Maneesh Agrawala. Visual rhythm and beat. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2532–2535, 2018.

[Re70] Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

[Re71] Fabrizio Pedersoli and Masataka Goto. Dance beat tracking from visual information alone. *ISMIR*, 2020.

[Re72] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In *ISMIR*, pages 501–510, 2019.

[Re73] Colin Raffel. *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching.* PhD thesis, Columbia University, 2016.

[Re74] Matthew EP Davies, Norberto Degara, and Mark D Plumbley. Evaluation methods for musical audio beat tracking algorithms. *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009.

[Re75] Eitan Richardson and Yair Weiss. On gans and gmms. In *Advances in Neural Information Processing Systems*, pages 5847–5858, 2018.

[Re76] Li-Chia Yang and Alexander Lerch. On the evaluation of generative models in music. *Neural Computing and Applications*, 32(9):4773–4784, 2020.

[Re77] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matthew Sharifi, Neil Zeghidour, and C. Frank. Musiclm: Generating music from text. *ArXiv*, abs/2301.11325, 2023.

[Re78] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. AI Choreographer: Music Conditioned 3D Dance Generation with AIST++. *arXiv:2101.08779*, 2021.

[Re79] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. Audiolm: a language modeling approach to audio generation. *arXiv preprint arXiv:2209.03143*, 2022.

[Re80] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE, 2021.

[Re81] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[Re82] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[Re83] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel PW Ellis. MuLan: A joint embedding of music audio and natural language. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

[Re84] Dídac Surís, Carl Vondrick, Bryan Russell, and Justin Salamon. It's time for artistic correspondence in music and video. 2022.

[Re85] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[Re86] Jort F Gemmeke, Daniel PW 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 *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017.

[Re87] Mubert-Inc. Mubert. https://mubert. com/, https://github.com/mubertai/ mubert-text-to-music. 2022.

[Re88] Seth* Forsgren and Hayk* Martiros. Riffusion - Stable diffusion for real-time music generation. 2022.

[Re89] Ye Zhu, Kyle Olszewski, Yu Wu, Panos Achlioptas, Menglei Chai, Yan Yan, and Sergey Tulyakov. Quantized GAN for complex music generation from dance videos. *arXiv:2204.00604*, 2022.

[Re90] Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. Discrete contrastive diffusion for cross-modal and conditional generation. *arXiv:2206.07771*, 2022.

[Re91] Joel Shor, Aren Jansen, Ronnie Maor, Oran Lang, Omry Tuval, Felix de Chaumont Quitry, Marco Tagliasacchi, Ira Shavitt, Dotan Emanuel, and Yinnon Haviv. Towards learning a universal non-semantic representation of speech. *arXiv:2002.12764*, 2020.

[Re92] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, et al. CNN architectures for large-scale audio classification. 2017.

[Re93] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *arXiv:2207.09983*, 2022.

[Re94] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. *arXiv:2209.15352*, 2022.

[Re95] Neil Zeghidour, Olivier Teboul, Félix de Chaumont Quitry, and Marco Tagliasacchi. Leaf: A learnable frontend for audio classification. *arXiv preprint arXiv:2101.08596*, 2021.

[Re96] Jiashuo Yu, Yaohui Wang, Xinyuan Chen, Xiao Sun, and Yu Qiao. Long-term rhythmic video soundtracker. *arXiv:2305.01319*, 2023.

[Re97] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2649–2656, 2014.

[Re98] Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

[Re99] Hossein Rahmani, Arif Mahmood, Du Q Huynh, and Ajmal Mian. Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition. In *European conference on computer vision*, pages 742–757. Springer, 2014.

[Re100] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[Re101] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Learning actionlet ensemble for 3d human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):914–927, 2013.

[Re102] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[Re103] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive neural networks for high performance skeleton-based human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[Re104] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1227–1236, 2019.

[Re105] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2203–2212, 2017.

[Re106] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan Kankanhalli. Unsupervised learning of view-invariant action representations. In *Advances in Neural Information Processing Systems*, pages 1254–1264, 2018.

[Re107] Nenggan Zheng, Jun Wen, Risheng Liu, Liangqu Long, Jianhua Dai, and Zhefeng Gong. Unsupervised representation learning with long-term dynamics for skeleton based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Re108] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. View invariant human action recognition using histograms of 3d joints. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–27. IEEE, 2012.

[Re109] Lei Wang, Du Q Huynh, and Piotr Koniusz. A comparative review of recent kinect-based action recognition algorithms. *arXiv preprint arXiv:1906.09955*, 2019.

[Re110] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014.

[Re111] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016.

[Re112] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[Re113] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[Re114] Bochen Li, Xinzhao Liu, Karthik Dinesh, Zhiyao Duan, and Gaurav Sharma. Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications. *IEEE Transactions on Multimedia*, 21(2):522–535, 2018.

[Re115] Juan F Montesinos, Olga Slizovskaia, and Gloria Haro. Solos: A dataset for audio-visual music analysis. *arXiv preprint arXiv:2006.07931*, 2020.

[Re116] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE, 2020.

[Re117] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[Re118] Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. Diff-tts: A denoising diffusion model for text-to-speech. *arXiv preprint arXiv:2104.01409*, 2021.

[Re119] Shiguang Liu and Dinesh Manocha. Sound synthesis, propagation, and rendering: a survey. *arXiv preprint arXiv:2011.05538*, 2020.

[Re120] Tor Erik Vigran. *Building acoustics*. CRC Press, 2014.

[Re121] Guy-Bart Stan, Jean-Jacques Embrechts, and Dominique Archambeau. Comparison of different impulse response measurement techniques. *Journal of the Audio engineering society*, 50(4):249–262, 2002.

[Re122] 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.

[Re123] Jeffrey Borish. Extension of the image model to arbitrary polyhedra. *The Journal of the Acoustical Society of America*, 75(6):1827–1836, 1984.

[Re124] Asbjørn Krokstad, Staffan Strom, and Svein Sørsdal. Calculating the acoustical room response by the use of a ray tracing technique. *Journal of Sound and Vibration*, 8(1):118–125, 1968.

[Re125] Michael Vorländer. Simulation of the transient and steady-state sound propagation in rooms using a new combined ray-tracing/image-source algorithm. *The Journal of the Acoustical Society of America*, 86(1):172–178, 1989.

[Re126] Nikunj Raghuvanshi and John Snyder. Parametric wave field coding for precomputed sound propagation. *ACM Transactions on Graphics (TOG)*, 33(4):1–11, 2014.

[Re127] Nikunj Raghuvanshi and John Snyder. Parametric directional coding for precomputed sound propagation. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[Re128] Lakulish Antani, Anish Chandak, Micah Taylor, and Dinesh Manocha. Direct-to-indirect acoustic radiance transfer. *IEEE Transactions on Visualization and Computer Graphics*, 18(2):261–269, 2011.

[Re129] Lakulish Antani, Anish Chandak, Lauri Savioja, and Dinesh Manocha. Interactive sound propagation using compact acoustic transfer operators. *ACM Transactions on Graphics (TOG)*, 31(1):1–12, 2012.

[Re130] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *European Conference on Computer Vision*, pages 17–36. Springer, 2020.

[Re131] Andrew Luo, Yilun Du, Michael J Tarr, Joshua B Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. *arXiv preprint arXiv:*, 2021.

[Re132] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

[Re133] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[Re134] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[Re135] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. *arXiv preprint arXiv:2306.05284*, 2023.

[Re136] 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*, 2020.

[Re137] Hugo Flores Garcia, Prem Seetharaman, Rithesh Kumar, and Bryan Pardo. Vampnet: Music generation via masked acoustic token modeling. *arXiv preprint arXiv:2307.04686*, 2023.

[Re138] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Re139] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[Re140] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *arXiv preprint arXiv:2306.06546*, 2023.

[Re141] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35:32897–32912, 2022.

[Re142] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.

[Re143] Tianhong Li, Huiwen Chang, Shlok Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. Mage: Masked generative encoder to unify representation learning and image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2142–2152, 2023.

[Re144] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[Re145] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset, 2020.

[Re146] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[Re147] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023.

[Re148] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\'echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

[Re149] Khaled Koutini, Jan Schlüter, Hamid Eghbal-Zadeh, and Gerhard Widmer. Efficient training of audio transformers with patchout. *arXiv preprint arXiv:2110.05069*, 2021.

[Re150] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[Re151] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12695–12705, 2020.

[Re152] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

[Re153] Haytham M Fayek and Anurag Kumar. Large scale audiovisual learning of sounds with weakly labeled data. *arXiv preprint arXiv:2006.01595*, 2020.

[Re154] Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018, 2015.

[Re155] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[Re156] Yuan Gong, Yu-An Chung, and James Glass. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*, 2021.

[Re157] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10699–10709, 2022.

[Re158] Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. *Advances in Neural Information Processing Systems*, 35:28708–28720, 2022.