

Physical and Computational Approaches Towards More Scalable Molecular Systems

Ashley Paige Stephenson

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

Doctor of Philosophy

University of Washington
2026

Reading Committee:
Jeffrey Nivala, Chair
Luis H. Ceze
Sheng Wang

Program Authorized to Offer Degree:
Paul G. Allen School of Computer Science and Engineering

© Copyright 2026

Ashley Paige Stephenson

University of Washington

Abstract

Physical and Computational Approaches Towards More Scalable Molecular Systems

Ashley Paige Stephenson

Chair of the Supervisory Committee:

Jeffrey Nivala

Paul G. Allen School of Computer Science and Engineering

The ability to understand and engineer biological systems depends on efficiently manipulating and interpreting molecular information. Design-Build-Test-Learn (DBTL) cycles remain limited by bottlenecks in both experimental workflows and computational analysis. This thesis addresses these complementary challenges through both physical laboratory automation and machine learning methods for interpreting nanopore sequencing signals from chemically diverse biopolymers.

First, I present work on digital microfluidic automation using the PurpleDrop platform, enabling programmable droplet manipulation for applications including DNA data storage and aptamer discovery. This work demonstrates the potential of integrated hardware-software systems for automated experimentation while highlighting challenges in reliability and system integration.

The primary focus of this thesis develops structure-informed machine learning models for nanopore sequencing of expanded molecular alphabets. Xenonucleic acids (XNAs), synthetic nucleotides with modified bases, sugars, or backbones, hold promise for therapeutics, diagnostics, and information storage but lack robust sequencing methods. Existing nanopore approaches require extensive experimental calibration across many sequence contexts. I show that models incorporating molecular structure and chemical features can predict nanopore ionic current signatures for unseen XNA bases, substantially reducing experimental requirements. Structure-informed models outperform sequence-only approaches in few-shot and zero-shot settings and enable practical XNA sequencing with limited calibration data.

Finally, I extend this framework to nanopore protein sequencing for detection of post-translational modifications such as phosphorylation. Despite the greater complexity of proteins, similar modeling approaches enable phosphorylation detection, suggesting broader applicability of structure-based methods for complex biopolymer sequencing.

Together, these contributions reduce experimental burden through both physical automation and computational prediction, enabling more scalable DBTL cycles for increasingly complex molecular systems.

Acknowledgements

This work was made possible by the support of many people and communities.

I am deeply grateful to my advisors, Luis Ceze, Karin Strauss, and Jeff Nivala. Luis and Karin gave me my first job after college as a research engineer in the Molecular Information Systems Lab (MISL) and later became my advisors when I joined the PhD program. They built a lab environment that valued collaboration and creativity, and I learned an enormous amount from both of them about research, leadership, and navigating academia. Karin also helped me secure a summer internship at Microsoft Research, where I had the opportunity to work with her directly. Their kindness, encouragement and belief in me have had an immeasurable impact on my life. Thank you for making MISL feel like home!

Jeff became my advisor after my qualifying exam. His expertise in synthetic biology opened up new research directions for me, and his mentorship style gave me the space to explore ideas while providing guidance when I needed it. I consistently left our meetings with clearer direction and renewed enthusiasm for the work. I am immensely grateful for his support, whether it was guiding me through research, supporting me through tough personal times, or motivating me to keep going through challenges. Jeff gave me the confidence to believe in myself and none of this would have been possible without him.

I would also like to thank my committee members, Jorge Marchand and Sheng Wang, for their time and thoughtful feedback throughout this process.

The PurpleDrop project was my introduction to research in the lab. Bichlien Nguyen welcomed me into the group and became both a collaborator and friend, offering invaluable research and career guidance. Chris Takahashi taught me about electronics, circuit design, and engineering for synthetic biology, and was always generous with his time and expertise. Max Willsey worked on the software side of the platform and helped me develop skills in computer science areas where I had less background. Jeff McBride joined

later to help revamp the platform and was an excellent engineer to work with. I learned a great deal from everyone on the PurpleDrop team.

For the XNA project, I collaborated with Jorge Marchand's lab. Jayson Sumabat and Hinako Kawabe generated the data that enabled all of my XNA modeling work, which formed a major component of this dissertation. I learned a great deal from Jorge, Jayson, and Hinako, and working with them on our publication was a highlight of my PhD. Sidharth Lakshmanan also contributed excellent work to a portion of the project.

For the protein sequencing project, Daphne Kontogiorgos-Heintz provided the data I used and was always willing to meet and discuss research. I am grateful for her time and help, as well as the MISL protein sequencing team.

Finally, I would like to thank my family and friends for their support throughout my PhD. I am especially grateful to my dad, Phil Stephenson, and my brother, Miles Stephenson for their constant support, love and encouragement. I could not have made it through this journey without them.

DEDICATION

Dedicated in loving memory to my mother and best friend, Barbara Stephenson.

Contents

1	Introduction	21
1.1	Motivation: Scalable Manipulation and Readout of Molecular Information	21
1.2	Design-Build-Test-Learn Cycles in Synthetic Biology	22
1.3	Physical Automation: Laboratory Robotics and Digital Microfluidics	24
1.4	Computational Methods: Structure-Informed Models for Nanopore Sequencing	25
2	Physical Automation and the PurpleDrop Digital Microfluidic Platform	29
2.1	The Automation Landscape: Positioning Digital Microfluidics	29
2.2	The PurpleDrop Platform	32
2.2.1	Design Philosophy	32
2.2.2	Hardware Architecture	33
2.2.3	Peripheral Systems	34
2.2.4	Software Control and Error Correction	36
2.3	Application Case Study: DNA Data Storage	37
2.4	Application Case Study: Aptamer Discovery via SELEX	39
2.5	Critical Assessment: Limitations and Lessons	42
2.6	Conclusions	44
3	Background: Nanopore Sequencing	47
3.1	Introduction to Nanopore Sequencing Technology	47
3.2	From Signal to Sequence: Basecalling and the Role of k-mer Models	48
3.3	Detection of Modified Bases: Challenges and Opportunities	49

3.4	Why Modified Base Detection Is Difficult	49
3.5	Current Approaches to Modified Base Detection	50
3.6	The Need for More Scalable Computational Methods	51
4	Structure-informed Models for Ionic Current Prediction in Nanopore Sequencing of Expanded DNA Alphabets	53
4.1	Introduction	54
4.2	Materials and Methods	56
4.2.1	Sequence-based CNN model	56
4.2.2	Graph Neural Network Model	57
4.2.3	Fine-tuned GPT-2 model with k-mer structures represented as SMILES strings . . .	60
4.3	Results	61
4.3.1	Comparing sequence- and structure-based architectures for k-mer-level generaliza- tion and DNA 5mC prediction	61
4.3.2	Generating XNA k-mer tables from machine learning models	64
4.3.3	Integrating model-generated k-mer tables in an XNA basecalling model	68
4.4	Discussion	71
5	Extending Structure-Informed Models to Nanopore Protein Sequencing	75
5.1	Introduction	75
5.2	Nanopore Protein Sequencing: Challenges and Methods	76
5.2.1	Obstacles in Protein Translocation	76
5.2.2	ClpX-Mediated Translocation	77
5.2.3	The PASTOR Dataset	77
5.3	Model Architectures and Training	78
5.3.1	Dataset Composition	78
5.3.2	Model Architectures	78
5.3.3	Pre-training Strategy	81
5.3.4	Fine-tuning and Evaluation Protocol	81

5.4	Results: Phosphorylation Detection and Model Generalization	82
5.4.1	Zero-Shot and Few-Shot Performance	82
5.4.2	Interpreting Model Behavior	83
5.4.3	Proteoform Squiggle Generation and DTW Analysis	84
5.4.4	Sources of Prediction Error	87
5.5	Discussion	88
5.5.1	Implications for Protein Nanopore Sequencing	88
5.5.2	Comparison with XNA Modeling	89
5.5.3	Limitations and Future Directions	90
5.5.4	Broader Context: Toward Proteoform Sequencing	91
6	Conclusion	93
6.1	Summary of Contributions	93
6.1.1	Physical Automation: The PurpleDrop Platform	93
6.1.2	Computational Methods: Structure-Informed Nanopore Signal Prediction	94
6.2	Broader Implications and Future Directions	94
6.3	Closing Perspective	95
A	Appendix One	107
A.1	Appendix section 1	107

List of Figures

1.1	Enabling technologies in synthetic biology. The first two decades of synthetic biology have resulted in a number of experimental methods, technologies, standards and commercial products and services that have contributed to the rise of Design-Build-Test-Learn cycles and automation in synthetic biology. Utilizing these hardware, software and experimental tools and connecting them in standardized formats is expected to accelerate scientific discovery via closed-loop, iterative experimentation. Examples are grouped loosely around the portion(s) of the DBTL cycle to which they contribute.	23
2.1	Levels of automation in synthetic biology laboratories achieved through hardware and software automation technologies.	30
2.2	PurpleDrop Platform.	32
2.3	Bead wash on PurpleDrop. A below-board motorized magnetic arm positions a magnet underneath a droplet containing magnetic beads in the first frame. The droplet is moved away via electrowetting while the magnet is held in place. The droplet on the left in the final frame contains the constituents of the initial droplet that did not bind to the magnetic beads upon mixing and incubation. Resuspension and elution of the pellet on the right yields the bound products.	35
2.4	PurpleDrop PCR automation.	36

2.5	Storing physically isolated spots of dehydrated DNA on glass cartridges enables high information density. a Spotted cartridges can be further organized into decks. b To retrieve data, the desired cartridge is first loaded onto the DMF device consisting of an electrode grid used to perform retrieval functions. An actual magnified spot is depicted, where the scale bar in the inset is 275 μm . c Next, a water droplet sandwiched between the cartridge and electrodes is actuated to move under the spotted DNA for rehydration. Depicted is an actual droplet moving on the device. After retrieval, we manually recover the sample and analyze it to assess DNA recovery rate and contamination across samples sharing paths on the DMF device.	38
2.6	SELEX translated into droplets. A. Prepare target-functionalized magnetic beads and initial single stranded random DNA library. B. Mix and incubate target-bead and library droplets. C. Perform magnetic bead wash and discard unbound strands to waste. D. Combine selection product droplet with PCR master mix and transfer to PDMS chamber for amplification. E. Perform magnetic bead based strand digestion of dsDNA PCR product via biotinylation of one strand of each helix during PCR with biotinylated reverse primer. F. Output aliquot of SELEX product to nanopore for sequencing; repeat B through F until sequence convergence.	41
4.1	Overview of structure-informed XNA nanopore sequencing workflow.	54
4.2	Structures of canonical DNA nucleobases and xenonucleobases.	64
4.3	Overview of model architectures.	66
4.4	Comparison of model performance on <i>de novo</i> XNA prediction.	67
4.5	Basecalling	69
5.2	Model comparison for phosphorylation detection and generalization. Left: average R^2 values for CNN and GIN models over phosphorylation train-test splits. Right: average RMSE values for CNN and GIN models over phosphorylation train-test splits. In both cases, train split refers to the percentage of the phosphorylated dataset that was added to the canonical training data during model fine-tuning. R^2 and RMSE values correspond to model performance in predicting the held out phosphorylated dataset fraction.	83

5.3	Top N DTW Rankings for CNN and GIN proteoform prediction. Left: Top N results for zero-shot CNN and GIN model generated signals compared to all experimental traces. Right: Top N results for zero-shot CNN and GIN model generated signals compared to a subset of all experimental traces filtered to prevent over-representation of canonical signal data over phosphorylation containing signals. The identification rate shows the percentage of cases where the true corresponding experimental signal was successfully aligned to the predicted signal within the top N lowest DTW rankings.	86
A.1	Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (B) dataset was added to the model training data. Solid lines represent <i>de novo</i> prediction cases where no examples of the held out XNA base were included in the training data.	108
A.2	Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (X^t) dataset was added to the model training data. Solid lines represent <i>de novo</i> prediction cases where no examples of the held out XNA base were included in the training data.	109
A.3	Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (P) dataset was added to the model training data. Solid lines represent <i>de novo</i> prediction cases where no examples of the held out XNA base were included in the training data.	110
A.4	Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (K^n) dataset was added to the model training data. Solid lines represent <i>de novo</i> prediction cases where no examples of the held out XNA base were included in the training data.	111

A.5	Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (V) dataset was added to the model training data. Solid lines represent <i>de novo</i> prediction cases where no examples of the held out XNA base were included in the training data.	112
A.6	Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (S^c) dataset was added to the model training data. Solid lines represent <i>de novo</i> prediction cases where no examples of the held out XNA base were included in the training data.	113
A.7	Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (S^n) dataset was added to the model training data. Solid lines represent <i>de novo</i> prediction cases where no examples of the held out XNA base were included in the training data.	114
A.8	Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (Z^a) dataset was added to the model training data. Solid lines represent <i>de novo</i> prediction cases where no examples of the held out XNA base were included in the training data.	115
A.9	Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (Z^n) dataset was added to the model training data. Solid lines represent <i>de novo</i> prediction cases where no examples of the held out XNA base were included in the training data.	116

A.10 Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (J) dataset was added to the model training data. Solid lines represent <i>de novo</i> prediction cases where no examples of the held out XNA base were included in the training data.	117
A.11 Example scatter plots for canonical (ATGC) dropout experiments. Models were trained with all XNA datasets and tested on the canonical 4-mer set. For 5% cases, 5% of the held out canonical set was added to the training set.	118
A.12 Example scatter plots for XNA base B dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.	119
A.13 Example scatter plots for XNA base J dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.	120
A.14 Example scatter plots for XNA base X^t dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.	121
A.15 Example scatter plots for XNA base P dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.	122
A.16 Example scatter plots for XNA base K^n dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.	123
A.17 Example scatter plots for XNA base V dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.	124

A.18 Example scatter plots for XNA base S^c dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.	125
A.19 Example scatter plots for XNA base S^n dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.	126
A.20 Example scatter plots for XNA base Z^a dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.	127
A.21 Example scatter plots for XNA base Z^n dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.	128
A.22 Example plots overlaying model-predicted k-mer mean current levels (blue) on measured current levels (orange) for representative heptamers/bases. Error bars indicate the standard deviation of the measured normalized signal level. Model-predicted levels shown here were generated by the GIN architecture trained under few-shot conditions (i.e. trained with 5% of each held-out XNA dataset added to the training data.)	129
A.23 Example plots overlaying model-predicted k-mer mean current levels (blue) on measured current levels (orange) for representative heptamers/bases. Error bars indicate the standard deviation of the measured normalized signal level. Model-predicted levels shown here were generated by the GPT-2 architecture trained under few-shot conditions (i.e. trained with 5% of each held-out XNA dataset added to the training data.)	129
A.24 Example plots overlaying model-predicted k-mer mean current levels (blue) on measured current levels (orange) for representative heptamers/bases. Error bars indicate the standard deviation of the measured normalized signal level. Model-predicted levels shown here were generated by the CNN architecture trained under few-shot conditions (i.e. trained with 5% of each held-out XNA dataset added to the training data.)	130

List of Tables

4.1	SMILES strings for individual nucleotides	58
4.2	Atom level features	59
5.1	SMILES strings for amino acids and phosphoserine	79
A.1	Average RMSE and Pearson Correlation values for CNN model 50-fold cross validation on canonical ONT 6mer. Run 1 and 2 are independent replicates with 50 repeats per train-test split size.	107
A.2	Average RMSE and Pearson Correlation values for CNN model 50-fold cross validation on 5mC-containing 6mer dataset. Run 1 and 2 are independent replicates with 50 repeats. Models were trained using all canonical 6mer data plus varying percentages of the 5mC-containing dataset, then evaluated on the remaining portion of the 5mC set.	108

Chapter 1

Introduction

1.1 Motivation: Scalable Manipulation and Readout of Molecular Information

The ability to engineer biological systems with designed functions has transformed fields ranging from medicine to materials science. From developing novel therapeutics and diagnostic tools to creating sustainable biomanufacturing processes and storing digital information in DNA, synthetic biology offers solutions to challenges spanning medicine, energy, materials, and computing[1, 2]. Central to this field is the ability to manipulate molecular information to design, construct, and characterize biomolecules with desired properties and functions.

Yet our capacity to engineer increasingly sophisticated molecular systems remains fundamentally limited by experimental throughput. Creating a novel enzyme, therapeutic antibody, or synthetic genetic circuit requires iterating through many design variants, each requiring synthesis, testing, and analysis before refinement. This iterative process, formalized as the Design-Build-Test-Learn (DBTL) cycle, represents the core workflow of modern synthetic biology[3, 4, 5]. However, traditional laboratory methods create severe bottlenecks at multiple stages: physically manipulating samples is labor-intensive and error-prone; synthesizing and assembling molecular constructs is slow and expensive; characterizing molecular properties requires specialized equipment and expertise; and interpreting complex experimental readouts demands substantial analytical effort.

These bottlenecks become increasingly acute as synthetic biology moves toward more ambitious goals. Engineering organisms with expanded genetic codes, developing protein therapeutics with precise post-translational modifications, or evolving aptamers from vast molecular libraries all require screening thousands to millions of variants. The experimental burden grows not only with the number of variants but also with the chemical complexity of the molecules being studied. Moving beyond the canonical building blocks of biology (the four DNA bases, twenty amino acids, standard post-translational modifications) into the realm of synthetic and chemically modified biopolymers introduces new challenges in both manipulation and measurement.

This thesis addresses these challenges through two complementary approaches: physical automation of experimental workflows and computational methods for interpreting molecular sensors. While these may initially appear as distinct research directions, they share a common goal: reducing the experimental burden of DBTL cycles to enable more rapid, scalable, and cost-effective molecular engineering. Physical automation addresses the "Build" and "Test" stages by enabling high-throughput, reproducible sample manipulation and measurement. Computational methods address the "Test" and "Learn" stages by extracting maximum information from experimental readouts and reducing the calibration requirements for new molecular systems. Together, these approaches work synergistically to accelerate the pace of synthetic biology research.

1.2 Design-Build-Test-Learn Cycles in Synthetic Biology

The DBTL framework provides a conceptual structure for understanding and optimizing the synthetic biology workflow (Figure 1.1). In the Design phase, researchers use existing knowledge from mechanistic models, prior experimental data, or machine learning predictions to propose molecular variants likely to exhibit desired properties. The Build phase involves physically constructing these designs, whether through DNA synthesis, protein expression, or assembly of more complex genetic circuits. The Test phase characterizes the resulting constructs, measuring relevant properties such as binding affinity, catalytic activity, or gene expression levels. Finally, the Learn phase analyzes these results to update models, refine design principles, and inform the next iteration.

The efficiency of DBTL cycles depends critically on throughput, accuracy, and cost at each stage. Advances in DNA synthesis have dramatically reduced the cost and time required to build genetic constructs,

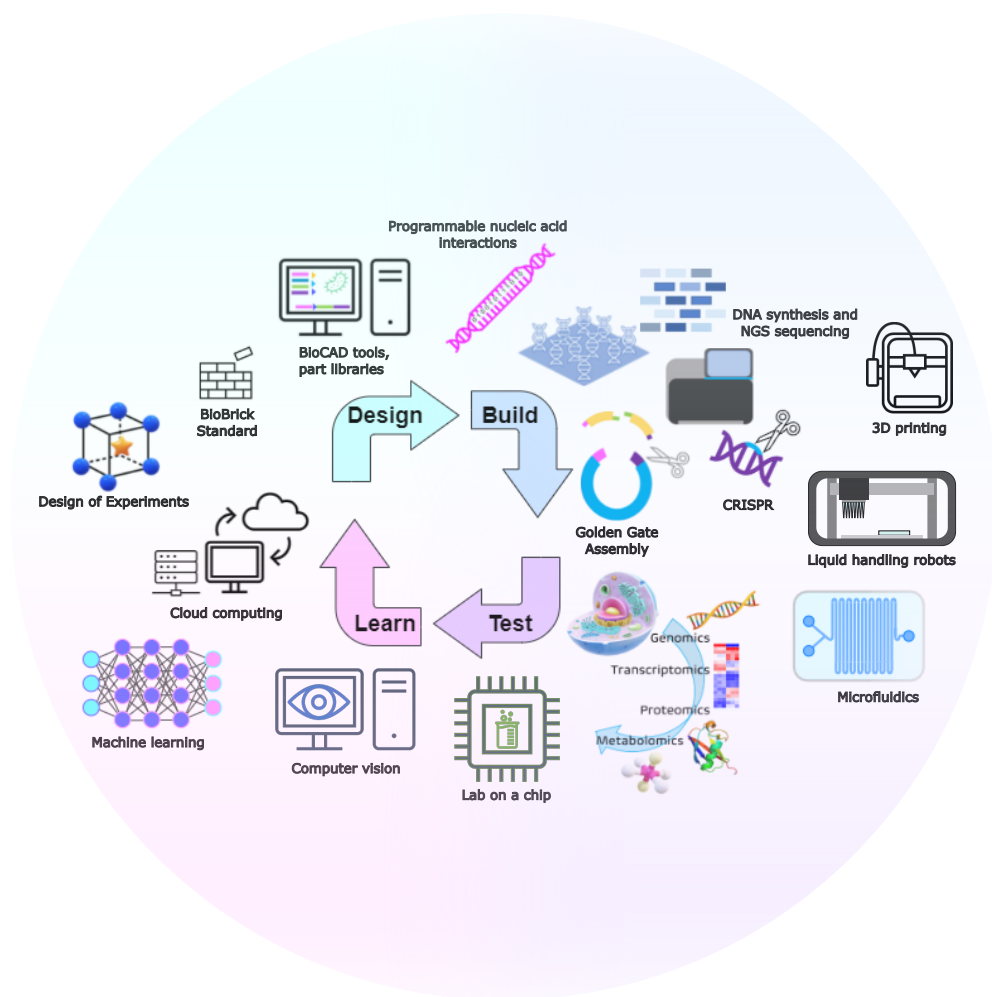


Figure 1.1: Enabling technologies in synthetic biology. The first two decades of synthetic biology have resulted in a number of experimental methods, technologies, standards and commercial products and services that have contributed to the rise of Design-Build-Test-Learn cycles and automation in synthetic biology. Utilizing these hardware, software and experimental tools and connecting them in standardized formats is expected to accelerate scientific discovery via closed-loop, iterative experimentation. Examples are grouped loosely around the portion(s) of the DBTL cycle to which they contribute.

while high-throughput screening technologies have increased testing capacity. However, significant bottlenecks remain, particularly in sample manipulation, measurement of complex molecular properties, and interpretation of high-dimensional experimental data.

Physical automation has long been recognized as essential for scaling DBTL cycles. Liquid-handling robots, automated colony pickers, and plate readers have become standard in many synthetic biology laboratories, enabling experiments that would be impractical or impossible manually. Yet these systems often

remain expensive, inflexible, and require substantial infrastructure. Moreover, they typically address individual unit operations rather than providing integrated platforms for complete workflows. There is a growing need for modular, reconfigurable automation systems that can be easily adapted to diverse experimental protocols and integrated into closed-loop DBTL cycles where computational design directly drives automated experimentation.

Equally important is the role of computational methods in reducing experimental burden. When characterizing new molecular systems, particularly those incorporating non-standard chemical modifications, researchers typically face a calibration problem: before measurements can be meaningfully interpreted, the system must be extensively calibrated using molecules with known properties. For example, developing a biosensor for a new small molecule requires measuring responses to many concentrations and chemical variants to establish a calibration curve. Similarly, sequencing DNA containing modified bases requires measuring characteristic signals from each modification in diverse sequence contexts. This calibration burden grows rapidly with chemical complexity, often becoming the rate-limiting step in applying new measurement technologies to novel molecular systems.

Computational models that can predict molecular properties or measurement readouts from first principles or from limited training data offer a powerful alternative. If models can learn the underlying relationships between molecular structure and experimental observables, they can potentially generalize to new molecules with minimal additional calibration. This is particularly valuable for high-dimensional measurement modalities like mass spectrometry or nanopore sequencing, where the relationship between molecular structure and signal is complex but governed by physical principles that models might capture.

1.3 Physical Automation: Laboratory Robotics and Digital Microfluidics

The first component of this thesis addresses physical automation through digital microfluidics, specifically using the PurpleDrop platform[6]. Digital microfluidics (DMF) enables programmable manipulation of discrete droplets on an array of electrodes, offering a flexible, reconfigurable alternative to traditional channel-based microfluidics or robotic liquid handlers. By applying voltage patterns to electrodes, individual droplets can be moved, merged, split, and mixed, enabling complex liquid-handling protocols on a compact device.

Chapter 2 presents work developing and applying the PurpleDrop system for synthetic biology applica-

tions. I describe the platform’s design, key features, and optimization for biological workflows, using DNA data storage and aptamer discovery as case studies. These applications demonstrate both the potential and limitations of current DMF technology. On one hand, the system enables automated, programmable workflows with minimal manual intervention and small sample volumes. On the other hand, challenges including electrode reliability, biofouling, and integration with off-board instrumentation remain significant barriers to widespread adoption.

This work provides important context for understanding the current state of laboratory automation and the path forward. While not the primary focus of the thesis, it illustrates the practical challenges of implementing physical automation and motivates the complementary approach of reducing experimental requirements through computational methods. Moreover, it highlights opportunities for future integration: as automation platforms mature and become more reliable, coupling them with predictive computational models could enable truly autonomous DBTL cycles where design, execution, and analysis proceed with minimal human intervention.

1.4 Computational Methods: Structure-Informed Models for Nanopore Sequencing

The primary contribution of this thesis is the development of molecular structure-informed machine learning models for interpreting nanopore sequencing signals from chemically diverse biopolymers. Nanopore sequencing has emerged as a powerful technology for directly reading molecular information at the single-molecule level[7]. Unlike traditional sequencing methods that rely on enzymatic incorporation of labeled nucleotides, nanopore sequencing measures ionic current as molecules pass through a protein nanopore. This direct physical readout offers unique advantages: it requires no amplification or labeling, can sequence long molecules processively, and has the inherent capacity to detect chemical modifications that alter the current signature.

However, realizing this potential for non-canonical molecules, particularly synthetic nucleotides (XNAs) and modified proteins, requires solving a fundamental calibration problem. The ionic current depends on the molecular structure occupying the nanopore’s sensing region, typically spanning 4-9 nucleotides for

DNA or around twenty amino acids for proteins. To decode these signals, existing approaches require empirically measuring characteristic currents for each molecule in every relevant sequence context. For DNA with four canonical bases, this is manageable: a 6-nucleotide context requires measuring $4^6 = 4,096$ possible sequences. But for expanded alphabets or modified bases, the experimental burden grows rapidly. A single modified base in 6-nucleotide contexts requires an additional 1,024 measurements; ten different modifications require over 10,000 measurements; and longer contexts or larger alphabets quickly become prohibitive.

This calibration bottleneck severely limits the application of nanopore sequencing to synthetic and modified biopolymers. Xenonucleic acids, nucleotides with modified bases, sugars, or backbones, offer transformative potential for applications including therapeutics (aptamers with enhanced stability), diagnostics (expanded-alphabet barcoding), information storage (higher bit density), and engineered organisms (expanded genetic codes)[8, 9, 10, 11, 12]. Similarly, detecting post-translational modifications in proteins could revolutionize proteomics, enabling direct readout of phosphorylation, acetylation, and other modifications that regulate cellular function[13, 14]. Yet the experimental effort required to calibrate nanopore sequencing for each new chemistry remains a major barrier.

Chapters 3-5 address this challenge by developing computational models that predict nanopore signals from molecular structure and chemical properties. Rather than requiring exhaustive empirical measurement of every possible sequence context, these models learn relationships between molecular features such as atom connectivity, functional groups, charge distributions, steric properties, and the resulting ionic currents. If successful, such models could enable sequencing of new chemistries with minimal calibration: a few measurements to anchor the model, rather than thousands to exhaustively characterize every context.

Chapter 3 provides background on nanopore sequencing technology, explaining how signals are generated, how basecalling converts signals to sequences, and why detecting modified bases is particularly challenging. This chapter establishes the technical foundation and motivates the need for computational approaches that can generalize beyond measured calibration data.

Chapter 4 presents the core methodological contribution, structure-informed machine learning models for predicting nanopore signals from XNA-containing DNA. I compare three approaches: a sequence-only convolutional neural network (CNN) serving as a baseline; a graph isomorphism network (GIN) that rep-

resents molecules as graphs with atoms as nodes and chemical bonds as edges; and a fine-tuned GPT-2 transformer model that processes SMILES string representations of molecular structure. Using a dataset of ten different XNA bases, each measured in 256 sequence contexts (4-nucleotide k-mers), I evaluate each model’s ability to predict signals for held-out XNA bases it has never seen during training (zero-shot learning) and with limited exposure to the target base (few-shot learning).

The results demonstrate that while sequence context captures much of the signal variability, incorporating explicit structural and chemical information improves performance, particularly for zero-shot generalization. The GIN model, which combines graph-based molecular representations with sequence context, achieves the best overall performance. Critically, I show that these model-generated predictions can be integrated into practical basecalling pipelines, where even predictions from zero-shot models enable XNA detection substantially above chance, and few-shot models (trained on just 5% of the target XNA’s data) approach the performance of fully empirical models. This demonstrates the practical utility of structure-informed prediction for reducing experimental calibration requirements.

Chapter 5 extends this framework to nanopore protein sequencing, focusing on detection of phosphorylation, a key post-translational modification. Proteins present additional challenges compared to nucleic acids including neutral backbones that complicate electrophoretic translocation, twenty chemically diverse amino acids rather than four similar nucleotides, and larger effective sensing regions. Despite these challenges, I demonstrate that similar structure-informed modeling approaches can predict signals for phosphorylated peptides and enable modification detection. This suggests that the framework developed for XNA sequencing may generalize broadly to other complex biopolymer sequencing problems.

Together, these contributions advance the scalability of molecular engineering by addressing bottlenecks in both physical manipulation and computational interpretation of molecular information. While the automation and modeling components represent distinct technical approaches, they share a common philosophy: reducing experimental burden to enable more efficient, higher-throughput DBTL cycles. As synthetic biology continues to explore increasingly complex and chemically diverse molecular systems, from expanded genetic alphabets to engineered proteins with novel modifications, such approaches will become increasingly essential.

Chapter 2

Physical Automation and the PurpleDrop Digital Microfluidic Platform

Early in my PhD, I explored physical automation as a strategy for reducing experimental burden in molecular engineering. This work resulted in the PurpleDrop platform and demonstrated its application to DNA data storage and aptamer discovery. While this work successfully showed that low-cost digital microfluidics could automate complex molecular workflows, it also revealed fundamental challenges in reliability and integration that motivated the computational approaches described in subsequent chapters. This chapter describes the PurpleDrop platform, its capabilities, and the lessons learned about scalability challenges in molecular engineering. Portions of this chapter have been adapted from [4, 6, 15, 16].

2.1 The Automation Landscape: Positioning Digital Microfluidics

Laboratory automation exists on a spectrum. At one end are the ubiquitous tools of bench biology: pipettes, thermocyclers, centrifuges, and plate readers. These devices automate single operations but require substantial human oversight and intervention to connect steps in a workflow. At the other end of the spectrum lies the vision of fully autonomous experimentation, where systems generate hypotheses, design experiments, execute protocols, and analyze results without human involvement. Most synthetic biology laboratories operate somewhere in the middle, with partial automation of specific bottleneck operations.

Drawing from classification systems developed for manufacturing and autonomous vehicles[17], we can characterize this spectrum by the degree of human intervention required[4]. Manual operation with basic assistance represents the baseline, where researchers use hand tools and single-function instruments. Partial automation of multi-step workflows, such as liquid handling robots that can execute predefined pipetting sequences, reduces hands-on time but still requires human setup, monitoring, and troubleshooting. Conditional automation represents a higher level, where systems can handle complete protocols but may require human assistance when errors occur. High automation minimizes human involvement to consumable replenishment and waste removal. Full autonomy, while largely theoretical, would eliminate human intervention entirely. This classification system is summarized in Figure 2.1.

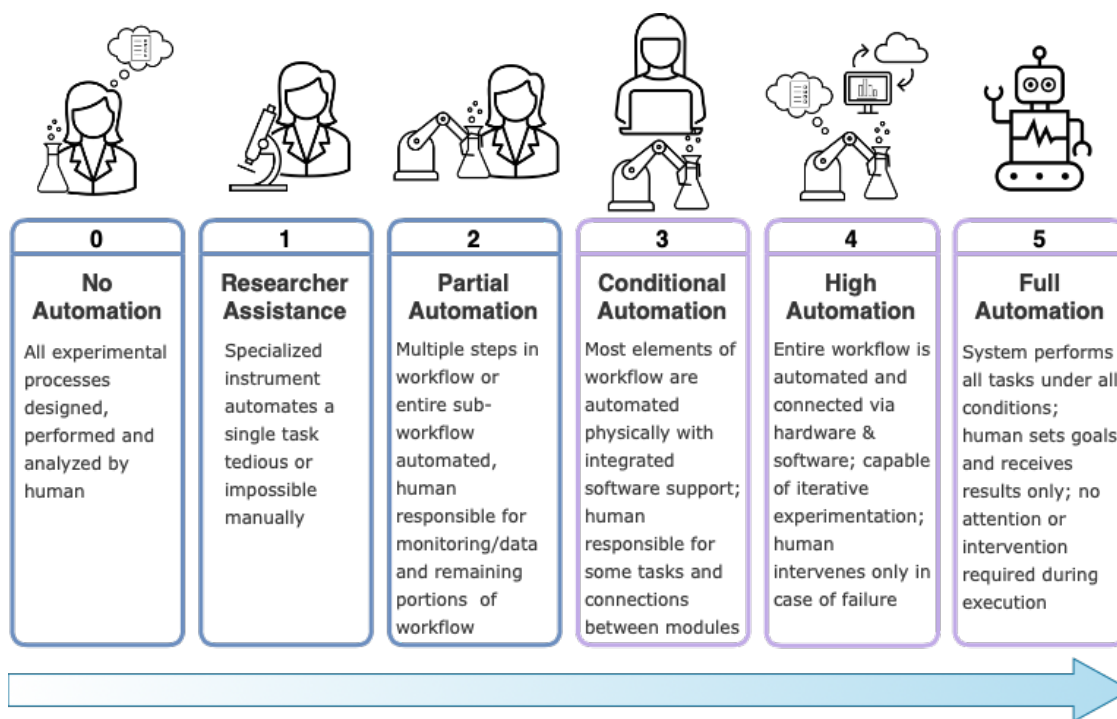


Figure 2.1: Levels of automation in synthetic biology laboratories achieved through hardware and software automation technologies.

The technology landscape reflects fundamental trade-offs between cost, throughput, flexibility, and ease of use. Robotic liquid handlers dominate in settings where throughput justifies capital investment. Basic pipetting robots like the Opentrons OT-2 cost \$5,000-10,000 and provide accessible entry points for automation, though with lower throughput than sophisticated workstations. High-end platforms like the Hamilton Star and Tecan Freedom EVO offer superior throughput and reliability but cost upward of \$100,000, require

significant laboratory space, and often need dedicated technical staff for operation and maintenance[4].

Microfluidic devices offer a different set of trade-offs[18]. By manipulating fluids at the microscale, these devices reduce reagent consumption and equipment size while taking advantage of unique fluid properties that emerge at small volumes. Channel-based microfluidics directs fluids through fixed paths etched into chips, enabling precise control but limiting flexibility. Each chip is essentially designed for a specific protocol, making these devices well-suited for established, high-volume applications but poorly suited for research environments where protocols evolve frequently.

Digital microfluidics takes a different approach[19]. Rather than flowing fluids through channels, DMF devices manipulate discrete droplets on arrays of electrodes. By applying voltage to specific electrodes, droplets can be moved, merged, mixed, or split anywhere on the array. This programmability means the same hardware can execute different protocols without physical modifications. The device can be reprogrammed through software alone, much like how a general-purpose CPU differs from fixed-function application-specific integrated circuits.

Despite these advantages, microfluidic technologies have struggled with commercialization and adoption[20, 21]. Several DMF-based commercial products, including Illumina's NeoPrep and NuGEN's Mondrian systems for sequencing library preparation, showed initial promise but were discontinued, reportedly due to reliability and manufacturing challenges[22]. The devices that remain on the market are either highly specialized for specific applications or prohibitively expensive for most academic laboratories.

This accessibility gap is particularly pronounced in academic settings. Budget structures favor personnel over equipment. High turnover of graduate students and postdocs makes long-term maintenance of complex automation systems challenging. Protocol variability across different projects limits the return on investment for specialized equipment. Until reliable, flexible automation systems become available at accessible price points, or until standardization enables broader adoption of existing technologies, laboratory automation in synthetic biology will remain concentrated in well-resourced industrial facilities and specialized biofoundries rather than becoming a standard research tool.

The development of PurpleDrop was motivated by this gap. Could an open-source, low-cost digital microfluidic platform provide sufficient reliability and flexibility to be useful in an academic research setting? What trade-offs would be necessary to achieve accessibility without sacrificing functionality?

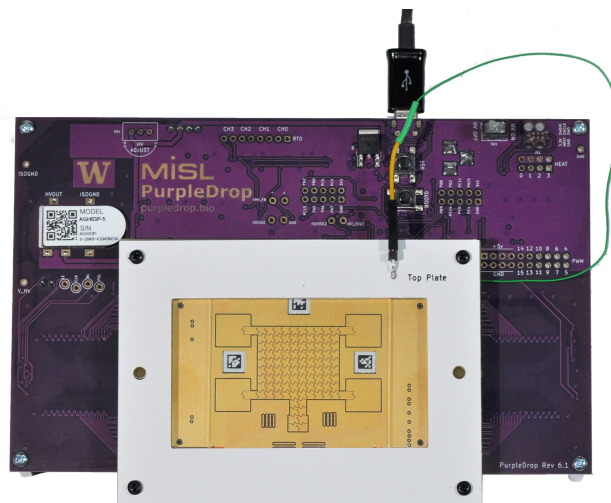


Figure 2.2: PurpleDrop Platform.

2.2 The PurpleDrop Platform

PurpleDrop (Figure 2.2) is a full-stack digital microfluidic platform designed to make flexible automation accessible to resource-constrained research environments[6]. The platform consists of custom hardware for droplet manipulation, peripheral systems for common molecular biology operations, and software for high-level control and error correction. The design prioritizes accessibility and modularity while maintaining the integration and closed-loop operation necessary for reliable automation.

2.2.1 Design Philosophy

Four principles guided the platform’s development. First, accessibility required minimizing cost and fabrication complexity. PurpleDrop uses affordable consumer materials and can be assembled for approximately \$300 without requiring cleanroom facilities. This represents orders of magnitude reduction in cost compared to commercial DMF systems or liquid handling robots.

Second, modularity enables adaptation to different applications without redesigning the entire system. The platform separates the motherboard printed circuit board, which houses all control electronics, from a daughterboard containing the electrode array. Different electrode configurations can be used with the same control hardware by swapping daughterboards. Peripheral modules for heating, fluidic input/output, and

magnetic bead manipulation attach to the core platform as needed.

Third, integration of multiple capabilities within a single platform reduces the need for sample transfers between instruments. Each transfer introduces opportunities for sample loss and contamination while requiring either manual intervention or complex robotic systems to move samples between stations. By incorporating heating, pumping, and sensing directly into the DMF device, entire protocols can execute without removing samples from the platform.

Fourth, closed-loop operation compensates for the limitations inherent in low-cost hardware. Commercial systems achieve reliability through precision manufacturing and high-quality components. PurpleDrop instead uses real-time feedback through computer vision and capacitance sensing to detect and correct errors as they occur. This software-level sophistication makes it possible to build a functional system from imperfect components.

2.2.2 Hardware Architecture

The core DMF platform exploits the electrowetting-on-dielectric phenomenon to manipulate aqueous droplets[23]. The electrode array consists of a printed circuit board patterned with a grid of individually addressable electrodes. Each electrode is covered with a thin dielectric layer of polyimide film and coated with Teflon AF to provide a hydrophobic surface. Droplets sit on this surface, sandwiched between the electrode array and a conductive top plate that serves as a common counter-electrode.

When voltage is applied between a specific electrode and the top plate, the electric field modulates the interfacial surface tension at the droplet-surface boundary. This change in surface tension causes the droplet to deform, pulling it toward the activated electrode. By sequentially activating adjacent electrodes, droplets can be moved along any path on the grid. Activating electrodes under different portions of a droplet can split it into smaller droplets, while moving two droplets onto the same electrode merges them. Repeated movement back and forth provides mixing through the chaotic advection that occurs within moving droplets.

Unlike many DMF devices that operate droplets in an oil medium to prevent evaporation, PurpleDrop functions in air. This simplifies setup and avoids contamination issues associated with oil, though it requires active management of evaporation for protocols involving heating or extended operation times.

The electronic control system uses a Raspberry Pi 3B single-board computer rather than a microcon-

troller. The Raspberry Pi runs Linux and features a quad-core processor with sufficient computational power to execute both the low-level electrode actuation and high-level planning and computer vision software directly on the device. This makes PurpleDrop self-sufficient, requiring no external host computer for operation. The Raspberry Pi communicates with shift registers on the motherboard through GPIO pins, allowing individual control of each electrode in the array.

2.2.3 Peripheral Systems

Several peripheral systems extend PurpleDrop's capabilities beyond basic droplet manipulation to enable complete protocol automation. The fluidic input/output system uses small peristaltic pumps to transfer samples on and off the device. For sample input, pumps connect to reservoirs via Teflon tubing fitted with pulled glass capillary tubes. The capillary tips are positioned at approximately 70 degrees above peripheral electrodes using PDMS support structures. This angled positioning, combined with the pointed tube geometry, allows droplets to be dispensed onto the board with minimal contact between the fluid and the input port. Contact can cause droplet pinning that electrowetting forces cannot overcome, resulting in sample loss.

Waste removal uses a similar approach. Standard capillary tubes positioned above peripheral electrodes use capillary action combined with pump pressure to draw waste droplets off the device. The same infrastructure supports sample transfer to external modules and output to downstream analysis instruments like sequencers.

Temperature control comes from resistive heaters integrated into three electrodes at one edge of the device. These heaters can maintain temperatures up to 100°C, sufficient for PCR thermocycling and heat-based elution steps. Integrated temperature sensors provide feedback for closed-loop temperature regulation.

Magnetic bead manipulation enables on-board purification and separation protocols. A motorized arm positions a magnet beneath the electrode surface, directly under droplets containing magnetic particles. When activated, the magnet pulls beads into a pellet at the bottom of the droplet. The supernatant can then be moved away via electrowetting, leaving the pelleted beads behind (Figure 2.3). After separation, beads can be resuspended in fresh buffer for washing or elution. This capability is particularly valuable because it eliminates sample transfer steps that often result in material loss. An entire magnetic bead wash protocol, from initial binding through multiple wash steps to final elution, can occur on the device with only the initial

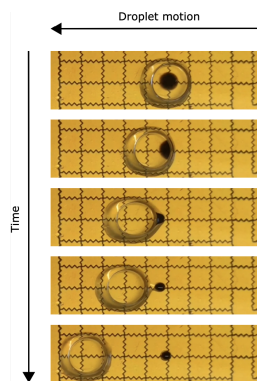


Figure 2.3: Bead wash on PurpleDrop. A below-board motorized magnetic arm positions a magnet underneath a droplet containing magnetic beads in the first frame. The droplet is moved away via electrowetting while the magnet is held in place. The droplet on the left in the final frame contains the constituents of the initial droplet that did not bind to the magnetic beads upon mixing and incubation. Resuspension and elution of the pellet on the right yields the bound products.

input and final output requiring fluid transfer.

PCR presented unique challenges for an air-based DMF device. Initial attempts at on-board thermocycling encountered severe evaporation and bubble formation. Droplets would fragment and eject microdroplets that contaminated the electrode surface, making the approach untenable. Testing various oil media to control evaporation proved unsuccessful despite trying silicone and mineral oils of different viscosities and purity grades.

The solution uses a hybrid digital/channel-based approach with a sealed PDMS chamber (see Figure 2.4). The chamber consists of a cylindrical cavity punched from PDMS with thin PDMS sheets forming top and bottom seals. A small channel connects the chamber to a capillary tube for fluid transfer. Samples are pumped from the electrode surface into the chamber, with a small air pump drawing air out through the gas-permeable PDMS to facilitate fluid entry.

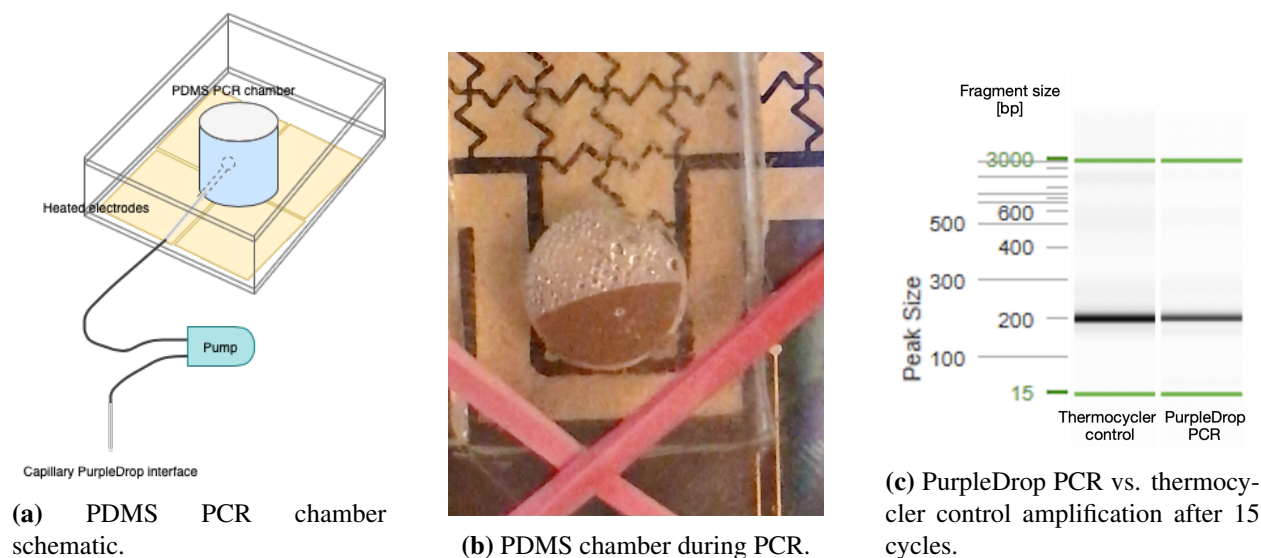


Figure 2.4: PurpleDrop PCR automation.

During thermocycling, the chamber is positioned over the heated electrodes and compressed via mechanical pressure to seal the input channel. The PDMS material allows air bubbles to diffuse out while containing the liquid sample. Evaporated water condenses on the smooth, hydrophobic PDMS lid and rolls back into the main sample rather than being lost. After thermocycling, reversing the pump flow direction retrieves the product. Using mineral oil as a carrier fluid further reduces evaporation and sample loss during transfer steps. This design enables efficient PCR amplification while maintaining the integrated, automated workflow.

2.2.4 Software Control and Error Correction

The Puddle runtime system (developed primarily by Max Willsey) provides high-level control of PurpleDrop through a Python API[16]. Rather than requiring users to manually specify electrode actuation sequences, Puddle presents operations like mixing, splitting, heating, and input/output as simple function calls. The system dynamically manages fluidic resources, automatically handling placement and routing of droplets on the device.

Real-time error detection and correction distinguishes Puddle from previous microfluidic control systems. A computer vision system identifies droplets and tracks their positions and sizes in real time. Unlike previous vision-based approaches that required template images or carefully controlled lighting, the hue-

based detection proves robust to lighting changes and handles droplets of varying shapes and sizes.

The vision system runs directly on the Raspberry Pi, processing 320×240 pixel images to detect droplet locations and volumes. When the actual state diverges from the expected state, for example if a droplet fails to move due to a faulty electrode, the system triggers a rollback and replanning. Failed electrodes are marked as unusable, and subsequent operations avoid those regions. This error correction significantly extends the operational lifetime of the device. In endurance testing, the system successfully detected and corrected errors arising from both inherent hardware flaws like poor electrical contacts and use-induced defects such as surface contamination from evaporation. Protocols could complete successfully despite accumulating multiple failures that would have caused traditional microfluidic devices to become unusable.

The combination of hardware and software systems provides PurpleDrop with capabilities spanning the requirements for complex protocol automation. The platform can manipulate droplets of various volumes, input samples from reservoirs and output products to collection tubes or other devices, perform thermocycling and temperature-controlled incubations, execute multi-step magnetic bead wash protocols with minimal sample loss, efficiently amplify DNA via PCR, measure droplet volumes in real time via computer vision, automatically detect and recover from hardware failures, and expose all of these capabilities through a high-level programmable interface. The next two sections demonstrate PurpleDrop’s performance through case studies in synthetic biology applications.

2.3 Application Case Study: DNA Data Storage

DNA offers extreme information density for data storage, with theoretical capacity reaching exabytes per cubic millimeter, along with exceptional durability for archival applications[24]. Recent work has demonstrated encoding digital data into DNA sequences, synthesizing those sequences, and reading them back via sequencing. However, practical DNA storage systems require physical organization of samples to enable random access without sequencing entire pools.

We developed a storage architecture where dehydrated DNA spots are arranged on glass cartridges[15]. Each spot represents a physically isolated pool that can use the same addressing scheme, scaling storage capacity beyond what a single liquid pool could provide. Individual cartridges could theoretically store up to 50 terabytes using current DNA storage techniques. The key challenge was retrieval: selectively

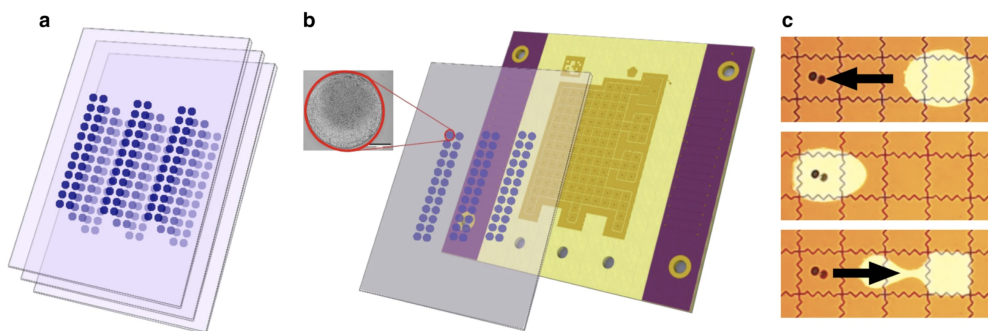


Figure 2.5: Storing physically isolated spots of dehydrated DNA on glass cartridges enables high information density. **a** Spotted cartridges can be further organized into decks. **b** To retrieve data, the desired cartridge is first loaded onto the DMF device consisting of an electrode grid used to perform retrieval functions. An actual magnified spot is depicted, where the scale bar in the inset is $275\ \mu\text{m}$. **c** Next, a water droplet sandwiched between the cartridge and electrodes is actuated to move under the spotted DNA for rehydration. Depicted is an actual droplet moving on the device. After retrieval, we manually recover the sample and analyze it to assess DNA recovery rate and contamination across samples sharing paths on the DMF device.

rehydrating and recovering specific DNA spots without contamination from neighboring spots or sample loss.

PurpleDrop’s capabilities proved well-suited to this challenge. Glass cartridges spotted with dehydrated DNA serve as the top plate of the DMF device. To retrieve a spot, a water droplet is actuated to move beneath the target spot, rehydrating its contents. After a brief dwell time, the droplet containing the dissolved DNA moves away for downstream processing or output. This process is depicted in Figure 2.5.

We evaluated retrieval efficiency across varying spot masses from 5 to 60 nanograms, dwell times from 1 to 120 seconds, and file sizes up to 2.7 megabytes of data encoded in DNA. Recovery rates averaged 79% of the spotted DNA mass after 60 seconds of dwell time. Even with just 1 second of dwell time, 77% of the mass was recovered. In all cases, enough DNA was retrieved to successfully decode the stored data with no bit errors, even when substantial fractions remained on the cartridge.

Testing dense spot arrangements in 3×4 arrays showed negligible contamination between neighboring spots, even when separated by only one electrode width. We also demonstrated sequential retrieval of three different files with a single water droplet, moving it from spot to spot to accumulate all three DNA pools. All files were successfully recovered with only 0.5 to 1.4% of unique sequences missing.

To demonstrate the scalability of this approach, we calculated that 18 nanograms of successfully se-

quenced DNA would contain 1 terabyte of data, accounting for the encoding efficiency and sequence structure used in current DNA storage systems. Allowing for 2× loss in spot retrieval and an additional 5× duplication for sequencing error correction, we spotted and retrieved in excess of 180 nanograms. Retrieval and quantification confirmed that 1 terabyte of data could be successfully stored and retrieved from a single spot using this method.

This work demonstrated that PurpleDrop’s combination of precise droplet control, computer vision feedback, and programmable automation could enable practical DNA storage systems with random access capabilities. The platform successfully addressed the physical storage and retrieval challenges that complement the encoding and sequencing aspects of DNA data storage.

2.4 Application Case Study: Aptamer Discovery via SELEX

Systematic Evolution of Ligands by Exponential Enrichment, commonly known as SELEX, is the standard method for discovering aptamers[25]. These short oligonucleotide sequences can fold into structures that bind specifically to molecular targets ranging from small molecules to proteins and cells. While powerful, SELEX is notoriously time-consuming, typically requiring 8 to 15 rounds over weeks or months to identify aptamers. The process is also labor-intensive and prone to failure due to the many variables that must be optimized, including library design, target concentration, incubation conditions, and PCR parameters.

A typical SELEX cycle involves incubating a random DNA library with immobilized target molecules, separating bound sequences from unbound sequences, amplifying the bound sequences via PCR, converting the double-stranded PCR products back to single-stranded DNA, and using the amplified pool as input for the next round. Each step involves multiple sub-steps prone to sample loss and human error.

PurpleDrop’s modular capabilities map naturally onto SELEX requirements (see Figure 2.6). For target-library incubation, magnetic beads functionalized with biotinylated target protein are mixed with the DNA library via droplet merging and electrowetting-based mixing. The magnetic arm system enables efficient separation of bead-bound sequences from unbound sequences, followed by multiple wash steps to remove non-specific binders. The PDMS chamber module amplifies selected sequences, with heating during the initial denaturation step serving to elute bound DNA from the magnetic beads. Computer vision tracks droplet volumes throughout, enabling replenishment if evaporation occurs. The entire process runs without

manual intervention once samples are loaded.

To validate the approach, we performed one complete SELEX round using a library spiked with a known thrombin-binding aptamer at 10% concentration. This allowed us to measure enrichment efficiency by comparing the aptamer percentage before and after selection and amplification. We used human alpha-thrombin as the target protein, immobilized on streptavidin magnetic beads via biotin linkages.

Results showed successful enrichment comparable to laboratory controls performed manually with conventional equipment. The positive test cases, where thrombin protein was present during selection, showed strong enrichment from 10% to approximately 75-79% aptamer content. PurpleDrop achieved 79.4% enrichment with 323,153 sequencing reads, while the laboratory control reached 75.9% with 419,101 reads. The negative control, performed without thrombin protein, remained near the initial 10% aptamer concentration at 8.6%, confirming that enrichment required the target protein rather than resulting from PCR bias or other artifacts.

These results established the feasibility of multi-cycle SELEX automation on PurpleDrop. The platform's modular design makes it straightforward to extend to multiple rounds, with each round's output serving as the next round's input. More importantly, the integration of sensing and computation enables closed-loop optimization that would be difficult to achieve with conventional manual SELEX. Real-time volume monitoring can trigger automatic replenishment to compensate for evaporation. Integration with nanopore sequencing via the input/output system would enable monitoring library diversity between rounds without waiting for batch sequencing results. Computational models could use sequencing data to predict convergence and determine when to stop iterating. Experimental parameters such as incubation time, wash stringency, and PCR cycle number could be adjusted dynamically based on intermediate results.

When performing optimally, this integrated workflow could provide a significant advance over conventional SELEX, where decisions about when to sequence, when to stop iterating, and how to adjust parameters rely on researcher intuition and are made between rounds rather than in real time. The automation also addresses the reproducibility challenges inherent in manual SELEX, where subtle variations in technique between researchers or even between days can significantly impact outcomes.

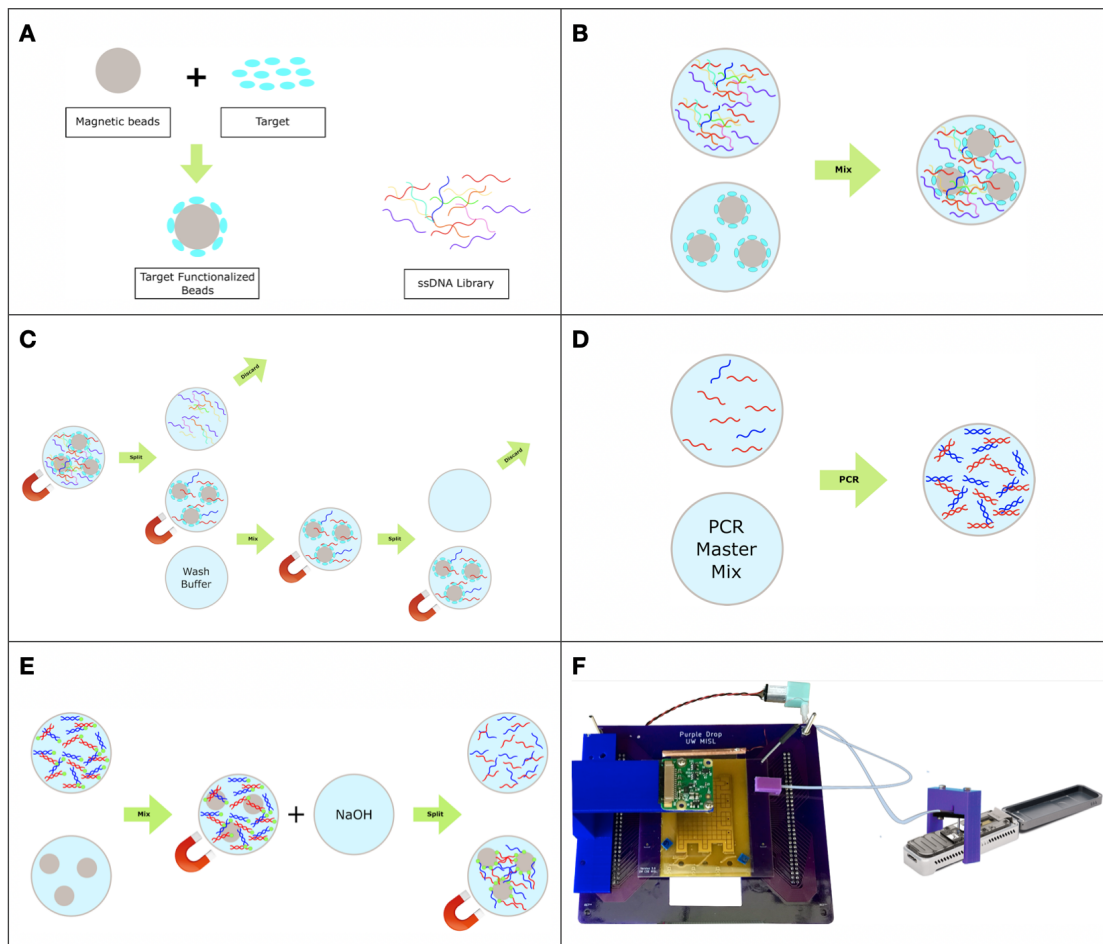


Figure 2.6: SELEX translated into droplets. A. Prepare target-functionalized magnetic beads and initial single stranded random DNA library. B. Mix and incubate target-bead and library droplets. C. Perform magnetic bead wash and discard unbound strands to waste. D. Combine selection product droplet with PCR master mix and transfer to PDMS chamber for amplification. E. Perform magnetic bead based strand digestion of dsDNA PCR product via biotinylation of one strand of each helix during PCR with biotinylated reverse primer. F. Output aliquot of SELEX product to nanopore for sequencing; repeat B through F until sequence convergence.

2.5 Critical Assessment: Limitations and Lessons

PurpleDrop successfully demonstrates that low-cost, flexible DMF automation is technically feasible. The platform’s modular design and closed-loop error correction enable complex multi-step protocols. The open-source design addresses the accessibility barrier that limits automation adoption in academic settings, with material costs around \$300 compared to tens or hundreds of thousands of dollars for commercial alternatives. However, the gap between proof-of-concept demonstrations and production-ready systems remains substantial. Several limitations became apparent through extended use of the platform, and these limitations reveal broader challenges facing accessible laboratory automation.

Reliability proved to be the most persistent challenge. Despite the error correction capabilities built into Puddle, hardware failures remain common. Electrodes degrade with repeated use, particularly those used for heating or frequently actuated paths. Surface contamination from evaporated droplets or oil degradation can cause inconsistent droplet actuation. Pump failures occasionally occur, requiring manual intervention to clear blockages or replace tubing. In practice, protocols that execute reliably one day may fail the next due to accumulated wear or subtle changes in environmental conditions.

The error correction system can compensate for isolated failures, such as a single faulty electrode or a droplet that temporarily fails to move. However, it cannot overcome systematic problems like widespread surface contamination or multiple simultaneous component failures. Troubleshooting these issues requires expertise in both the hardware and software systems, creating a barrier for users who simply want to run protocols rather than maintain the platform.

Throughput represents another significant constraint. PurpleDrop handles one or a few samples at a time, operating on individual droplets in sequence. While the low cost makes parallel operation of multiple devices feasible in principle, this approach requires multiplying not just the hardware but also the space, maintenance burden, and potential points of failure. Throughput remains far below commercial liquid handlers that process 96-well plates in parallel, limiting PurpleDrop’s utility for applications requiring high sample numbers.

The PCR module, while functional, shows reduced thermal efficiency compared to commercial thermocyclers. The PDMS chamber approach successfully prevents evaporation and enables thermocycling, but the thermal mass of the chamber and imperfect contact with the heating elements result in slower temper-

ature ramps and less uniform heating. This can be partially compensated through protocol optimization, such as extending cycle times, but represents a fundamental trade-off inherent in the hybrid digital-channel approach.

Integration complexity emerged as a more subtle but equally important limitation. While individual modules for bead wash, PCR, and input/output function independently, integrating them into seamless multi-step workflows requires careful protocol development and debugging. The transition points between operations, such as transferring a sample from the electrode surface into the PCR chamber or recovering it afterward, introduce opportunities for failure. Small variations in droplet volume, surface conditions, or pump timing can cause these transitions to fail in ways that are difficult to predict or diagnose.

This highlights a gap between "it works in principle" and "it works reliably." Demonstrating that a platform can perform a complete protocol once or a few times differs substantially from creating a system that performs that protocol reliably enough to be useful for routine research. The latter requires addressing countless edge cases, failure modes, and usability issues that consume disproportionate development effort relative to their visibility in publications.

Perhaps most tellingly, despite the open-source design and published documentation, PurpleDrop has not been widely adopted by other research groups. This suggests that barriers remain in fabrication, setup, or usability that prevent the platform from being truly accessible. Even with detailed instructions and available materials, the expertise required to assemble, calibrate, and troubleshoot the system may be prohibitive for groups without microfluidics experience.

These limitations reveal several broader challenges facing accessible laboratory automation. First, the reliability-cost trade-off is real and difficult to overcome. Low-cost hardware necessitates sophisticated software compensation, but software cannot overcome all hardware limitations. There may be a minimum threshold of hardware quality below which no amount of clever software can achieve production-level reliability.

Second, integration is substantially harder than automating individual operations. Automating discrete steps like mixing, heating, or washing is tractable with current technology. Integrating these steps into robust end-to-end workflows that handle the transitions and error conditions gracefully requires a level of system engineering that is often underestimated.

Third, flexibility and optimization exist in tension. DMF's programmability is both its strength and weakness. The ability to reconfigure the platform for different applications prevents the deep optimization possible with dedicated hardware. A thermocycler optimized solely for PCR will likely always outperform a general-purpose platform that also does PCR among other operations.

Finally, the "last 10%" problem is pervasive in automation. Moving from demonstrations that work under controlled conditions to systems that work reliably in diverse real-world conditions requires addressing an enormous number of details. These details are often invisible in published work but determine whether a platform becomes a useful tool or remains a research prototype.

Addressing these limitations would require several advances. Improved electrode materials and surface coatings could enhance durability and resistance to contamination. Better thermal management for PCR, perhaps through a dedicated heated metal block around the PDMS chamber rather than relying solely on electrode heaters, could improve amplification efficiency. Standardized protocols and comprehensive troubleshooting guides could lower the barrier for new users. Hybrid approaches that combine DMF's flexibility for sample manipulation with dedicated modules for high-reliability operations like thermocycling might offer better trade-offs than attempting to do everything with DMF alone.

More fundamentally, the PurpleDrop experience suggests that widespread laboratory automation may require not a single platform but an ecosystem of modular, interoperable tools at various price points and capability levels. This mirrors the current landscape of laboratory equipment, where different instruments handle different operations, but would require substantially better standardization and integration than exists today. Efforts like the SiLA consortium[26] working on standardized communication protocols for laboratory instruments represent steps in this direction, but widespread adoption remains distant.

2.6 Conclusions

Physical automation addresses complementary bottlenecks to the computational methods that form the focus of subsequent chapters. The work described in this chapter demonstrates that flexible, low-cost automation is technically achievable and can enable applications from DNA data storage to aptamer discovery. PurpleDrop successfully automated complex molecular biology workflows using digital microfluidics at a fraction of the cost of commercial alternatives.

However, the platform’s current limitations in reliability, throughput, and integration complexity reflect broader challenges facing accessible laboratory automation. The gap between proof-of-concept demonstrations and production-ready systems remains substantial. Until these challenges are addressed through improved hardware, better software abstractions, or new approaches to modular system design, laboratory automation will likely remain concentrated in specialized facilities with dedicated technical staff rather than becoming a ubiquitous research tool.

Nevertheless, the successful demonstrations of DNA storage retrieval and SELEX automation illustrate the potential for digital microfluidics to enable experimental workflows that would be impractical with manual execution. The lessons learned from developing and using PurpleDrop informed my thinking about scalability challenges in molecular engineering more broadly. Physical automation can reduce certain bottlenecks, particularly those involving repetitive liquid handling operations, but introduces its own complexity and maintenance burden.

The subsequent chapters focus on computational approaches to reducing experimental burden in nanopore sequencing applications. These computational methods complement physical automation by reducing the number of experiments needed rather than automating their execution. Where PurpleDrop sought to make existing experimental workflows faster and more reproducible, the modeling work in Chapters 3-5 seeks to replace some experiments entirely with computational predictions. Both approaches address the fundamental challenge of making molecular engineering more scalable, but they operate at different levels and face different trade-offs between accuracy, generality, and resource requirements.

Chapter 3

Background: Nanopore Sequencing

3.1 Introduction to Nanopore Sequencing Technology

Nanopore sequencing represents a transformative approach to nucleic acid analysis that enables direct, single-molecule interrogation of biopolymers in real-time[7]. While this chapter focuses on DNA sequencing, the technology's single-molecule readout capabilities extend to other biopolymers including proteins, which will be addressed in later chapters. Unlike traditional sequencing-by-synthesis platforms, nanopore technology reads molecules as they pass through a nanoscale pore, measuring characteristic changes in ionic current that correspond to the molecular structure occupying the sensing region. This direct readout mechanism offers several key advantages: it requires no amplification or labeling, can sequence long molecules processively, and, most relevant to this work, has the inherent capacity to detect chemical modifications that alter the current signature.

The basic principle of nanopore sequencing involves applying a voltage across a membrane containing a single nanopore, which drives ionic current through the pore. When a charged biopolymer (such as DNA) is electrophoretically driven through the nanopore, it partially occludes ion flow, producing a characteristic reduction in current. The magnitude and duration of this current blockade depends on the physical and chemical properties of the molecule within the pore's sensing region. For DNA, this typically encompasses 4-9 nucleotides simultaneously, depending on the pore geometry and chemistry[27, 28]. As the molecule translocates, the changing sequence context produces a time-varying current signal, often called a "squig-

gle", that can be computationally decoded back into the underlying nucleotide sequence.

Commercial nanopore sequencing platforms, particularly those developed by Oxford Nanopore Technologies (ONT), have matured substantially in recent years. These systems employ biological protein nanopores embedded in synthetic membranes, with sophisticated instrumentation to measure picoampere-scale current fluctuations. The MinION and other ONT devices enable massively parallel sequencing across arrays of hundreds to thousands of individual nanopores, providing high-throughput, real-time sequencing capabilities that have found applications ranging from clinical diagnostics to environmental monitoring[7].

3.2 From Signal to Sequence: Basecalling and the Role of k-mer Models

The process of converting raw ionic current measurements into nucleotide sequences, known as basecalling, represents a central computational challenge in nanopore sequencing[29]. Early basecalling approaches relied on hidden Markov models (HMMs) and explicit k-mer models that mapped each possible k-nucleotide sequence context to an expected current level. These k-mer models reflect the physical reality that multiple nucleotides simultaneously occupy the nanopore’s sensing region, with each contributing to the observed current in a context-dependent manner.

A k-mer model typically consists of a lookup table associating each possible k-nucleotide sequence with statistical parameters (mean and standard deviation) describing its characteristic current distribution. For example, a 6-mer model for canonical DNA contains $4^6 = 4,096$ entries, each representing a unique sequence context. During basecalling, the observed current trace is segmented and compared against these expected distributions to identify the most likely sequence that generated the observed signals. The k-mer length reflects the effective sensing region of the nanopore. Longer k-mers generally improve accuracy by capturing more context, but exponentially increase the number of parameters that must be empirically determined[27, 30].

Modern basecalling has largely transitioned to deep learning approaches that learn to map raw signals directly to sequences. ONT’s current production basecallers (such as Guppy and the newer Remora and Dorado) employ sophisticated neural architectures trained on large datasets of nanopore reads with known ground-truth sequences[31, 32]. These end-to-end models can implicitly learn context dependencies without requiring explicit k-mer tables, and have dramatically improved basecalling accuracy, with recent models

achieving >99% raw read accuracy for canonical DNA under optimal conditions.

However, k-mer models remain important for several reasons. First, they provide interpretable, biophysically grounded representations of how sequence context influences current. Second, they enable rapid prototyping and testing of new pore chemistries or modifications without requiring massive training datasets. Third, they continue to play roles in signal processing pipelines, including event detection, signal normalization, and modified base detection. Understanding k-mer-level relationships between sequence and signal thus remains foundational, even as neural network basecallers dominate production workflows[27].

3.3 Detection of Modified Bases: Challenges and Opportunities

One compelling application of nanopore sequencing is its potential to directly detect chemical modifications to nucleic acids. Epigenetic modifications such as 5-methylcytosine (5mC) and N6-methyladenine (6mA) in DNA, or pseudouridine and N6-methyladenosine (m6A) in RNA, play crucial regulatory roles in biology[33, 34]. Traditional sequencing methods typically cannot distinguish modified from canonical bases, requiring separate chemical treatments (such as bisulfite conversion for 5mC) that add complexity, potential artifacts, and loss of information[35, 36].

Nanopore sequencing offers a fundamentally different approach: because the ionic current depends on the physical and chemical properties of the molecule in the pore, modifications that alter these properties can produce distinguishable current signatures. This has enabled direct detection of various natural modifications, with ONT and others developing specialized models and algorithms for modification calling. However, this capability comes with significant challenges.

3.4 Why Modified Base Detection Is Difficult

The difficulty of detecting modifications stems from several factors. First, the current differences between modified and canonical bases are often subtle and typically much smaller than the differences between the four canonical bases themselves. A single methyl group, for instance, represents a minor perturbation to the overall molecular structure compared to the distinction between, for example, adenine and thymine[37]. This small signal must be reliably detected above the substantial intrinsic noise in nanopore measurements.

Second, modifications occur in varied sequence contexts, and the context itself strongly influences the current signal[38, 39]. A modified base may produce different signatures depending on its neighboring nucleotides, requiring models that account for both the modification and its context. For k-mer-based approaches, this means the number of possible contexts grows multiplicatively: a single modification in a 6-mer context requires characterizing $4^5 = 1,024$ different modified k-mers, compared to the 4,096 canonical k-mers.

Third, many modifications are chemically similar to canonical bases or to each other, making discrimination challenging. For example, distinguishing between unmethylated cytosine, 5-methylcytosine, and 5-hydroxymethylcytosine requires resolving very fine differences in current signatures[34]. The similarity between certain modifications and canonical bases (or between different modifications) sets fundamental limits on detection sensitivity and specificity.

Fourth, experimental calibration for each modification is resource-intensive. Generating reliable k-mer models requires synthesizing or obtaining DNA/RNA containing the modification in diverse sequence contexts, generating nanopore data, and carefully analyzing signal distributions[30]. For natural modifications this is challenging but feasible; for synthetic or rare modifications, obtaining sufficient calibration material becomes prohibitive.

These challenges are substantially amplified when moving beyond natural modifications to synthetic or xenonucleic acid (XNA) bases, which may have dramatically different chemical structures and properties compared to canonical nucleotides. While natural modifications typically involve small chemical additions to standard bases (methyl groups, hydroxyl groups, etc.), XNAs can incorporate entirely novel base structures, modified sugar backbones, or alternative chemical scaffolds[40]. This greatly expands both the opportunity space, enabling applications from expanded genetic alphabets for data storage to novel therapeutic molecules, and the difficulty of developing sequencing methods.

3.5 Current Approaches to Modified Base Detection

Several strategies have emerged for nanopore-based modification detection. The most direct approach involves training specialized basecalling models that explicitly include modification states. ONT’s Remora and Dorado platforms, for example, can be trained to output not just the canonical base call but also a prob-

ability that the base carries a specific modification. These models typically use neural networks trained on datasets containing known modified and unmodified examples in diverse contexts.

An alternative approach uses k-mer-level models to perform statistical hypothesis testing. After initial basecalling with a canonical model, the algorithm examines whether the observed current levels at each position are more consistent with a canonical or modified k-mer model. This can be implemented using likelihood ratio tests or similar statistical frameworks. This approach has been successfully applied to detect 5mC and other modifications[37], though it requires having empirically measured k-mer models for the modification of interest.

Hybrid approaches combine elements of both strategies, using neural networks for initial basecalling but incorporating k-mer-level information or statistical testing for modification calling[41, 42]. The key limitation across all these approaches is the requirement for training or calibration to detect a modification, the system must have been exposed to examples of that modification during development.

3.6 The Need for More Scalable Computational Methods

The challenges outlined above, particularly the exponential growth in required calibration data as k-mer length, alphabet size, and number of modifications increase, motivate the development of computational methods that can predict nanopore signals from molecular properties. If models could learn the underlying relationship between chemical structure and ionic current, they could potentially generalize to unseen modifications with minimal or no additional experimental calibration.

This is precisely the gap that the following chapter addresses. Rather than requiring exhaustive measurement of every possible k-mer context for each XNA base, computational models that incorporate structural and chemical information could enable prediction of current signatures from molecular features alone. Such models would dramatically reduce the experimental burden of developing sequencing capabilities for expanded alphabets, accelerate the design-build-test cycle for novel nucleic acids, and potentially enable true *de novo* detection of previously uncharacterized modifications.

The development of these computational approaches draws on recent advances in molecular machine learning, including graph neural networks that can represent molecular structures as networks of atoms and bonds, and large language models that can learn patterns from chemical string representations. By

training these models on available data from canonical and modified bases, we can explore whether they capture sufficient information about the sequence-structure-signal relationship to make accurate predictions for novel chemistries.

This background sets the stage for the work presented in the next chapter, where we develop and benchmark structure-informed machine learning models for predicting nanopore ionic currents from DNA containing xenonucleic acid modifications. The goal is to move toward a more scalable, generalizable framework for expanded alphabet sequencing: one that reduces dependence on exhaustive experimental measurements while maintaining the accuracy needed for practical applications.

Chapter 4

Structure-informed Models for Ionic Current Prediction in Nanopore Sequencing of Expanded DNA Alphabets

Nanopore sequencing enables direct, single-molecule interrogation of biopolymers and shows promise for analyzing not only DNA and RNA, but also chemically modified bases, proteins, and other polymers. Expanded DNA alphabets, such as those found in xenonucleic acids (XNAs), open new possibilities for diagnostics, therapeutics, data storage, and engineered biology. However, robust sequencing strategies for these modified molecules remain lacking. While nanopore-based tools exist for some noncanonical bases, they often require extensive experimental calibration by measuring each base across many sequence contexts, which limits scalability and increases cost. In this chapter, we investigate computational methods for predicting the ionic current signals produced during nanopore sequencing of DNA containing noncanonical XNA bases, aiming to reduce the need for experimental calibration. We compare a sequence-based predictive model with two structure-aware approaches: one using graph-based molecular representations and another adapting a generative language model to molecular SMILES. Our findings show that while sequence context captures much of the signal variability, incorporating structural and chemical information improves predictive accuracy in specific cases. These results highlight the value of structural data representations and model design in scaling XNA sequencing, and suggest this framework could extend to modeling

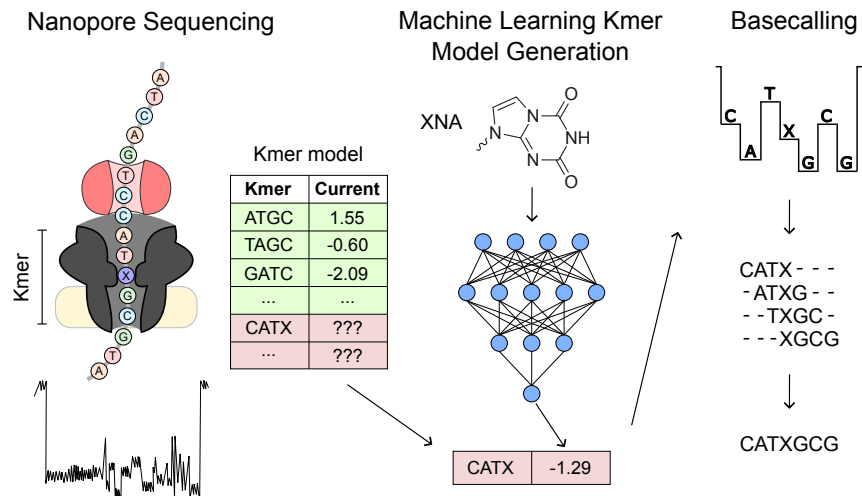


Figure 4.1: Overview of structure-informed XNA nanopore sequencing workflow. Machine learning models trained on k-mer current levels from canonical DNA and characterized XNA bases are used to predict ionic currents for previously unseen XNA-containing k-mers (X represents a held-out XNA base). Model-generated k-mer tables are integrated into a basecalling pipeline to enable sequencing of DNA containing xenonucleic acid substitutions.

ionic currents from other complex biomolecules, such as proteins. This chapter includes materials originally published in [27].

4.1 Introduction

Expanding the four-letter alphabet of DNA beyond the standard nucleotides (A, T, G, C) holds great promise for synthetic biology, biotechnology, and medicine. As DNA synthesis and sequencing technologies have matured, researchers can now design synthetic nucleotides with new chemical functionalities or novel base-pairing properties. Unnatural base-pairing xenonucleic acids (XNAs) are at the forefront of this work, offering the potential to expand the chemical, structural, and binding capabilities of DNA beyond those found in nature. Expanded genetic alphabets could enable applications ranging from novel therapeutics to enhanced data storage systems and engineered organisms with new capabilities[10, 9, 11, 12, 8]. Many XNAs have been synthesized with modified sugars, backbones, or nucleobases[40], but compared to natural DNA there remains a shortage of biomolecular tools and commercial infrastructure to support their widespread use[35, 30].

A major barrier to XNA research is the lack of compatible next-generation sequencing methods. While

natural DNA can be read rapidly, cheaply, and at scale, sequencing methods for modified or non-standard nucleotides lag decades behind. Current approaches rely on low-throughput and less generalizable techniques such as mass spectrometry, gel shift assays, or enzymatic conversion of XNAs into standard bases for Sanger sequencing[30]. Adapting sequencing-by-synthesis platforms like Illumina would require major changes to reagents, polymerases, instruments, and analysis pipelines[30]. Some progress has been made, for example, Wang et al. used an enzyme-assisted strategy to sequence a six-letter alphabet (A, T, G, C, Z^n , P) by enzymatically converting synthetic bases into canonical ones[35]. However, this strategy is limited to a single unnatural base pair, and requires sequencing both treated and untreated samples, reducing generality and efficiency.

Nanopore sequencing provides a more flexible platform for reading non-standard nucleotides due to its ability to directly sequence single molecules without amplification or labeling[7, 34, 43]. Prior work has explored its use for XNAs. Perez et al. combined nanopore sequencing with machine learning to detect hydrophobic bases Ds and Px[44], while Kawabe et al. synthesized and sequenced 12-letter alphabets containing up to four orthogonal base pairs[30]. In these studies, characteristic current signals for each XNA were empirically measured and assembled into models for classification.

Nanopore basecalling relies on decoding ionic current signals as DNA passes through a pore[7]. Existing algorithms are trained on canonical DNA and cannot readily identify XNAs. K-mer-based models have therefore played a key role in detecting modified nucleotides. These models assume that the current is influenced not only by the central base but also by its immediate neighbors, which together form the k-mer. Longer k-mers generally improve accuracy[28, 39], but the number of sequence contexts grows exponentially with k-mer size, driving up experimental cost. Flow cell chemistry also affects the effective k-mer size; for example, ONT’s R10.4.1 pores often require 5–9mers due to double reader-head geometry[28, 45]. Modeling becomes even more challenging when multiple modifications are considered, as the number of possible k-mers grows polynomially. Moreover, empirical k-mer models neglect the chemical and structural features of nucleotides, limiting their ability to generalize to unseen or chemically related variants.

These limitations highlight the need for models that incorporate chemical and structural information. Richer molecular representations could improve robustness and generalizability, reducing dependence on exhaustive empirical calibration and opening the door to *de novo* predictions. Recent advances in artificial

intelligence have spurred progress in molecular representation learning. Approaches range from molecular fingerprints and SMILES notations to graph-based encodings[46], and deep learning models including convolutional neural networks (CNNs), recurrent networks, graph neural networks (GNNs), and transformer-based large language models (LLMs)[47, 48]. GNNs capture molecular connectivity by modeling atoms and bonds as graphs, while LLMs, originally developed for natural language, can be applied to chemical strings such as SMILES, uncovering patterns that inform molecular property prediction.

Here we evaluate molecular structure-based architectures for predicting nanopore current levels of DNA containing single XNA substitutions. Unlike prior XNA sequencing studies, which relied on fully empirical k-mer measurements[30, 44], our approach aims to reduce experimental burden by modeling the chemical and structural properties of XNA-containing k-mers. As a baseline, we implemented a sequence-only CNN. We then compared two structure-based methods: (i) a graph isomorphism network (GIN) that encodes k-mer molecular graphs with chemical features, and (ii) a GPT-2 model fine-tuned on SMILES representations of k-mers. We benchmarked all three approaches on ten distinct XNA bases and further assessed their utility for basecalling by constructing predicted k-mer tables and testing them in a custom XNA-aware basecaller.

4.2 Materials and Methods

Detailed protocols for XNA library preparation, nanopore sample preparation, data acquisition, raw data pre-processing, signal-to-sequence mapping, and k-mer model benchmarking are described in [27, 30]. Briefly, barcoded heptamer libraries containing all possible combinations of NNNNNNN (N = A,T,G,C; N = XNA base) were synthesized, sequenced on R9.4.1 Flongle flow cells, and processed using the Xenomorph v1.2.0 pipeline (<https://github.com/xenobiolab/xenomorph>) to extract normalized mean current levels for all 256 possible 4-mer contexts per XNA base. The following sections describe the machine learning model architectures developed for this work.

4.2.1 Sequence-based CNN model

To compare our graph-based model and the GPT-2 model that incorporate k-mer chemical structure information with a baseline model that relies solely on sequence-level information, we trained a simple convolutional neural network (CNN). Each k-mer sequence was represented as a one-hot encoded matrix (k-mer length X

number of possible bases), and used as input to the model. The associated mean ionic current level served as the label for each example.

The architecture of the CNN consists of two 1D convolutional layers, each followed by a ReLU activation function and max pooling. The first convolutional layer uses 32 filters with a kernel size of 3, while the second convolutional layer uses 16 filters with the same kernel size. The output of the convolutional layers is flattened and passed through two fully connected layers: the first with 16 units and the second with a single unit to predict the current level for each k-mer.

The model was implemented using PyTorch, and the Adam optimizer was used for gradient descent. Training was conducted for a maximum of 100 epochs with early stopping based on validation loss to prevent overfitting. Mean squared error (MSE) was used as the loss function during training, and R-squared values were computed on the test set to evaluate model performance.

For both the 5mC prediction and XNA experiments, the same architecture and training protocol were applied. Independent training repeats (n=50 for 5mC experiments; n=10 for each XNA base-dropout experiment) were performed with different random seeds to evaluate model performance. The average R-squared value and root mean squared error (RMSE) were used as performance metrics.

4.2.2 Graph Neural Network Model

Representing k-mer chemical structures as graphs

Based on the workflow described in [39], each individual nucleotide was first represented as a simplified molecular-input line-entry system (SMILES) string. These individual SMILES strings, shown in Table 4.1, were then combined to represent full k-mer chemical structures depending on each k-mer's sequence. Specifically, SMILES strings for each nucleotide were concatenated with a "." character in between each nucleotide. While this character signifies a "non-bond" in SMILES notation, we found that this did not impact model performance compared to following the method described in [39], where an "O" was appended to the end of each concatenation to represent the unbonded hydroxyl group of the sugar phosphate backbone. We chose our approach for convenience in constructing k-mer molecular objects using the rdkit Chem library. We also explored two different sets of SMILES representations, one that includes the sugar-phosphate backbone in the structural representation and one that includes only the base structure. For the 5mC prediction

task, we found that SMILES strings including the backbone structure performed better than the base-only representation. This representation was also chosen for this dataset to be consistent with the representation in [39] that our implementation was benchmarked against. For XNA experiments, no significant difference in performance was observed between the two representations and the base-only representation was used for convenience.

K-mer SMILES strings were used to construct graphs of the form $\mathbf{G}(\mathbf{A}, \mathbf{X})$. $\mathbf{A}$ is an adjacency matrix depicting the connectivity of atoms in each k-mer where $\mathbf{A}_{i,j}$ equals 1 if and only if the i th and j th atoms are covalently bonded, with atom order determined by each SMILES string. $\mathbf{X}$ is a feature matrix where $\mathbf{X}_i$ is a feature vector storing atom level chemical properties for each atom in $\mathbf{A}$. Included chemical properties are displayed in Table 4.2 and were chosen based on the approach described in [39]. For 5mC 6-mers, adjacency matrices were padded to the maximum number of atoms of the largest k-mer in the dataset (133x133). XNA 4-mers were padded to a maximum size of (44x44). Before model training all adjacency matrices are converted to edge index format to be compatible with Pytorch Geometric.

Table 4.1: SMILES strings for individual nucleotides

Nucleotide	SMILES string	SMILES string with sugar-phosphate backbone
A	<chem>Nc1ncnc2[nH]cnc12</chem>	<chem>Nc1ncnc2c1ncn2C3CCC(COP(=O)(O)O)O3</chem>
T	<chem>Cc1c[nH]c(=O)[nH]c1=O</chem>	<chem>OP(=O)(O)OCC1OC(N2C(=O)NC(=O)C(C)=C2)CC1</chem>
G	<chem>Nc2nc1[nH]cnc1c(=O)[nH]2</chem>	<chem>OP(=O)(O)OCC1OC(N2C=NC3=C2N=C(N)NC3=O)CC1</chem>
C	<chem>Nc1cc[nH]c(=O)n1</chem>	<chem>OP(=O)(O)OCC1OC(N2C(=O)N=C(N)C=C2)CC1</chem>
S^n	<chem>Cc1c[nH]c(N)nc1=O</chem>	<chem>Cc2cn(C1CCC(COP(=O)(O)O)O1)c(N)nc2=O</chem>
S^c	<chem>Cn1ccc(N)nc1=O</chem>	<chem>Cn2cc(C1CCC(COP(=O)(O)O)O1)c(N)nc2=O</chem>
B	<chem>Nc1[nH]c(=O)nc2[nH]cnc12</chem>	<chem>Nc1[nH]c(=O)nc2c1ncn2C3CCC(COP(=O)(O)O)O3</chem>
P	<chem>Nc2nc(=O)n1cc[nH]c1n2</chem>	<chem>Nc3nc(=O)n2ccn(C1CCC(COP(=O)(O)O)O1)c2n3</chem>
Z^n	<chem>Nc1[nH]c(=O)ccc1[N+](=O)O</chem>	<chem>Nc2[nH]c(=O)c(C1CCC(COP(=O)(O)O)O1)cc2[N+](=O)O</chem>
J	<chem>Nc1nc(=O)nc2[nH]ccn12</chem>	<chem>Nc2nc(=O)nc3n(C1CCC(COP(=O)(O)O)O1)ccn23</chem>
V	<chem>Nc1ccc([N+](=O)O)c(=O)[nH]1</chem>	<chem>Nc1[nH]c(=O)c([N+](=O)O)cc1C2CCC(COP(=O)(O)O)O2</chem>
X^t	<chem>O=c2nc1[nH]ccn1c(=O)[nH]2</chem>	<chem>O=c3nc2n(C1CCC(COP(=O)(O)O)O1)ccn2c(=O)[nH]3</chem>
K^n	<chem>Nc1ccc([N+](=O)O)c(N)n1</chem>	<chem>NC(=NC(N)=C1C(OC2COP(O)(=O)O)CC2)C(=C1)[N+](O)=O</chem>
Z^a	<chem>NC(=O)c1ccc(=O)[nH]c1N</chem>	<chem>NC(=O)c2cc(C1OC(COP(=O)(O)O)C(O)C1F)c(=O)[nH]c2N</chem>

Table 4.2: Atom level features

Feature	Description
Carbon	1 if the atom is carbon, 0 otherwise (boolean)
Nitrogen	1 if the atom is nitrogen, 0 otherwise (boolean)
Oxygen	1 if the atom is oxygen, 0 otherwise (boolean)
Phosphorus	1 if the atom is phosphorus, 0 otherwise (boolean)
Fluorine	1 if the atom is fluorine, 0 otherwise (boolean)
Atom degree	Total number of covalent bonds around an atom (integer)
Implicit valence	The valence of the atom minus the valence calculated from the bond connections (integer)
Number of hydrogens	Total count of hydrogens (integer)
Aromaticity	1 if atom in an aromatic ring, 0 otherwise (boolean)

Model Architecture

Our graph-based model architecture is a hybrid neural network combining Graph Isomorphism Network (GIN) layers and Convolutional Neural Network (CNN) layers for processing graph-structured data with additional molecular features. The model begins with four GINConv layers, each equipped with a multi-layer perceptron (MLP) consisting of linear transformations and ReLU activations. These layers capture the graph’s structural information by aggregating neighborhood features.

In each GIN layer, the node features are updated through a neighborhood aggregation mechanism. The update rule for each node v at the k -th GIN layer can be expressed as:

$$h_v^{(k)} = \text{MLP}^{(k)} \left((1 + \epsilon) \cdot h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(k-1)} \right)$$

In this equation, $h_v^{(k)}$ represents the embedding of node v at the k -th layer of the network. The function $\text{MLP}^{(k)}$ is a multi-layer perceptron applied at the k -th layer, which learns the transformation of the node embeddings at that specific layer. The term $h_v^{(k-1)}$ refers to the embedding of node v from the previous layer (or the initial node features when $k=1$), while $\mathcal{N}(v)$ denotes the set of neighbors of node v . The sum $\sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}$ aggregates the features of the neighboring nodes u of node v , effectively capturing information from the local graph structure.

The parameter ϵ is learnable, allowing the model to balance the importance of a node’s own features and the aggregated features from its neighbors. This mechanism provides flexibility in how much weight is

given to a node's local structure during the feature update process. The GIN framework has demonstrated strong representational power by capturing intricate graph structures through these adaptive aggregation functions, making it well-suited for graph-level discrimination tasks. This feature aggregation and update process is carried out for each graph layer, and by stacking GIN layers, atom feature vectors contain chemical information for all neighboring atoms within a specific distance equal to the number of graph layers applied.

Following the GIN layers, the model applies a series of 1D convolutional layers, with ReLU activations, dropout for regularization, and average pooling to reduce the dimensionality. After the convolutional operations, the output is flattened and concatenated with a flattened one-hot encoded representation of the k-mer sequence. Finally, the model passes this concatenated output through two fully connected layers, leading to a single output prediction for the ionic current level mean. The architecture is designed to leverage both graph-based representations and traditional CNNs, making it suitable for tasks that require processing both topological and feature-based data.

Training

The GIN-based model was implemented using Pytorch Geometric. Mean squared error (MSE) was used as the loss function and the Adam optimizer was used for gradient descent. Models were trained for a maximum of 500 epochs with early stopping based on validation loss to prevent overfitting. Independent training repeats were performed to evaluate model performance (n=50 for 5mC experiments; n=10 for each individual XNA base-dropout experiment). Average R-squared and RMSE values across training runs were calculated as performance metrics.

4.2.3 Fine-tuned GPT-2 model with k-mer structures represented as SMILES strings

Pretrained Model and Tokenization

Our fine-tuned GPT-2 model was initialized with weights from the "gpt2" configuration provided by the HuggingFace library(<https://huggingface.co/openai-community/gpt2>). XNA k-mers were converted to SMILES string representations following the same protocol described in "representing k-mer chemical structures as graphs" above.

The SMILES strings were then tokenized using a custom tokenizer built to accommodate the specific

alphabet used in the SMILES representation. Each token corresponds to either a single chemical symbol or group of symbols in the SMILES string. For instance, the SMILES string "Nc1ncnc2[nH]cnc12" would be tokenized into meaningful components like "N", "c", "1", "[nH]", etc. The SMILES tokenization process breaks down the input sequences into smaller units to ensure the model can handle the chemical structure effectively.

Data Augmentation

To enhance the diversity of the training dataset, we employed a data augmentation strategy, generating multiple variations of each SMILES string. For each k-mer sequence, 100 different SMILES strings were created by randomizing the atom ordering while preserving the chemical structure. This augmentation significantly expanded the dataset and introduced variation that allowed the model to generalize better to unseen data. For example, from the original SMILES string "Nc1ncnc2[nH]cnc12", we generated multiple valid structural variations, each of which was tokenized and used during training.

Model Fine-tuning

The fine-tuning process involved adapting the pretrained GPT-2 model to perform regression on augmented SMILES data. The GPT-2 model, originally designed for language modeling, was modified by attaching a regression head, enabling it to predict continuous values. We initialized the regression head randomly, while the remainder of the model used the pretrained weights from GPT-2. The model was then trained on the augmented SMILES dataset using mean squared error as the loss function.

4.3 Results

4.3.1 Comparing sequence- and structure-based architectures for k-mer-level generalization and DNA 5mC prediction

We first investigated how sequence-based and structure-based learning approaches compare in predicting nanopore signals. In a recent work by Ding et al., the authors developed a graph-based neural network model that predicts ionic currents from DNA k-mer sequences by encoding the molecular structure of each

k-mer as a graph with chemical features[39]. Using a graph convolutional network (GCN) to represent atoms and bonds within each k-mer, the model leverages structural information to capture the relationship between nucleotide composition and sequencing currents. The reported model architecture incorporated GCN layers to create embeddings of the molecular graphs, CNN layers to summarize longer range chemical information above the graph layers, and a final fully connected layer that outputs a predicted ionic current level for each k-mer.

To benchmark this approach, we assessed the predictive advantage of structural information by comparing it against a simpler, structure-agnostic model. Specifically, we implemented a CNN model that relies solely on sequence-based information. Our goal was to assess whether the inclusion of molecular structure yields substantial improvement over traditional sequence-only approaches.

Ding et al. first tested their GCN-based framework using the canonical DNA 6mer model provided by ONT, which contains all DNA 6mer sequences ($4^6=4096$ k-mers). They randomly partitioned the dataset into train-test splits, simulating a scenario where the model must infer missing k-mers from limited data. After 50-fold cross-validation for each split, the authors reported that performance stabilized once more than 40% of 6-mers were included in training, using root mean square error (RMSE) and Pearson’s correlation (r) as metrics. For 6-mers only in the test set, the model achieved an average RMSE of 1 and a Pearson’s correlation of 0.995.

To compare these results to a sequence-only CNN model, we first converted each DNA 6-mer into a one-hot encoded representation of its sequence. Sequence matrices were flattened and fed into the model, which consists of two 1D-convolutional layers followed by two fully connected layers that output an ionic current prediction for each k-mer. We then performed 50-fold cross validation for each train-test split, dividing the dataset in increments of 5% as in the reported GCN experiment. The average RMSE and Pearson’s correlation across two independent runs were 0.997 and 1 respectively (see Table A.1), performing comparably to the structure-based GCN model in Ding et al.

Ding et al. also explored performance of the graph-based model on predicting 5-methylcytosine (5mC)-containing DNA 6-mers. First, they trained the model using all possible canonical DNA 6-mers plus {1%, 5%, 10%, 30%, 50%, 70%, 90%} of randomly selected 5mC-containing 6-mers. They reported decent goodness of fit when 5% of 5mC-containing 6-mers were included in the training data, with an average

RMSE and Pearson’s correlation of 1.2 and 0.995, respectively. We replicated this experiment using our CNN model, again performing 50 independent repeats for each training partition (see Table A.2). When 5% of the 5mC-containing 6-mers were included in our training set, the average RMSE and Pearson’s correlation reached 1.1 and 0.996, respectively. When only 1% of the 5mC data was used for training, the average RMSE and Pearson’s correlation were 1.5 and 0.991. Again, these results were comparable to the GCN performance.

A key area of interest in predictive sequencing models is *de novo* prediction of previously unseen modifications (zero-shot learning). The authors explored the GCN’s performance when trained on all canonical DNA 6-mers and tested on 5mC-containing DNA 6-mers. They found that while the goodness of fit for the 5mC-containing 6-mers was significantly worse than for the canonical 6-mers, the model was able to reach an average RMSE and Pearson’s correlation of 1.8 and 0.993, respectively. Despite the comparable performance of our CNN model and the reported GCN model in 5mC prediction when even a small number of 5mC-containing examples are included in the training data (few-shot learning), we found that the CNN model performed significantly worse in *de novo* zero-shot 5mC prediction cases (average RMSE of 21.2 and Pearson’s correlation of 0.632).

To replicate this, we developed a similar graph-based architecture to the reported model, replacing the GCN layers with Graph Isomorphism Network (GIN) layers. The GIN architecture has demonstrated robust discriminative power for graph level regression tasks compared to other popular GNN architectures such as GCN and GraphSAGE due to a more expressive aggregation mechanism[49]. K-mers were first converted into SMILES strings by combining the individual base SMILES strings shown in Table 4.1. The full k-mer SMILES string was then used to generate an adjacency matrix showing the connectivity of each atom in the molecule with atoms as nodes and covalent bonds as edges, as well as a feature matrix describing the chemical properties for each atom (see 4.2). After several GIN layers, each node’s feature vector contains detailed chemical information within a specific graph distance (equal to the number of graph layers applied). As in the original approach, we subsequently applied CNN layers to summarize longer-range chemical interactions, followed by fully connected layers to predict the k-mer ionic current.

We repeated the *de novo* 5mC prediction experiment with our GIN model and achieved an average RMSE of 1.3 and a Pearson’s correlation of 0.995. This finding corroborates Ding et al.’s model performance

and suggests that while a considerable amount of signal information is contained in the sequence context alone—as evidenced by the CNN model’s performance—incorporating structural and chemical information can significantly enhance model performance and generalization. Models that capture more complex relationships between k-mers and their resultant sequencing current levels have the potential to enable accurate *de novo* predictions for previously unseen modifications. This capability makes them practically valuable by reducing the need for exhaustive experimental measurements for every new modification or base under study.

4.3.2 Generating XNA k-mer tables from machine learning models

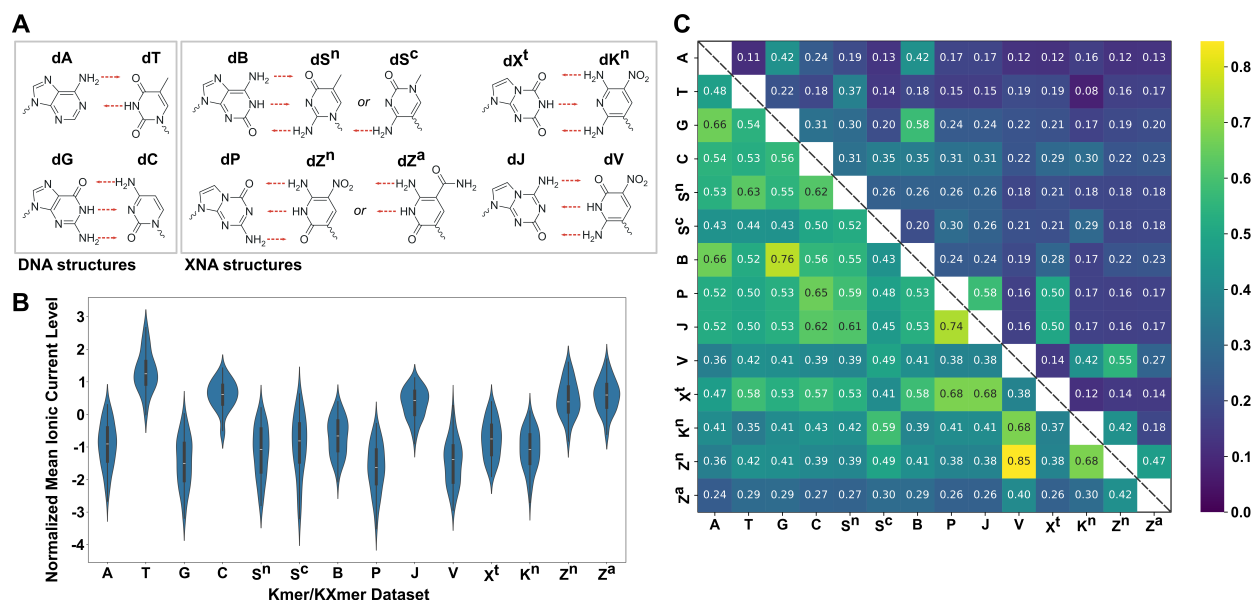


Figure 4.2: **A** Structures of canonical DNA nucleobases and xenonucleobases. Arrows indicate hydrogen bonding between base pairs, drawn in the direction of donor-to-acceptor. S and Z nucleobases have two possible structures which base pair with B (N-nucleoside: S^n and C-nucleoside S^c) and P (C-nucleoside: Z^n and C-nucleoside Z^a), respectively. Z^n has a nitro functional group while Z^a has an amide functional group as well as a 2'-fluoro ribose modification. **B** Distribution of normalized mean nanopore ionic current levels for canonical DNA and XNA nucleobases. Distributions are centered with each specific nucleobase in the 0th position (-1, 0, +1, +2) in measured k-mer levels. **C** Tanimoto similarity matrix of canonical DNA and XNA chemical structures. The lower-left triangle of the matrix shows similarity scores calculated using the full nucleotide/xenonucleotide structure, including the sugar-phosphate backbone. The upper-right triangle shows similarity scores calculated using only the nucleobase/xenonucleobase structures. The Tanimoto coefficient quantifies structural similarity between two molecules, with a value of 1 indicating identical structures and a value of 0 indicating no shared features.

Building on the promising results from the 5mC prediction task, we further explored structure-based architectures for predicting nanopore sequencing currents in DNA sequences containing single XNA substitutions. Specifically, we evaluated the performance of the baseline CNN model, our GIN model, and a fine-tuned GPT-2 model using an XNA dataset comprising ten different XNA bases from Kawabe et al.[30]. An overview of the dataset is shown in Figure 4.2, and model architectures are depicted in Figure 4.3.

For each XNA base, nanopore signal measurements were collected for all 256 possible 4-nt k-mers containing a single XNA substitution ($4^4 = 256$ k-mers per XNA). Using k-mers as inputs and their associated normalized mean current levels as target values, we individually evaluated the three models’ ability to predict sequencing current levels. To assess generalization to unseen XNA structures, we conducted base-dropout experiments, where models were trained on all XNA and canonical DNA datasets except one held-out XNA type, which was used for testing. This approach allowed us to evaluate how well each model could infer current levels for previously unseen XNA substitutions based on learned sequence and structural representations.

Each base-dropout experiment generated a model-predicted k-mer table for the held-out XNA k-mer set. Model performance was assessed by computing R^2 values for predicted versus true (measured) current levels. The average R^2 values for each XNA base and model are shown in Figure 4.4A, calculated across ten independent base-dropout runs per XNA base. The sequence-only CNN model achieved an average R^2 of 0.85 across all ten XNA bases. Scatter plots for individual XNA bases are shown in Figure 4.4B and Figures A.11 to A.21.

Next, we examined the effect of incorporating XNA k-mer chemical structures using the GIN model. In this case, the average R^2 across all XNA bases was 0.84. Given the strong performance of the CNN model, we hypothesized that incorporating sequence context into the GIN model could improve predictions. To test this, we appended one-hot encoded k-mer sequence representations (as used in the CNN) to the output of the GIN and CNN layers before passing them to the final fully connected layers. This modification allowed the model to leverage both sequence and structural information. Including sequence context improved the GIN model’s average R^2 to 0.88.

We also explored representing k-mer structures as 1D SMILES strings rather than 2D molecular graphs. Given the success of pre-trained transformer models in molecular property prediction, we fine-tuned a GPT-

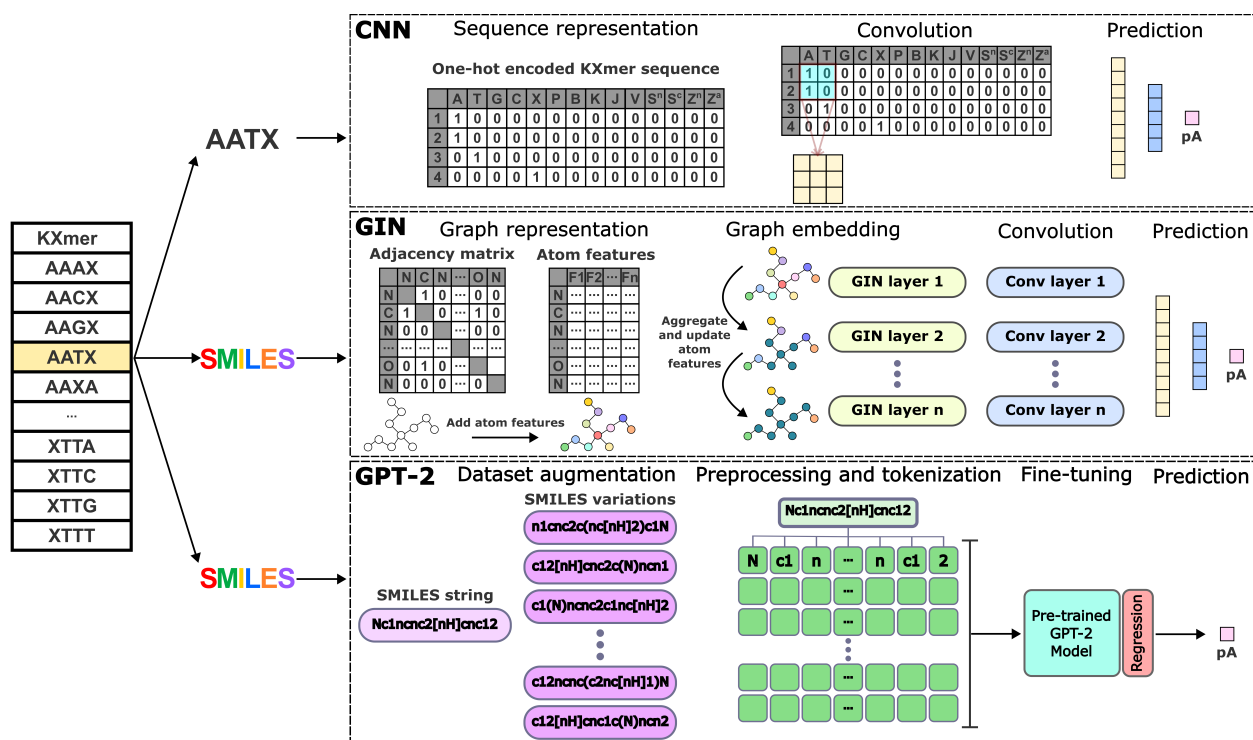


Figure 4.3: Overview of model architectures. **CNN** KXmer sequences are first converted into one-hot encoded representations. Standard convolution layers are applied, followed by fully connected layers that output a current level prediction for each KXmer. **GIN** Sequences are converted into SMILES strings representing KXmer chemical structures. Each SMILES string is used to generate an adjacency matrix showing atom connectivity, along with a feature matrix encoding chemical properties of each atom. GIN layers are applied to aggregate and update the node (atom) features with the guidance of the adjacency matrix. Each atom’s features are summed with its neighboring atoms’ features, and the aggregated features are passed through an MLP layer that learns a transformed representation of the features. The atom features are then updated. Atoms within a graph distance equal to the number of GIN layers are aggregated and updated following this process. Due to the relatively small encoding distance of the GIN layers (1-hop per layer), standard CNN layers are subsequently applied to summarize longer-range chemical information. Finally, fully connected layers output a current level prediction for each KXmer. **GPT-2** Sequences are converted into SMILES strings representing KXmer chemical structures. To generate more diverse training data, data augmentation is applied where for each SMILES string, 100 variations are created by shuffling the atom order while preserving the encoded chemical structure. SMILES strings are tokenized at the atom level to create a vocabulary suitable for the model. A pre-trained GPT-2 model is fine-tuned on the SMILES data, and a regression head is added to output a current level prediction for each KXmer.

2 model to predict XNA sequencing currents. Each k-mer was first converted into a SMILES string encoding its molecular structure. To address the limited dataset size (256 examples per XNA base), we employed data augmentation by generating 100 randomized SMILES variants per k-mer, preserving molecular structure while diversifying textual representations[50]. These SMILES strings were tokenized at the atom level

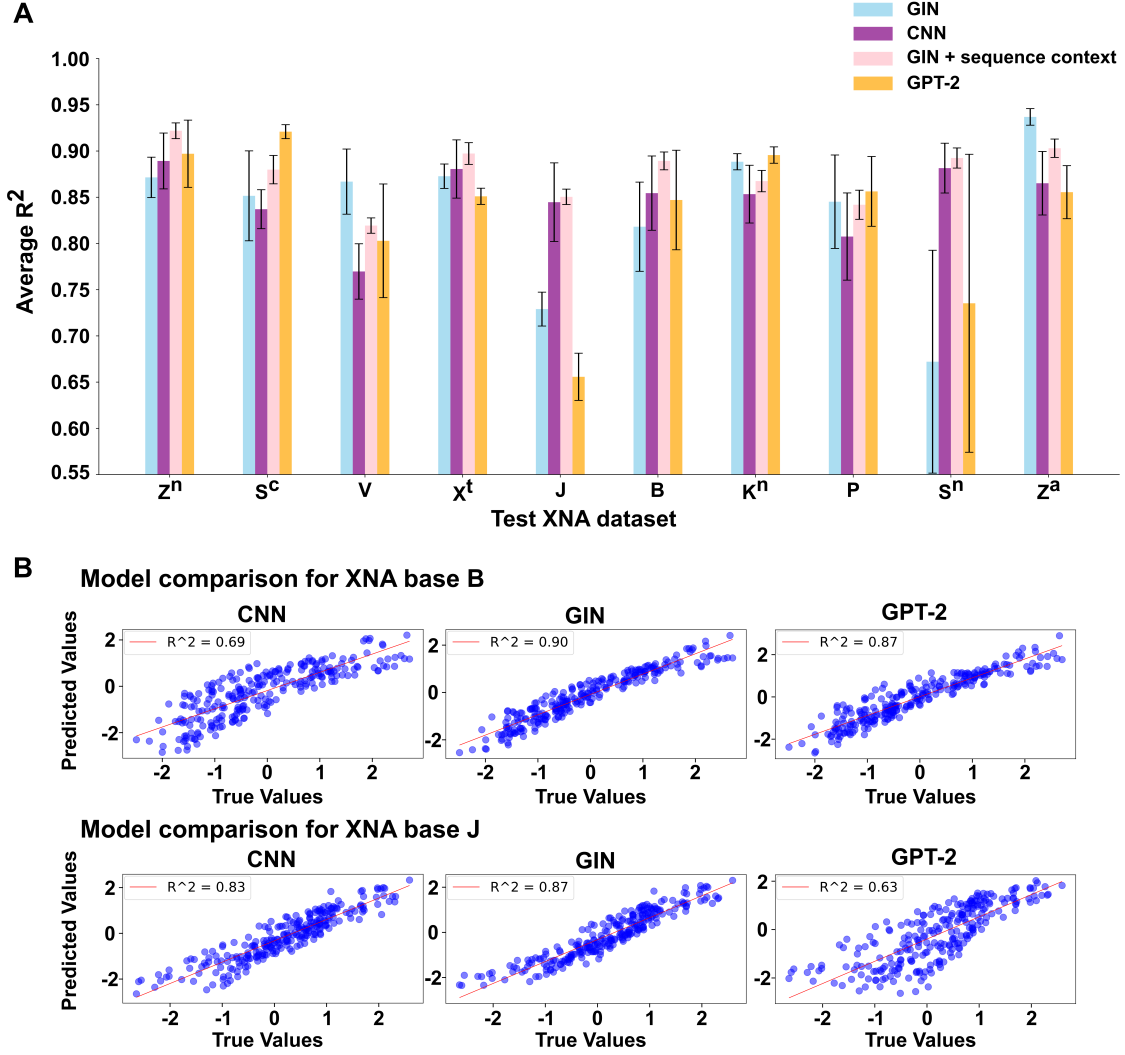


Figure 4.4: Comparison of model performance on *de novo* XNA prediction. **A** Average R^2 values achieved by each model with $n=10$ independent training runs per base. Models were trained on all other canonical DNA and XNA datasets and tested on one held-out XNA dataset at a time. **B** Example scatter plots showing model predicted k-mer characteristic ionic current levels vs measured values. Scatter plots for other XNA bases are available in Figs. Figures A.11 to A.21).

using a custom tokenizer, and the model was fine-tuned on this data to learn a domain-specific vocabulary. Following fine-tuning, we added a regression layer to predict current levels. As with the other models, we conducted base-dropout experiments, training on all but one XNA dataset and testing on the held-out set. This model achieved an average R^2 of 0.83 across all XNA bases.

While model performance differences were less pronounced in the XNA task compared to 5mC prediction task, the available training data for the XNA task was significantly smaller. Nevertheless, the GIN

model with sequence context outperformed both the CNN and the sequence-agnostic GIN model on average. Our results highlight the importance of sequence context in determining sequencing current signals while demonstrating that structural and chemical information can enhance predictive power. With larger datasets, we hypothesize that these effects may become more pronounced.

4.3.3 Integrating model-generated k-mer tables in an XNA basecalling model

To assess how the XNA-trained models perform in a sequencing context, we integrated model-generated k-mer tables into a basecaller for single xenonucleotide substitution sequencing. In the work by Kawabe et al., the authors used empirically measured sequencing current levels to generate k-mer tables for basecalling. Using measured 4-nt k-mer models, the authors demonstrate basecalling of heptamer libraries containing single XNA substitutions (NNNNNNNN, N = modified nucleotide). During basecalling, heptamer sequences are broken down into their constitutive k-mers (e.g. GTAXCAC $\rightarrow [\mu_{GTAX}, \mu_{TAXC}, \mu_{AXCA}, \mu_{XCAC}]$) and the level means of each 4-nt k-mer are used to model current transitions. The probability that a given 4-nt k-mer sequence will produce an ionic current signal is modeled as a normal distribution, and for any observed set of signals, the modeled probability density function is used to calculate the likelihood that an observed set of signal levels was emitted from a particular sequence.

For each of our three models, we generated k-mer tables with predicted current level means for each XNA dataset. These k-mer tables were then integrated into the basecalling model in place of the measured k-mer tables to perform benchmarking. It is important to note that the measured k-mer models have an inherent advantage over model-generated k-mer tables, as parts of the heptamer datasets used for sequencing are also used in generating the measured 4-mer models. With this in mind, we evaluated the performance of each model by focusing on three main metrics: root mean square deviation (RMSD) between each model-generated k-mer table and the measured k-mer levels, per-read basecalling accuracy, and area under the curve (AUC) scores calculated from receiver operating characteristic (ROC) curves. Furthermore, to assess the generalization power of each model, we examined two different methods for generating k-mer tables for each model: a *de novo* prediction case where no examples of the specific XNA base are included in the model training data (zero-shot learning), and a case where 5% of the specific XNA dataset is added during the training process (few-shot learning).

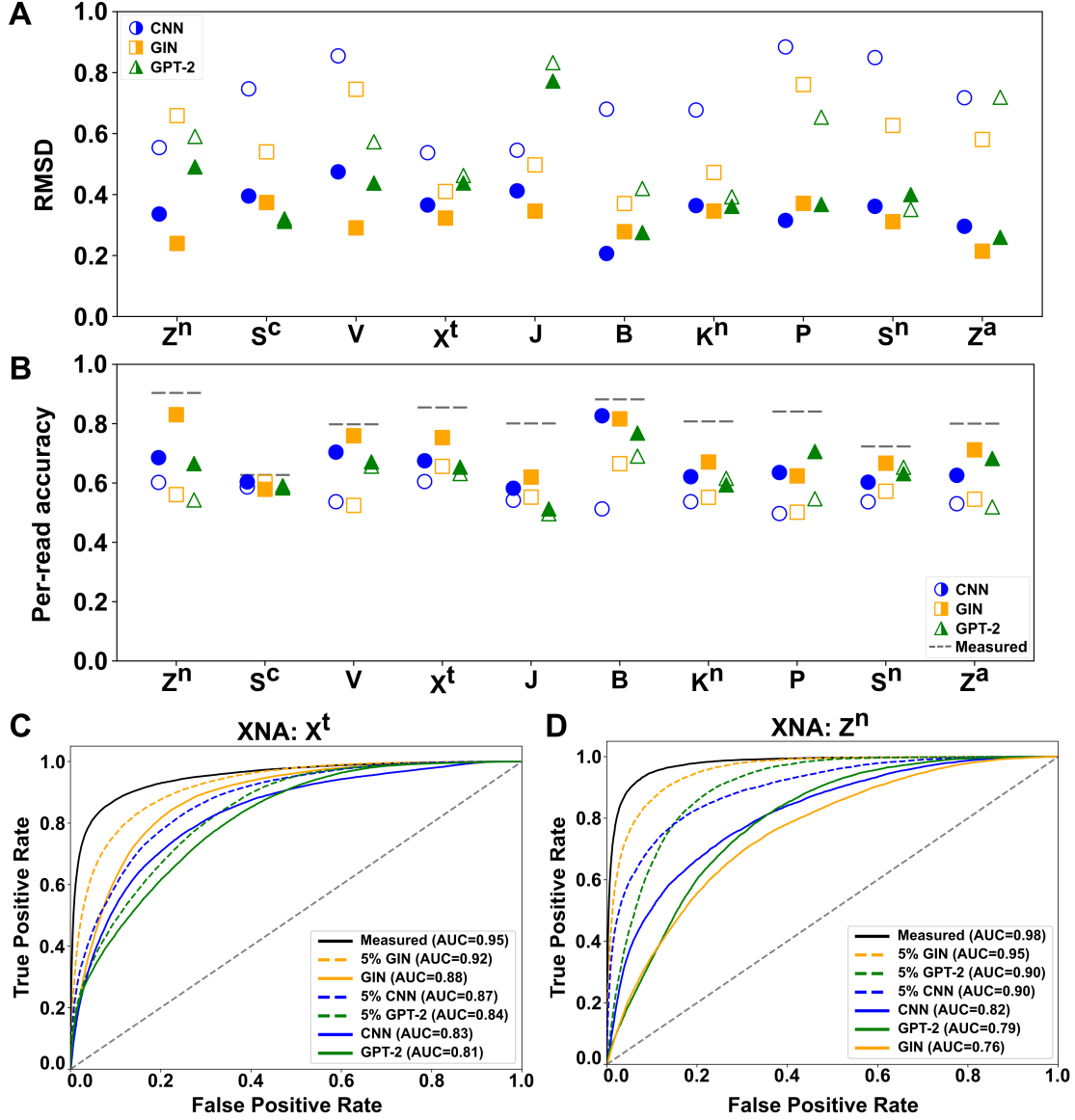


Figure 4.5: **A** Root mean square deviation (RMSD) between model-generated k-mer characteristic ionic current values and empirically measured k-mers. **B** Comparison of per-read accuracy in XNA basecalling using model-generated k-mer tables and empirically measured k-mer tables. **C-D** Example receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (filled markers) were trained with 5% of the held-out XNA dataset added to the training data (few-shot learning).

To quantify how closely the predicted current levels matched experimentally determined values, we calculated the RMSD between the predicted and measured current values (Figure 4.5A). Lower RMSD values indicate better agreement between predicted and true current levels. Among the three models, the GIN

model consistently outperformed the other two in the zero-shot task, exhibiting the lowest RMSD values across most XNA bases (0.37 to 0.76, avg 0.57). The GPT-2 model showed a wider range of performance, with some bases performing nearly as well as GIN but others displaying higher deviation (0.32 to 0.83, avg 0.53). The CNN model had the highest RMSD values overall (0.54 to 0.88, avg 0.70), indicating that it struggled the most to approximate the measured currents. Importantly, few-shot learning improved RMSD values across all models, suggesting that even limited exposure to the target XNA base significantly enhances predictive accuracy.

We next assessed how well each model preserved sequencing accuracy when its k-mer tables were used in place of measured values in the basecalling model (Figure 4.5B). While measured k-mer tables unsurprisingly led to the highest per-read sequencing accuracy (0.63 to 0.90, avg 0.80), the model-generated tables exhibited varying degrees of performance. The GIN and GPT-2 models performed comparably in this metric, with GIN achieving per-read accuracy values between 0.50 and 0.67 (avg 0.57), and GPT-2 showing a slightly broader range from 0.50 to 0.69 (avg 0.59). The CNN model again underperformed relative to the other two, with accuracy values between 0.50 and 0.60 (avg 0.55). This aligns with the RMSD results, where the CNN model's poorer current predictions translated to reduced sequencing accuracy. As observed with RMSD, few-shot learning led to noticeable improvements across all models, with the GIN performing over 10% higher accuracy in the Zn sequencing task. This reinforces both the idea that partial exposure to the target XNA base can enhance model-generated predictions, and that the structure based models are more capable of generalization.

To evaluate the models' ability to distinguish correct basecalls from errors, we analyzed ROC curves and calculated AUC scores. Two example ROC curves are shown in Figure 4.5C and 4.5D, with curves for the remaining XNA bases available in Figures A.1 to A.10. Measured k-mer tables yielded the highest AUC values, ranging from 0.79 for the worst-performing base (S^c) to 0.98 for the best-performing base (Z^n), with an average of 0.92. Among the models, the CNN exhibited the best AUC scores on average (0.64 to 0.96, avg 0.82), suggesting that despite its poorer RMSD and per-read accuracy, its predicted k-mer tables still preserved some ability to distinguish correct from incorrect basecalls. The GIN model followed closely behind (0.68 to 0.90, avg 0.78), while GPT-2 showed the most variability, with AUC scores ranging from 0.41 to 0.86 (avg 0.74). The broader spread in GPT-2's AUC values suggests that its performance is

more dependent on the specific XNA base being tested. As seen in the other metrics, incorporating 5% of the held-out base improved AUC scores across nearly all models, further emphasizing the benefit of even limited k-mer contexts.

Overall, the GIN model exhibited the best balance of performance across all metrics, with the lowest RMSD values and competitive per-read accuracy and AUC scores. The GPT-2 model, while comparable to GIN in some cases, displayed more variability, with some bases performing well and others lagging behind. The CNN model, despite achieving the highest AUC scores in some cases, had the highest RMSD and lowest per-read sequencing accuracy, indicating that while it could still differentiate correct from incorrect calls, its current predictions were the least accurate overall. Across all models, few-shot learning consistently improved performance, highlighting the impact of even partial exposure to the target XNA dataset. This suggests that the need to exhaustively measure every possible k-mer for a given XNA base in every sequencing context is not required to achieve competitive performance across all k-mer contexts. Example current traces showing each model’s predicted mean current levels overlaying measured current levels are shown in Figures A.22 to A.24.

In summary, these results demonstrate that while measured k-mer tables still provide the most accurate sequencing performance, model-generated tables represent a viable alternative, particularly when some training data from the target XNA base is available. The relative strengths and weaknesses of each model suggest different trade-offs: GIN offers the most consistent accuracy, CNN provides strong classification performance despite higher deviation, and GPT-2 has potential but is more sensitive to dataset composition.

4.4 Discussion

Advancing sequencing technology is essential for unlocking the full potential of expanded genetic alphabets. In this chapter, we evaluated three machine learning approaches for predicting nanopore ionic currents from DNA sequences containing single XNA substitutions: a sequence-based CNN, a GIN model that encodes k-mer molecular graphs with chemical features, and a GPT-2 model fine-tuned on SMILES representations.

Nanopore signals are influenced by sequence context, molecular structure, and interactions within the pore environment[39, 38]. Traditional k-mer models, built empirically from sequence data alone, capture much of this variability but neglect the underlying chemistry and structure that shape ionic currents. Our re-

sults show that sequence information alone explains a large fraction of signal variance, yet adding structural features can improve performance. Hybrid approaches, such as combining sequence and molecular graph features, yielded the best results: the GIN model achieved the highest average R^2 across all XNA bases, highlighting the promise of structure-informed modeling.

A key limitation of our study is dataset size and diversity. The ten XNAs examined represent only a small subset of possible modifications, limiting our ability to fully assess how structural features generalize. Expanding datasets to cover more chemical diversity will be essential. Future work should focus not on “sequence vs. structure,” but on quantifying how structural representations complement sequence features, and under what conditions they most improve performance.

Although our experiments used R9.4.1 flow cells, the modeling strategy is broadly applicable to newer chemistries. Challenges such as predicting currents for nonstandard bases persist across pore generations. Even state-of-the-art pipelines like Remora still use k-mer-segmented labels for modified base training, underscoring the ongoing relevance of k-mer-level modeling[41, 28]. Our framework is therefore complementary to end-to-end approaches like Dorado, which require large, diverse datasets[45]. In contrast, structure-informed k-mer prediction can bootstrap signal expectations for new chemistries, enabling early model development before extensive training data are available.

While nanopore sequencing uniquely enables direct, amplification-free detection of modified bases, it faces challenges such as noisy signals that can obscure subtle modifications, signal overlap between similar modifications and canonical bases, and reliance on training data from known contexts which can limit generalization[51, 52, 53]. Compared to some established methods (e.g. enzymatic transliteration, mass spectrometry), nanopore sequencing can exhibit lower sensitivity or higher false-positive and false-negative rates, particularly in single-read settings.

Sequencing accuracy requirements differ across applications. Consensus accuracy improves with read depth, since random errors average out, but systematic errors can persist even at high coverage[29]. Some applications tolerate noisy single reads, provided coverage is high. Aptamer evolution[9, 54, 55], for example, can rely on consensus sequences from enriched library members. Similarly, consensus-based detection suffices for diagnostics with modest sensitivity[56, 57], monitoring whether engineered organisms retain modified bases[58, 59], or performing QC of synthetic constructs. Other applications require

highly accurate single-read calls. Examples include polymerase fidelity quantification at the single-base level[60, 61, 62, 63, 64], digital DNA data storage[65, 66] (e.g., ATGC = 2 bits vs. ATGCXYWZ = 3 bits), barcode readout in expanded alphabets, and QC of libraries without deep consensus[67, 68].

The models presented here fall into the first, consensus-driven tier. They do not yet reach the per-read accuracy needed for quantitative single-molecule tasks or fully replace measured k-mer tables. However, they performed under stringent conditions: limited datasets, restricted k-mer size ($4^4 = 256$ per modified base), and zero- or few-shot evaluation. Despite these challenges, the models generalized to unseen bases, indicating that performance is not fundamentally constrained by the method.

A central motivation of this work is scalability. Measuring every k-mer context for each new base is prohibitively time- and cost-intensive. In contrast, predictive modeling dramatically reduces data requirements, enabling rapid prototyping for new chemistries. While accuracy improvements are needed before single-molecule applications are feasible, our results outline a generalizable and resource-efficient path toward that goal.

There are also clear routes for improvement. Transfer learning has proven effective for both GNNs and transformers: models pretrained on large molecular datasets (e.g., ChemBERTa [69, 70], pretrained GNNs[71]) can be fine-tuned to downstream tasks, improving performance in low-data regimes. Applying similar strategies to our models could substantially boost predictive power for poorly represented modifications.

Beyond GPT-2, larger and more advanced LLMs (GPT-3/4, LLaMA, Mistral/Mixtral, Qwen) offer additional promise[72, 70]. These models excel in zero- and few-shot settings, where they generalize with minimal task-specific data, and recent work shows they can learn chemically meaningful embeddings from SMILES[73, 74]. Although originally designed for language tasks, LLMs are increasingly applied to molecular property prediction[75], raising the possibility that they could transfer knowledge across novel nucleotide chemistries. Many questions remain, particularly regarding pretraining transferability to nucleic acid-like polymers, but this is a promising direction.

Our experiments also highlight the practical value of few-shot learning. While true *de novo* prediction remains a long-term goal, we find that limited exposure to new XNA data substantially improves performance, reducing experimental burden while retaining accuracy. This was evident for both 5mC and XNA

tasks, where models trained on small subsets generalized effectively. Such few-shot strategies will be crucial for scaling nanopore sequencing to expanded alphabets.

Finally, this work points to future opportunities. Incorporating higher-order sequence dependencies, leveraging 3D structural information, or developing multimodal models that integrate nanopore traces with sequence and structural features could all further improve performance. As datasets expand in size and modality, we anticipate that machine learning models capable of directly inferring nanopore signals from molecular features will play a central role in enabling sequencing of expanded genetic alphabets and other synthetic biopolymers.

Chapter 5

Extending Structure-Informed Models to Nanopore Protein Sequencing

5.1 Introduction

The structure-informed modeling framework developed in Chapter 4 demonstrated that incorporating molecular-level information beyond sequence context can significantly improve nanopore current prediction for XNA-containing DNA. These promising results raised a natural question: could similar approaches extend to the even more challenging domain of nanopore protein sequencing?

Protein sequencing presents substantially greater complexity than nucleic acid sequencing[76, 77]. While DNA consists of four chemically similar bases differing primarily in hydrogen bonding patterns, proteins comprise twenty structurally diverse amino acids with widely varying sizes, charges, hydrophobicities, and functional groups. This chemical diversity creates a vastly larger signal space that models must learn to navigate. Moreover, proteins exist not merely as linear sequences but as dynamic molecules subject to extensive post-translational modifications that create additional layers of proteoform diversity[13]. The ability to detect and map these modifications, particularly phosphorylation, which plays critical regulatory roles in cellular signaling, would represent a significant advance for single-molecule proteomics.

Recent work from our laboratory established a method for nanopore protein sequencing using ClpX unfoldase to processively translocate proteins through commercial CsgG nanopores[14]. This approach en-

abled the development of the PASTOR (Proteins for Amino acid Sequencing Through Optimized Regions) dataset, which provided the first systematic measurements of amino acid-specific nanopore signals in controlled sequence contexts. The PASTOR constructs, combined with the demonstrated sensitivity to single amino acid substitutions and post-translational modifications, created an opportunity to explore whether structure-informed computational models could enhance protein sequencing accuracy and generalization.

This chapter describes my efforts to extend the modeling approaches from Chapter 4 to the protein domain. I focus specifically on phosphorylation detection, an important post-translational modification with broad implications for understanding cellular regulation and disease states. After introducing the experimental method and PASTOR dataset, I describe adaptations to the CNN and GIN architectures necessitated by the smaller dataset size and different molecular representation requirements. I then present results demonstrating that structure-informed models, particularly the GIN architecture, achieve superior generalization in zero-shot and few-shot phosphorylation detection tasks. Finally, I examine model-predicted signals for specific phosphorylated sequences and discuss what these results reveal about the potential and current limitations of computational approaches to nanopore protein sequencing.

5.2 Nanopore Protein Sequencing: Challenges and Methods

5.2.1 Obstacles in Protein Translocation

The nanopore sequencing principles that enable DNA and RNA analysis face fundamental challenges when applied to proteins. DNA’s negatively charged phosphate backbone facilitates electrophoretic capture and translocation through nanopores under applied voltage. Proteins, by contrast, have electrically neutral polypeptide backbones with charge determined by amino acid side chains. At physiological pH, proteins can be positively charged, negatively charged, or neutral depending on their amino acid composition, making electrophoretic capture unreliable for arbitrary protein sequences.

Protein structural complexity compounds these challenges. While DNA maintains a relatively uniform double helix, proteins form diverse secondary and tertiary structures that resist translocation and can block pores entirely. The chemical diversity of amino acids further increases the difficulty: distinguishing among twenty amino acids spanning a wide range of volumes, charges, polarities, and chemical functionalities cre-

ates a substantially more complex signal space than the four-base DNA alphabet. Furthermore, the nanopore sensing region typically encompasses around twenty amino acids simultaneously, meaning the measured current at any moment reflects contributions from multiple residues. Disentangling these overlapping contributions to achieve single-amino-acid resolution represents a significant analytical challenge.

5.2.2 ClpX-Mediated Translocation

To address the translocation challenge, our laboratory developed a two-stage approach using the ClpX unfoldase enzyme[14]. Protein substrates are designed with a negatively charged N-terminal tail that enables electrophoretic capture into CsgG nanopores (cis-to-trans direction). A stably folded domain near the C-terminus prevents complete translocation through the pore, while an ssrA recognition tag remains accessible in the cis compartment. Addition of ClpX to the cis solution allows the enzyme to bind the ssrA tag and processively pull the protein back out of the nanopore (trans-to-cis direction) using ATP hydrolysis.

This ClpX-mediated translocation occurs at a much slower and more controlled rate than electrophoretic threading, enabling measurement of ionic current changes as the protein passes through the sensing region. Importantly, structural analysis revealed that ClpX moves proteins through the pore in discrete two-amino-acid steps. This stepping behavior provides a natural framework for understanding signal generation: each step advances the protein by two residues, shifting which amino acids occupy the nanopore sensing region and thereby changing the measured current. The stepwise translocation, combined with the approximately twenty-residue sensing window, means that current levels reflect a weighted contribution from multiple amino acids, with residues near the center of the pore contributing most strongly.

5.2.3 The PASTOR Dataset

To characterize amino acid-specific signals systematically, the PASTOR protein constructs were designed with strategic sequence architecture. Each PASTOR contains five repeating blocks of 59 amino acids built from a base sequence of glycine, serine, aspartic acid, and glutamic acid. A single unique amino acid mutation occupies the central position of each block, and double tyrosine mutations demarcate block boundaries. The tyrosine pairs create distinctive signal features (characteristic dips in the ionic current) that serve as landmarks for segmenting the signal into analyzable regions.

This design addresses the overlapping contribution problem by spacing mutations far enough apart (approximately 29 amino acids between block centers) so that each mutation’s signal can be isolated without interference from neighboring mutations. Eight different PASTOR variants were synthesized, each containing a different sequence of five mutations, enabling measurement of all twenty amino acids with each appearing in two different sequence contexts.

Analysis of the PASTOR signals revealed clear relationships between amino acid properties and measured currents. For uncharged amino acids, current levels showed negative correlation with amino acid volume, consistent with a volume-exclusion model where larger residues block more ionic flow. Charged amino acids deviated from this pattern: positively charged residues (lysine and arginine) decreased current below the baseline sequence, while negatively charged residues (aspartate and glutamate) substantially increased current. These charge effects likely reflect both electrostatic interactions with the applied voltage field and changes in protein strand conformation as charged residues are pulled through the pore[14, 78].

5.3 Model Architectures and Training

5.3.1 Dataset Composition

The modeling work in this chapter utilized two primary k-mer level datasets extracted from the PASTOR experiments. The canonical dataset consisted of 1,497 25-mer peptide sequences with their corresponding normalized mean current levels, representing unmodified amino acid combinations. The phosphorylation dataset contained 235 25-mers with one, two, or three phosphorylated residues per sequence, again labeled with normalized current values. This substantial imbalance between canonical and phosphorylated examples, combined with the inherent noise in extracting clean k-mer signals from raw traces, presented significant challenges for model development.

5.3.2 Model Architectures

I adapted the CNN and GIN architectures from Chapter 4 for peptide analysis. Transformer architectures including GPT-2 and Mistral were initially included in benchmarking, but are omitted from this chapter primarily due to dataset size limitations that led to poor performance in testing.

The sequence-only CNN model served as a baseline. Input peptides were represented as 25-mers, where each residue was one-hot encoded into a 21-dimensional vector (accounting for the 20 canonical amino acids plus phosphoserine). While the CNN effectively captures local sequence motifs and positional dependencies within the k-mer window, it relies entirely on the training data to infer the underlying physical properties of the amino acids from their one-hot representations.

Table 5.1: SMILES strings for amino acids and phosphoserine

Amino Acid	SMILES string
Ala	<chem>N[C@@H](C)C(=O)O</chem>
Arg	<chem>N[C@@H](CCCNC(=[NH2+])N)C(=O)O</chem>
Asn	<chem>N[C@@H](CC(=O)N)C(=O)O</chem>
Asp	<chem>N[C@@H](CC(=O)[O-])C(=O)O</chem>
Cys	<chem>N[C@@H](CS)C(=O)O</chem>
Glu	<chem>N[C@@H](CCC(=O)[O-])C(=O)O</chem>
Gln	<chem>N[C@@H](CCC(=O)N)C(=O)O</chem>
Gly	<chem>NCC(=O)O</chem>
His	<chem>N[C@@H](CC1=CN=CN1)C(=O)O</chem>
Ile	<chem>N[C@@H](C(C)CC)C(=O)O</chem>
Leu	<chem>N[C@@H](CC(C)C)C(=O)O</chem>
Lys	<chem>N[C@@H](CCCC[NH3+])C(=O)O</chem>
Met	<chem>N[C@@H](CCSC)C(=O)O</chem>
Phe	<chem>N[C@@H](CC1=CC=CC=C1)C(=O)O</chem>
Pro	<chem>N1CCC[C@H]1C(=O)O</chem>
Ser	<chem>N[C@@H](CO)C(=O)O</chem>
Thr	<chem>N[C@@H](C(O)C)C(=O)O</chem>
Trp	<chem>N[C@@H](CC1=CNC2=CC=CC=C12)C(=O)O</chem>
Tyr	<chem>N[C@@H](CC1=CC=C(O)C=C1)C(=O)O</chem>
Val	<chem>N[C@@H](C(C)C)C(=O)O</chem>
pSer	<chem>N[C@@H](CO[P](=O)([O-])[O-])C(=O)O</chem>

The Graph Isomorphism Network (GIN) followed the workflow established in Chapter 4. For each 25-mer input peptide sequence, a k-mer level SMILES representation is constructed by concatenating SMILES strings from a dictionary of residue-level representations. This SMILES representation is then used to construct a 2D molecular graph that encodes the full sequence of amino acids including peptide bonds and the backbone structure, with nodes representing atoms (annotated with chemical feature vectors) and edges representing covalent bonds. Individual amino acid SMILES strings are shown in Table 5.1. Figure 5.1 shows the Tanimoto similarity matrix computed from these structures, revealing the chemical relationships

the GIN can leverage during learning.

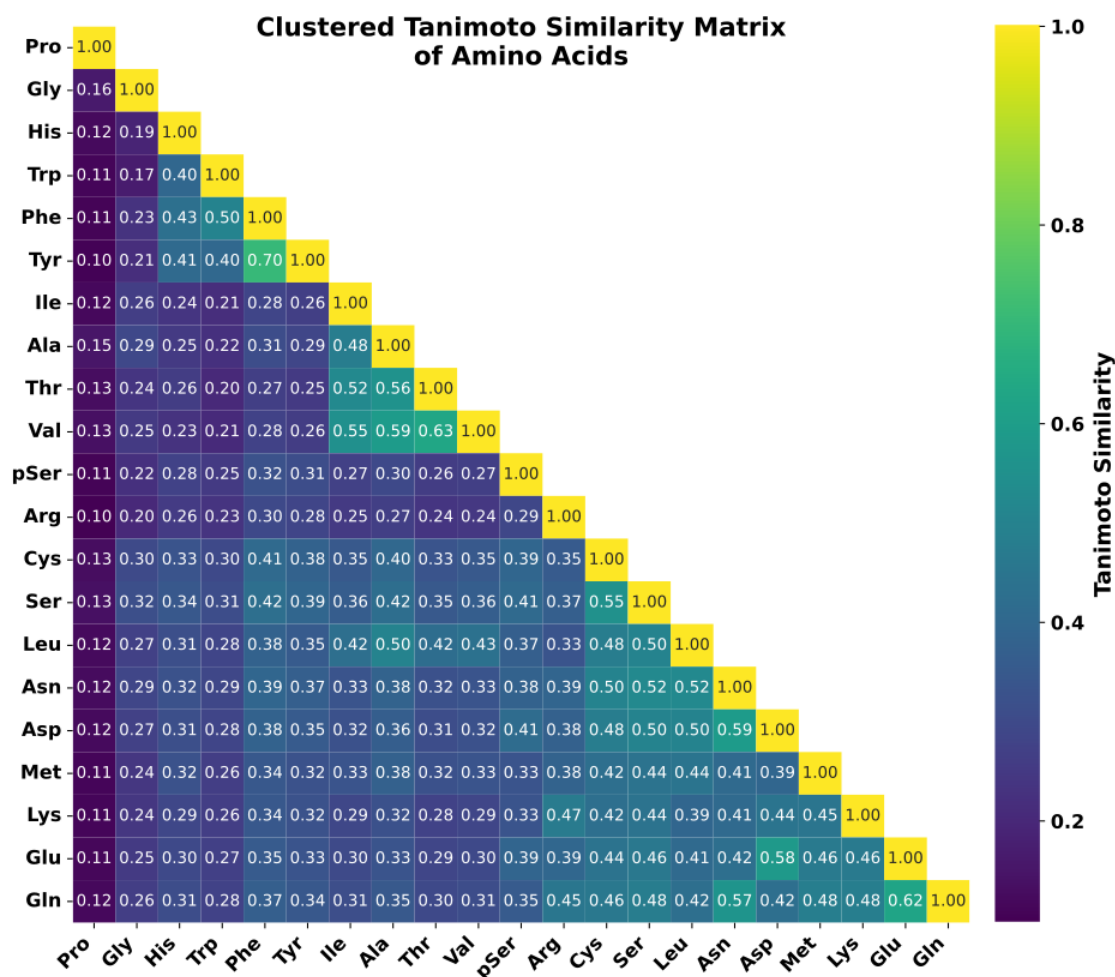


Figure 5.1: Tanimoto similarity matrix for individual amino acids. A value of 0 indicates no structural similarity between amino acids, while a value of 1 indicates an exact structural match.

The GIN architecture utilized multiple graph convolution layers augmented with Jumping Knowledge (JK) connections[79]. This allowed the network to aggregate both local atomic motifs (such as a phosphate group) and larger sub-structures (such as a bulky aromatic ring) simultaneously. As before, the graph layers were followed by standard convolution layers to summarize longer range chemical features about the graph layers, as well as fully connected layers that output the predicted current level. By operating directly on the molecular topology, the GIN was provided with a strong inductive bias regarding the actual physical volume and structure of the peptide inside the nanopore.

5.3.3 Pre-training Strategy

The limited availability of segmented and labeled PASTOR k-mer data presented a significant bottleneck for training data-hungry deep learning models. To overcome this, I implemented a two-stage pre-training and fine-tuning pipeline.

First, I synthesized a large dataset of 100,000 artificial 25-mer sequences. Initially, all samples were generated with randomized amino acid content and sequences. The target current values for these sequences were computationally generated using an updated biophysical squiggler model (see [14]) derived from experimental nanopore data. Unlike earlier models that estimated current blockade levels solely from amino acid volume and charge, this optimized model learned residue-specific signal contributions by fitting model parameters directly to experimentally measured nanopore squiggles using gradient-based optimization. This enabled the synthetic training signals to better capture the complex relationships between sequence composition and observed nanopore current patterns.

The first random synthetic dataset resulted in a normalized current level range of -2.88 to 0.98, while the range of normalized levels in the PASTOR set is -2.32 to 3.55. To ensure the models were exposed to examples with diverse enough charge distributions to capture the effects on the nanopore signal, the pre-training dataset was regenerated with additional aspartic acid and glutamic acid residues added to 5% of the sequences (with a weight of 20% for each residue). This expanded the range of normalized current signals to -5.38 to 3.92, which encompasses the range of the experimental dataset.

Both the CNN and GIN models were pre-trained on this synthetic dataset for 50 epochs with early stopping to prevent overfitting. This step was intended to encourage the networks to learn the underlying biophysical relationships governing nanopore blockades, including how residue composition, charge, and local sequence context influence signal depth, before exposure to noisy experimental measurements. Pre-trained models were then saved and used for downstream fine-tuning.

5.3.4 Fine-tuning and Evaluation Protocol

Following pre-training, the models were fine-tuned using the canonical (unphosphorylated) experimentally measured PASTOR 25-mer dataset. To rigorously evaluate the models' ability to detect post-translational modifications, I introduced a controlled data-splitting regimen for the phosphorylated sequences.

The phosphorylated PASTOR dataset was iteratively split into training and testing fractions, ranging from 0.0 (0% phosphorylated data seen during training) to 0.9 (90% seen during training). For each split size, the training fraction was concatenated with the canonical data to fine-tune the models, which were subsequently evaluated on the unseen test fraction. To ensure statistical robustness, this entire pipeline was repeated for $n=15$ independent runs per split size per model.

5.4 Results: Phosphorylation Detection and Model Generalization

5.4.1 Zero-Shot and Few-Shot Performance

The central question motivating this work was whether structure-informed models could generalize to detect post-translational modifications with limited or no training examples. Figure 5.2 presents the results of the systematic train-test split experiment across both CNN and GIN architectures.

In the zero-shot case (0% of phosphorylated data included during training), the performance difference between architectures was striking. The GIN achieved an average R^2 of 0.62 with RMSE of 0.69, while the CNN managed only R^2 of 0.12 with RMSE of 1.1. This substantial gap demonstrates that the GIN’s explicit encoding of molecular structure enables it to infer the signal contribution of phosphorylated residues based on their chemical properties, even without direct training examples. The phosphate group’s additional atoms, negative charges, and steric bulk are all explicitly represented in the molecular graph, allowing the model to extrapolate from its understanding of how similar chemical features affect nanopore signals.

As the fraction of phosphorylated examples in the training set increased, both models showed improved performance, but the gap between them narrowed considerably. By the time 50% of phosphorylated data was included in training, both architectures achieved R^2 values above 0.80, with the GIN maintaining only a modest advantage. This convergence suggests that with sufficient direct exposure to phosphorylated examples, even the sequence-only CNN can learn to recognize their distinctive signal patterns through memorization of sequence motifs containing phosphorylated residues.

The pattern observed here closely mirrors the results from Chapter 4 for DNA 5-methylcytosine prediction. In both cases, structure-informed models demonstrated substantially superior generalization in zero-shot and few-shot regimes, while sequence-based models caught up given sufficient training data. This

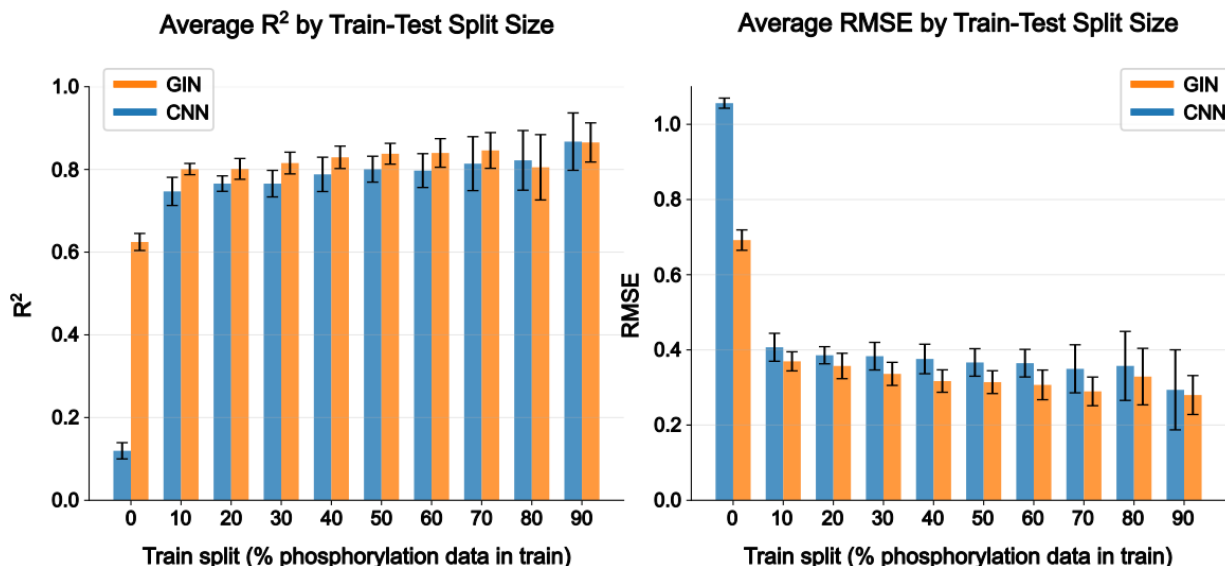


Figure 5.2: Model comparison for phosphorylation detection and generalization. Left: average R^2 values for CNN and GIN models over phosphorylation train-test splits. Right: average RMSE values for CNN and GIN models over phosphorylation train-test splits. In both cases, train split refers to the percentage of the phosphorylated dataset that was added to the canonical training data during model fine-tuning. R^2 and RMSE values correspond to model performance in predicting the held out phosphorylated dataset fraction.

consistency across domains reinforces the conclusion that molecular structure provides a valuable inductive bias for learning nanopore signal relationships, particularly when training data is limited.

5.4.2 Interpreting Model Behavior

The strong zero-shot performance of the GIN merits closer examination. Unlike the CNN, which must learn associations between abstract one-hot encoded symbols and signal values purely from training examples, the GIN operates on a representation that explicitly encodes the physical and chemical properties of the peptide. When encountering a phosphorylated serine for the first time, the GIN "sees" the additional phosphate group as a cluster of phosphorus, oxygen, and hydrogen atoms with specific bonding patterns and charges. The model can leverage its learned understanding of how negatively charged groups and bulky substituents affect ionic current to make reasonable predictions about this novel modification.

This capability is particularly valuable given the experimental realities of protein nanopore sequencing. The k-mer level dataset extracted from the PASTOR dataset contains only 235 phosphorylated 25-mers compared to 1,497 canonical examples. The phosphorylated examples are further subdivided by the number of

phosphorylation sites per k-mer (1, 2, or 3), creating even sparser coverage of the possible phosphorylation patterns. Moreover, the inherent noise in the peptide signal data arising from ClpX motor variability, thermal fluctuations, and the difficulty of cleanly segmenting signals means that each training example provides less information than an idealized noise-free measurement would. Under these constraints, the GIN’s ability to generalize from limited data becomes especially important.

However, the eventual convergence of CNN and GIN performance at higher training fractions raises questions about whether the CNN is truly learning generalizable principles or simply memorizing training examples. The relatively small size of the phosphorylated dataset (235 examples total) means that at 90% training fraction, the model has seen 211 examples, leaving only 24 for testing. With such limited held-out data, distinguishing between genuine generalization and memorization becomes difficult. The CNN may be overfitting to specific sequence contexts where phosphorylation appears in the training set, rather than learning context-independent representations of phosphorylation’s effect on signal.

Testing this hypothesis would require substantially more phosphorylated data spanning diverse sequence contexts. If the CNN’s strong performance at high training fractions reflects memorization rather than true learning, we would expect its accuracy to degrade when tested on phosphorylated sequences in novel contexts not represented in the training set. The GIN, by contrast, should maintain more stable performance across contexts because its predictions are grounded in the chemical structure of the phosphate group rather than memorized sequence patterns.

5.4.3 Proteoform Squiggle Generation and DTW Analysis

While the foundational models described above were trained to predict the nanopore current of discrete 25-mer peptides, the ultimate goal of the system is to process and evaluate full-length proteoforms. To achieve this, I developed a sliding-window inference pipeline capable of generating predicted current-over-time signatures ("squiggles") for arbitrary target sequences of any length.

To predict the signal of a target proteoform, its sequence is first parsed and decomposed into a series of overlapping sub-k-mers. A sliding window of length 25, matching the receptive field of the pre-trained and fine-tuned models, is passed over the target sequence with a step size of 1. For each discrete 25-mer window, the sequence is processed by the respective data-loader (either one-hot encoded for the CNN or converted

to a SMILES-based molecular graph for the GIN).

The saved model checkpoints are then used to predict the expected current level for every resulting sub-sequence. Because nanopore sequencing is an inherently stochastic process affected by noise and motor protein ratcheting variability, Monte Carlo (MC) dropout was enabled during inference[80]. By performing multiple forward passes ($n = 30$) with active dropout for each k-mer, the models generate both a mean predicted current and a standard deviation, providing a measure of model confidence for each region of the sequence.

Once the predictions for all sub-k-mers are generated, they are sequentially stitched back together to form a complete 1D predicted signal representing the expected trace of the full proteoform translocating through the pore. However, because the translocation speed of a peptide through a biological nanopore via the ClpX motor is non-uniform, the empirically measured squiggles contain temporal distortions (e.g., variable dwell times for different residues). To accurately benchmark model predictions against the real-world traces, Dynamic Time Warping (DTW) is utilized. DTW algorithmically aligns the temporal axes of the uniform, model-predicted signal and the non-uniform, empirically measured signal by permitting local stretching and compression along the time dimension. This enables corresponding signal features to be matched even when transient stalls, accelerations, or differences in residue dwell time occur during translocation. As a result, DTW provides a more robust similarity metric for nanopore squiggle comparison by emphasizing agreement in signal shape and structure rather than strict point-by-point temporal correspondence.

To evaluate the practical utility of these models for *de novo* protein sequencing and proteoform identification, I utilized a Top-N ranking approach based on the length-normalized DTW distance. A highly accurate predictive model should generate a theoretical squiggle that yields the lowest possible DTW distance when aligned to its true experimental counterpart, allowing it to be successfully retrieved from a large background of competing experimental traces. By comparing each predicted squiggle against the entire set of experimental squiggles, I computed the Identification Rate (the percentage of traces where the ground-truth corresponding sequence was ranked in the Top-N lowest DTW distances). The GIN architecture consistently demonstrated a superior identification rate compared to the CNN.

Results for the Top-N DTW rankings are shown in Figure 5.3. Performance was evaluated across two datasets: one containing all experimental traces, and a filtered subset designed to limit the over-

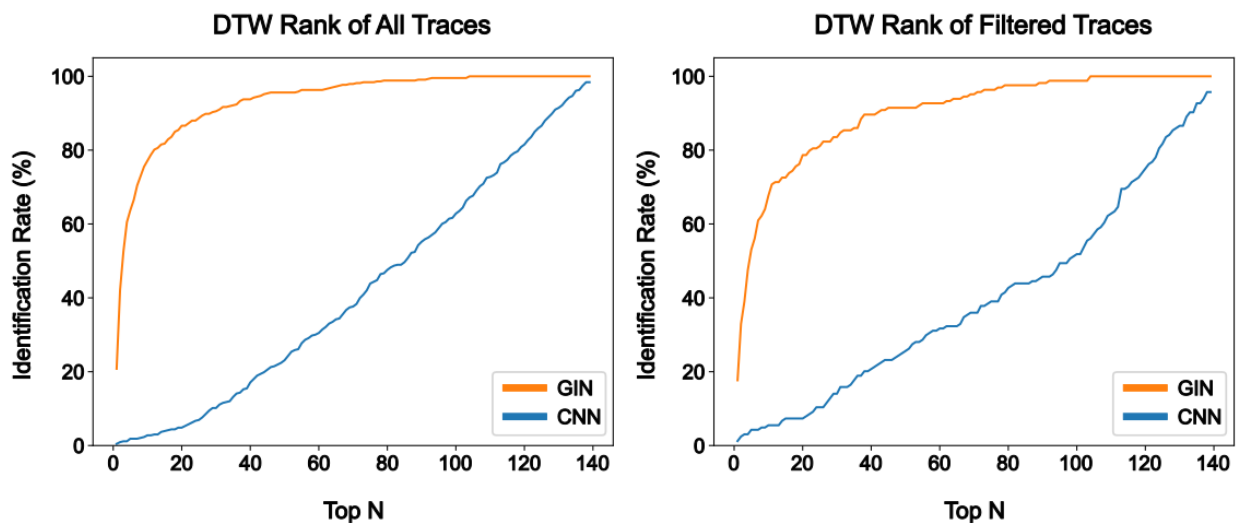


Figure 5.3: Top N DTW Rankings for CNN and GIN proteoform prediction. Left: Top N results for zero-shot CNN and GIN model generated signals compared to all experimental traces. Right: Top N results for zero-shot CNN and GIN model generated signals compared to a subset of all experimental traces filtered to prevent over-representation of canonical signal data over phosphorylation containing signals. The identification rate shows the percentage of cases where the true corresponding experimental signal was successfully aligned to the predicted signal within the top N lowest DTW rankings.

representation of canonical, unphosphorylated reads. For both models, predicted signals were generated from the saved zero-shot trained models, meaning neither model had seen phosphorylated training examples before inference. To calculate the identification rate, each model-generated predicted signal was aligned against every experimental signal in the target set. The traces were then sorted by their length-normalized DTW distance. The identification rate reflects the percentage of cases where the predicted signal and its true corresponding experimental signal were successfully aligned within the Top-N lowest distance rankings.

Quantitative analysis of the alignment results highlights the stark performance gap between the architectures. When evaluated across the full dataset, the GIN achieved a Top-1 identification rate of 20.8%, meaning the true experimental trace was retrieved as the absolute best match in roughly one out of every five cases. In contrast, the CNN baseline achieved a Top-1 rate of only 0.5%. This performance advantage was maintained as the retrieval window expanded. By Top-5, the GIN successfully identified the correct sequence in 63.7% of cases, rapidly approaching a performance asymptote, and reached 77.1% by Top-10. The CNN, however, struggled to exceed a 2.8% identification rate even when considering the Top-10 matches. This trend was consistent in the filtered subset, where over-representation of unphosphorylated,

canonical reads was minimized. In this more challenging targeted dataset, the GIN maintained a strong Top-10 identification rate of 67.7% (compared to the CNN's 5.5%), further validating that the molecular graph representations provide a more robust inductive bias for complex proteoform identification.

5.4.4 Sources of Prediction Error

Several factors contribute to the discrepancies between predicted and measured signals, even for the better-performing GIN model. The training data itself contains substantial noise. The process of extracting clean 25-mer signals from raw PASTOR traces is complicated by the stochastic behavior of the ClpX motor. The motor occasionally pauses, slips backward, or skips forward, creating temporal distortions and misalignments between the sequence and the signal. While the PASTOR design with tyrosine landmarks helps mitigate these issues, the extracted k-mer signals remain noisier than the idealized data the models are designed to predict. This noise in the training labels fundamentally limits model performance.

The twenty-residue sensing window creates a complex many-to-one mapping between sequence and signal. The current level at any moment reflects contributions from approximately twenty amino acids with position-dependent weights. The precise weighting function depends on the detailed geometry of the pore and the conformation of the protein strand within it. While the biophysical squiggler model used to generate pre-training data was optimized on experimental nanopore data and captures many aspects of this relationship, it still represents a simplified approximation. The true weighting function may vary dynamically based on local protein conformation, transient interactions with the pore wall, and other factors not fully captured by the model. The networks must learn these additional complexities implicitly from the training data, and errors in the learned weighting function propagate to all predictions.

Real signals are influenced by factors beyond those explicitly modeled in the pre-training data. These include amino acid-specific interactions with the pore wall, local changes in protein secondary structure as the chain is pulled through the constriction, variations in ionic concentration near charged residues, and the precise geometry of how amino acids pack within the sensing region[78]. While the biophysical model captures the dominant volume-exclusion and charge effects by fitting to experimental data, it necessarily simplifies or omits some of these secondary effects.

Finally, the limited size and diversity of the training data constrains what the models can learn. With

only 1,497 canonical k-mers and 235 phosphorylated k-mers, the models have limited exposure to the full space of possible sequence contexts and modification patterns. Additionally, as shown in Figure 5.4, the PASTOR sequences primarily consist of glycine, serine, aspartic acid, and glutamic acid. Certain amino acid combinations may be underrepresented or absent from the training set, forcing the models to extrapolate in ways that may not be reliable. Expanding the dataset to include more diverse sequences and modification patterns would likely improve model performance substantially.

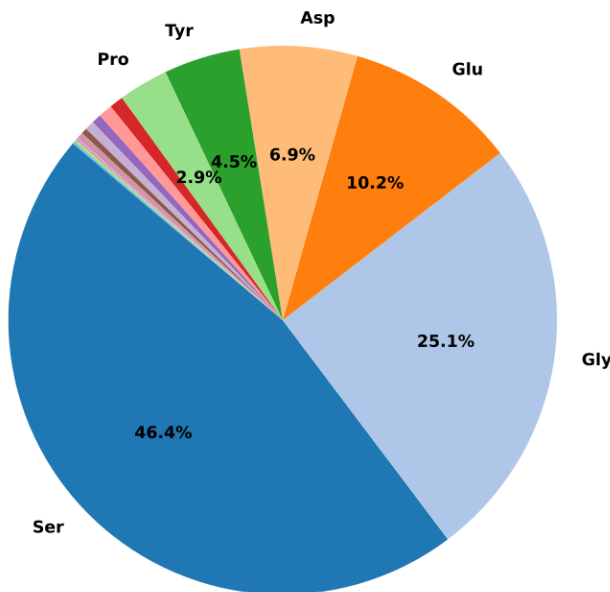


Figure 5.4: Frequencies of amino acids in PASTOR k-mer dataset.

5.5 Discussion

5.5.1 Implications for Protein Nanopore Sequencing

The results presented in this chapter demonstrate that structure-informed models can meaningfully improve prediction of nanopore signals for post-translationally modified proteins, particularly in data-limited regimes. The GIN’s ability to achieve R^2 of 0.62 for zero-shot phosphorylation detection represents a significant advance over purely sequence-based approaches. This capability becomes increasingly important when considering the broader landscape of post-translational modifications.

Phosphorylation represents just one of hundreds of possible PTMs. Proteins can be modified by acety-

lation, methylation, ubiquitination, glycosylation, and numerous other enzymatic and non-enzymatic processes, each potentially occurring at multiple sites and in various combinations[13]. Comprehensively training models to recognize all possible modifications through direct exposure to labeled examples would require an impractically large dataset. Structure-informed models offer a potential path forward by enabling generalization to novel modifications based on their chemical properties rather than requiring exhaustive training data for each modification type.

The Top-N DTW ranking analysis demonstrates the practical utility of this approach for proteoform identification tasks. The GIN’s 20.8% top-1 identification rate in the zero-shot setting, achieved without any exposure to phosphorylated training examples, substantially exceeds the CNN’s 0.5% rate. More importantly, the GIN successfully retrieves the correct sequence within the top 10 matches in 67-77% of cases, transforming an intractable exhaustive search problem into a manageable candidate validation problem. This performance level could enable practical database search workflows where experimental signals are matched against libraries of predicted signals from candidate sequences.

However, current results also highlight important limitations. The gap between top-1 (20.8%) and top-10 (77.1%) performance indicates that models cannot yet reliably identify proteoforms with single-match confidence, suggesting that experimental validation or orthogonal information sources remain necessary for practical applications.

5.5.2 Comparison with XNA Modeling

Comparing the protein modeling results with the XNA work from Chapter 4 reveals a consistent pattern: structure-informed models demonstrate superior generalization in zero-shot and few-shot scenarios across both domains, while sequence-based models catch up given sufficient training data. This consistency suggests that molecular structure provides a generally valuable inductive bias for nanopore signal prediction across different types of biopolymers.

The absolute performance levels differ between domains. The XNA GIN achieved an average R^2 above 0.88 for zero-shot prediction of novel bases, compared to 0.62 for zero-shot phosphorylation detection in proteins; however, direct comparison is complicated by several factors. The protein training data contains more label noise due to variable ClpX motor kinetics and the difficulty of cleanly segmenting signals from

raw traces, whereas the XNA data benefited from more uniform translocation kinetics. Additionally, protein modifications like phosphorylation add bulky chemical groups to amino acids that are already chemically diverse (20 amino acid types), creating more complex and variable signal contributions than the single-position XNA substitutions examined in Chapter 4.

Despite these challenges, achieving R^2 of 0.62 for zero-shot phosphorylation detection and 77.1% top-10 identification rates demonstrates that structure-informed approaches can achieve practical utility even under these more difficult conditions. The consistency of results across both domains strengthens the case that this modeling framework can generalize to other challenging biopolymer sequencing problems.

5.5.3 Limitations and Future Directions

Several limitations of the current work suggest clear directions for future investigation. Dataset size remains a major constraint, as the total number of k-mer examples (1,497 canonical plus 235 phosphorylated) is small by deep learning standards. Expanding the dataset to include more diverse sequences, additional PTMs beyond phosphorylation, and multiple modification sites would enable training more sophisticated models and better characterizing their generalization capabilities.

The current models operate on fixed-length 25-mer windows, matching the approximate size of the nanopore sensing region. This design choice simplifies training but may not optimally capture the complex, position-dependent contributions of amino acids to the signal. Future work could explore attention-based architectures that learn the position-dependent weighting function directly from data, potentially improving prediction accuracy.

Additionally, the models currently predict only the mean current level for each k-mer, while real nanopore signals contain rich temporal dynamics. Extending the models to predict full signal distributions or time-series rather than just mean levels could enable more sophisticated signal analysis and error correction. The current approach also treats each k-mer independently, while protein sequence context beyond the immediate 25-mer window may influence the signal through longer-range effects on protein conformation or motor protein behavior.

Finally, the models operate in a supervised learning framework, requiring labeled training data pairing sequences with measured signals. Semi-supervised or self-supervised approaches that could learn from

unlabeled protein sequences or leverage large protein structure databases might enable better generalization with less experimental data.

5.5.4 Broader Context: Toward Proteoform Sequencing

The ultimate goal of nanopore protein sequencing is to comprehensively characterize unknown proteoforms at single-molecule resolution. Achieving this vision will require advances on multiple fronts including improved experimental methods for controlled protein translocation, larger and more diverse training datasets, more sophisticated computational models, and better integration of complementary information sources.

The structure-informed modeling approach demonstrated in this chapter represents one piece of this larger puzzle. By enabling generalization to novel modifications based on chemical structure, these models could reduce the experimental burden of characterizing new PTMs and accelerate the development of comprehensive proteoform analysis pipelines. Looking forward, as experimental methods mature and datasets grow, the convergence of improving data quality and advancing computational approaches will make comprehensive single-molecule proteoform characterization increasingly feasible. The work presented in this chapter demonstrates that structure-informed machine learning can contribute meaningfully to this effort by enabling models to generalize beyond their direct training experience based on fundamental chemical principles.

Chapter 6

Conclusion

This thesis addressed a central challenge in synthetic biology and molecular engineering: how to make the manipulation and readout of molecular information more scalable. The exponential growth in our ability to design novel molecular systems, from expanded genetic alphabets to engineered proteins, has created a bottleneck at the experimental validation stage. The Design-Build-Test-Learn cycle, while conceptually powerful, remains constrained by the throughput, cost, and labor requirements of traditional laboratory methods. This work explored two complementary approaches to alleviating these constraints: physical automation through digital microfluidics, and computational methods for nanopore sequencing of non-canonical biopolymers.

6.1 Summary of Contributions

6.1.1 Physical Automation: The PurpleDrop Platform

Chapter 2 described the development of PurpleDrop, an open-source digital microfluidic platform designed to make laboratory automation more accessible. The platform's application to DNA data storage workflows and aptamer discovery via SELEX illustrated how digital microfluidics can accelerate iterative molecular engineering processes while reducing reagent consumption and manual labor. However, the critical assessment of PurpleDrop's limitations (particularly regarding reliability, durability, and integration with existing laboratory workflows) highlighted that no single automation technology provides a universal solution at

present. The choice of automation platform depends heavily on the specific application requirements, and digital microfluidics occupies a particular niche in the broader automation landscape.

6.1.2 Computational Methods: Structure-Informed Nanopore Signal Prediction

Chapters 4 and 5 developed a complementary approach to scaling molecular readout by using structure-informed machine learning models to enable nanopore sequencing of molecules for which limited training data exists. The core insight was that explicitly encoding molecular structure, rather than relying solely on sequence information, provides models with inductive biases that enable generalization to novel chemical modifications.

For XNA sequencing (Chapter 4), graph neural networks substantially outperformed sequence-only models in zero-shot prediction of modified nucleobases. This enabled the generation of k-mer tables for expanded genetic alphabets without requiring exhaustive experimental characterization of all possible sequence contexts, and demonstrated practical utility when integrated into basecalling pipelines for DNA containing modifications.

Extension to protein sequencing (Chapter 5) validated the generality of this approach across biopolymer types. Despite the substantially greater complexity of the protein domain (twenty chemically diverse amino acids, variable translocation kinetics, and noisier training data), structure-informed models achieved meaningful performance for zero-shot phosphorylation detection and proteoform identification. These results demonstrated that computational models can reduce the experimental burden of characterizing post-translational modifications, even under challenging data-limited conditions.

Across both domains, a consistent pattern emerged. Structure-informed models excel in zero-shot and few-shot regimes where training data is scarce, while sequence-based models catch up given sufficient examples. This suggests that molecular structure provides the most value precisely where it is needed most, when exploring novel chemical spaces for which comprehensive training data does not yet exist.

6.2 Broader Implications and Future Directions

The convergence of physical automation and computational prediction offers a path toward more scalable molecular engineering. Automation platforms can execute DBTL cycles with greater throughput and con-

sistency, while structure-informed models can extract maximal information from limited experimental data and guide the design of informative experiments. Together, these approaches address different aspects of the scalability challenge. Automation reduces the time and cost per experiment, while computational methods reduce the number of experiments required.

Several directions for future work emerge from this thesis. On the automation front, integration of automated liquid handling with real-time analytical readouts (such as direct coupling of digital microfluidics with nanopore sequencing) could further accelerate DBTL cycles by eliminating manual transfer steps and enabling closed-loop optimization.

For computational methods, expanding structure-informed models to predict full signal time-series rather than just mean k-mer levels could improve accuracy and enable more sophisticated error correction. Semi-supervised and self-supervised learning approaches that leverage unlabeled molecular data or large structure databases could reduce dependence on expensive labeled training data. Transfer learning across related sequencing tasks (for example, using knowledge learned from DNA sequencing to initialize protein sequencing models) might further improve data efficiency.

The structure-informed modeling framework developed here may also extend beyond nanopore sequencing. Many analytical techniques in molecular biology face similar challenges when characterizing novel molecules with limited training data. The principle that explicit encoding of molecular structure enables generalization beyond training distributions could prove valuable across multiple measurement modalities.

6.3 Closing Perspective

The ultimate goal of synthetic biology is not merely to engineer individual molecules but to systematically explore and exploit the vast space of possible molecular functions. Achieving this vision requires moving beyond artisanal, low-throughput approaches toward scalable platforms that can test hypotheses and iterate designs at speeds commensurate with our ability to generate them computationally.

This thesis demonstrated that scalability can be approached from multiple angles. Physical automation makes individual experiments faster and cheaper. Computational prediction makes experiments more informative and reduces the number required. Neither approach alone solves the scalability challenge, but together they create a framework for more systematic molecular engineering.

The specific technologies developed here (PurpleDrop and structure-informed nanopore models) represent steps toward this larger goal rather than complete solutions. Both technologies are most powerful when thoughtfully integrated into broader experimental workflows that leverage their strengths while acknowledging their limitations.

As the field continues to mature, the interplay between automation, computation, and human insight will shape how quickly we can move from molecular designs to validated systems. The work presented in this thesis suggests that by carefully considering where automation adds value, where computation can reduce experimental burden, and how these tools integrate with existing laboratory practice, we can make meaningful progress toward the scalable molecular engineering platforms that the field requires.

Bibliography

- [1] Olivia Gallup, Hia Ming, and Tom Ellis. Ten future challenges for synthetic biology. *Engineering Biology*, 2021.
- [2] Ellis T. Meng F. The second decade of synthetic biology: 2010-2020. *Nature Communications*, 11(1):5174, 2020.
- [3] Pablo Carbonell, Adrian J. Jervis, Christopher J. Robinson, Cunyu Yan, Mark Dunstan, Neil Swainston, Maria Vinaixa, Katherine A. Hollywood, Andrew Currin, Nicholas J. W. Rattray, Sandra Taylor, Reynard Spiess, Rehana Sung, Alan R. Williams, Donal Fellows, Natalie J. Stanford, Paul Mulherin, Rosalind Le Feuvre, Perdita Barran, Royston Goodacre, Nicholas J. Turner, Carole Goble, George Guoqiang Chen, Douglas B. Kell, Jason Micklefield, Rainer Breitling, Eriko Takano, Jean-Loup Faulon, and Nigel S. Scrutton. An automated design-build-test-learn pipeline for enhanced microbial production of fine chemicals. *Communications Biology*, 1(1):66–66, 2018.
- [4] Ashley Stephenson, Lauren Lastra, Bichlien Nguyen, Yuan-Jyue Chen, Jeff Nivala, Luis Ceze, and Karin Strauss. Physical laboratory automation in synthetic biology. *ACS Synthetic Biology*, 12(11), 2023.
- [5] Jameson K. Rogers and George M. Church. Multiplexed engineering in biology. *Trends in Biotechnology*, 34(3):198–206, 2016.
- [6] Ashley Stephenson, Max Willsey, Jeff McBride, Sharon Newman, Bichlien Nguyen, Christopher Takahashi, Karin Strauss, and Luis Ceze. Purpledrops: A digital microfluidics-based platform for hybrid molecular-electronics applications. *IEEE MICRO*, 40(5), 2020.

- [7] Yunhao Wang, Yue Zhao, Audrey Bollas, Yuru Wang, and Kin Fai Au. Nanopore sequencing technology, bioinformatics and applications. *Nature Biotechnology*, 39, 2021.
- [8] Karen Duffy, Sebastian Arangundy-Franklin, and Philipp Holliger. Modified nucleic acids: replication, evolution, and next-generation therapeutics. *BMC Biology*, 18(112), 2020.
- [9] Elisa Biondi and Steven A. Benner. Artificially expanded genetic information systems for new aptamer technologies. *Biomedicines*, 6(2), 2018.
- [10] Hannah Depmeier and Stephanie Kath-Schorr. Expanding the horizon of the xeno nucleic acid space: Threose nucleic acids with increased information storage. *Journal of the American Chemical Society*, 146(11), 2024.
- [11] Shriya Hans, Nilesh Kumar, Nisarg Gohil, Khushal Khambhati, Gargi Bhattacharjee, Shalini S. Deb, Rupesh Maurya, Vinod Kumar, Shamlan M. S. Reshamwala, and Vijai Singh. Rebooting life: engineering non-natural nucleic acids, proteins and metabolites in microorganisms. *Microbial Cell Factories*, 21(100), 2022.
- [12] Michael P. Ledbetter, Rebekah J. Karadeema, and Floyd E. Romesberg. Reprograming the replisome of a semisynthetic organism for the expansion of the genetic alphabet. *Journal of the American Chemical Society*, 140(2), 2018.
- [13] Qian Zhong, Xina Xiao, Yijie Qiu, Zhiqiang Xu, Chunyu Chen, Baochen Chong, Xinjun Zhao, Shan Hai, Shuangqing Li, Zhenmei An, and Lunzhi Dai. Protein posttranslational modifications in health and diseases: Functions, regulatory mechanisms, and therapeutic implications. *MedComm*, 4(3), 2023.
- [14] Keisuke Motone, Daphne Kontogiorgos-Heintz, Wee Jasmine, Kyoko Kurihara, Sangbeom Yang, Gwendolin Roote, Oren E. Fox, Yishu Fang, Melissa Queen, Mattias Tolhurst, Nicolas Cardozo, Miten Jain, and Jeff Nivala. Multi-pass, single-molecule nanopore reading of long protein strands. *Nature*, 633, 2024.
- [15] S. Newman, A. P. Stephenson, M. Willsey, B. H. Nguyen, C. N. Takahashi, K. Strauss, and L. Ceze. High density dna data storage library via dehydration with digital microfluidic retrieval. *Nature Communications*, 10(1), 2019.

- [16] Max Willsey, Ashley P. Stephenson, Chris Takahashi, Pranav Vaid, Bichlien H. Nguyen, Michal Piszczek, Christine Betts, Sharon Newman, Sarang Joshi, Karin Strauss, and Luis Ceze. Puddle: A dynamic, error-correcting, full-stack microfluidics platform. In *Proceedings of ASPLOS '19*, New York, NY, USA, 04 2019. ACM.
- [17] Jörgen Frohm, Veronica Lindström, Mats Winroth, and Johan Stahre. Levels of automation in manufacturing. *Ergonomia*, page 29, 2008.
- [18] P Vinuselvi, S Park, M Kim, JM Park, T Kim, and SK Lee. Microfluidic technologies for synthetic biology. *International Journal of Molecular Sciences*, 12(6):3576–3593, 2011.
- [19] Kihwan Choi, Alphonsus H.C. Ng, Ryan Fobel, and Aaron R. Wheeler. Digital microfluidics. *Annual Review of Analytical Chemistry*, 5(1):413–440, jul 2012.
- [20] Nathan Blow. Microfluidics: the great divide. *Nature Methods*, 6:683–686, 2009.
- [21] Lisa R. Volpatti and Ali K. Yetisen. Commercialization of microfluidic devices. *Trends in Biotechnology*, 32(7):347–350, 2014.
- [22] Keith Robison. Omics! omics! illumina drops neoprep. Accessed: 2023-03-20.
- [23] Michael G. Pollack, Richard B. Fair, and Alexander D. Shenderov. Electrowetting-based actuation of liquid droplets for microfluidic applications. *Applied Physics Letters*, 77(11):1725–1726, 2000.
- [24] Lee Organick, Siena Dumas Ang, Yuan-Jyue Chen, Randolph Lopez, Sergey Yekhanin, Konstantin Makarychev, Miklos Z Racz, Govinda Kamath, Parikshit Gopalan, Bichlien Nguyen, et al. Random access in large-scale dna data storage. *Nature biotechnology*, 36(3):242, 2018.
- [25] Natalia Komarova and Alexander Kuznetsov. Inside the black box: What makes selex better? *Molecules (Basel, Switzerland)*, 24(19):3598, 2019.
- [26] Henning Bär, Remo Hochstrasser, and Bernd Papenfuß. Sila: Basic standards for rapid integration in laboratory automation. *Journal of Laboratory Automation*, 17(2):86–95, 2012.

- [27] Ashley Stephenson, Jayson Sumabat, Hinako Kawabe, Sidharth Lakshmanan, Hyo Joong Kim, Yubing Li, Jorge A Marchand, and Jeff Nivala. Structure-informed models for ionic current prediction in nanopore sequencing of expanded dna alphabets. *Nucleic Acids Research*, 53(22), 2025.
- [28] Hiruna Samarakoon, Yuk Kei Wan, Sri Parameswaran, Jonathan Göke, Hasindu Gamaarachchi, and Ira W. Deveson. Leveraging basecaller’s move table to generate a lightweight k-mer model. *Bioinformatics (Oxford, England)*, 41(4), 2025.
- [29] Ryan R. Wick, Louise M. Judd, and Kathryn E. Holt. Performance of neural network basecalling tools for oxford nanopore sequencing. *Genome Biol*, 20(129), 2019.
- [30] Hinako Kawabe, Christopher A. Thomas, Shuichi Hoshika, Myong-Jung Kim, Myong-Sang Kim, Logan Miessner, Nicholas Kaplan, Jonathan M. Craig, Jens H. Gundlach, Andrew H. Laszlo, Steven A. Benner, and Jorge A. Marchand. Enzymatic synthesis and nanopore sequencing of 12-letter supernumerary dna. *Nature Communications*, 14(6820), 2023.
- [31] Oxford Nanopore Technologies. Dorado. <https://github.com/nanoporetech/dorado>. Accessed: September 2025.
- [32] Oxford Nanopore Technologies. Remora. <https://github.com/nanoporetech/remora>. Accessed: September 2025.
- [33] You Wu, Wenna Shao, Mengxiao Yan, Yuqin Wang, Pengfei Xu, Guoqiang Huang, Xiaofei Li, Brian D. Gregory, Jun Yang, Hongxia Wang, and Xiang Yu. Transfer learning enables identification of multiple types of rna modifications using nanopore direct rna sequencing. *Nature Communications*, 15(4049), 2024.
- [34] Michael C Schatz. Nanopore sequencing meets epigenetics. *Nature Methods*, 14, 2017.
- [35] Bang Wang, Kevin M. Bradley, Myong-Jung Kim, Roberto Laos, Cen Chen, Dietlind L. Gerloff, Luran Manfio, Zunyi Yang, and Steven A. Benner. Enzyme-assisted high throughput sequencing of an expanded genetic alphabet at single base resolution. *Nature Communications*, 15(4057), 2024.

- [36] Qing Dai, Chang Ye, Iryna Irkliyenko, Yiding Wang, Hui-Lung Sun, Yun Gao, Yushuai Liu, Alana Beadell, José Perea, Ajay Goel, and Chuan He. Ultrafast bisulfite sequencing detection of 5-methylcytosine in dna and rna. *Nature Biotechnology*, 42, 2024.
- [37] Jared T Simpson, Rachael E Workman, P C Zuzarte, Matei David, L J Dursi, and Winston Timp. Detecting dna cytosine methylation using nanopore sequencing. *Nature Methods*, 14(4), 2017.
- [38] Meng-Yin Li, Yi-Lun Ying, Jie Yu, Shao-Chuang Liu, Ya-Qian Wang, Shuang Li, and Yi-Tao Long. Revisiting the origin of nanopore current blockage for volume difference sensing at the atomic level. *JACS Au*, 1(7):967–976, 2021.
- [39] Hongxu Ding, Ioannis Anastopoulos, Andrew D. Bailey IV, Joshua Stuart, and Benedict Paten. Towards inferring nanopore sequencing ionic currents from nucleotide chemical structures. *Nature Communications*, 12(6545), 2021.
- [40] Kevin G Devine and Sohan Jheeta. De novo nucleic acids: A review of synthetic alternatives to dna and rna that could act as bio-information storage molecules. *Life*, 10(12), 2020.
- [41] Roy Straver, Carlo Vermeulen, Joe R Verity-Legg, Marc Pagès-Gallego, Dieter G G Stoker, Alexander van Oudenaarden, and Jeroen de Ridder. Requant: improved base modification calling by k-mer value imputation. *Nucleic Acids Research*, 53(9), 2025.
- [42] Z. Wang, Y. Fang, Z. Liu, N. Hao, H. H. Zhang, X. Sun, J. Que, and H. Ding. Adapting nanopore sequencing basecalling models for modification detection via incremental learning and anomaly detection. *Nature Communications*, 15(1), 2024.
- [43] Xiufei Chen, Haiqi Xu, Xiao Shu, and Chun-Xiao Song. Mapping epigenetic modifications by sequencing technologies. *Cell Death Differentiation*, 32, 2025.
- [44] Mauricio Perez, Michiko Kimoto, Priscilla Rajakumar, Chayaporn Suphavitai, Rafael Peres da Silva, Hui Pen Tan, Nicholas Ting Xun Ong, Hannah Nicholas, Ichiro Hirao, Wei Leong Chew, and Niranjan Nagarajan. Direct high-throughput deconvolution of unnatural bases via nanopore sequencing and bootstrapped learning. *Nature Communications*, 16(1), 2025.

- [45] Sam Kovaka, Paul W. Hook, Katharine M. Jenike, Vikram Shivakumar, Luke B. Morina, Roham Razaghi, Winston Timp, and Michael C. Schatz. Uncalled4 improves nanopore dna and rna modification detection via fast and accurate signal alignment. *Nature Methods*, 22, 2025.
- [46] Laurianne David, Amol Thakkar, Rocío Mercado, and Ola Engkvist. Molecular representations in ai-driven drug discovery: a review and practical guide. *Journal of Cheminformatics*, 12(56), 2020.
- [47] Medard Edmund Mswahili and Young-Seob Jeong. Transformer-based models for chemical smiles representation: A comprehensive literature review. *Heliyon*, 10(20), 2024.
- [48] Jianyuan Deng, Zhibo Yang, Hehe Wang, Iwao Ojima, Dimitris Samaras, and Fusheng Wang. A systematic study of key elements underlying molecular property prediction. *Nature Communications*, 14(6395), 2023.
- [49] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.
- [50] Josep Arús-Pous, Simon Viet Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Randomized smiles strings improve the quality of molecular generative models. *Journal of Cheminformatics*, 11(71), 2019.
- [51] Yimeng Kong, Edward A. Mead, and Gang Fang. Navigating the pitfalls of mapping dna and rna modifications. *Nature Reviews Genetics*, 24(6):363–381, 2023.
- [52] Ziyuan Wang, Ziyang Liu, Yinshan Fang, Hao Helen Zhang, Xiaoxiao Sun, Ning Hao, Jianwen Que, and Hongxu Ding. Training data diversity enhances the basecalling of novel rna modification-induced nanopore sequencing readouts. *Nature Communications*, 16:679, 2025.
- [53] Wang Liu-Wei, Wiep van der Toorn, Patrick Bohn, Martin Holzer, Redmond P. Smyth, and Max von Kleist. Sequencing accuracy and systematic errors of nanopore direct rna sequencing. *BMC Genomics*, 25:528, 2024.
- [54] Michiko Kimoto, Rie Yamashige, Ken-ichiro Matsunaga, Shigeyuki Yokoyama, and Ichiro Hirao.

- Generation of high-affinity dna aptamers using an expanded genetic alphabet. *Nature Biotechnology*, 31(5), 2013.
- [55] Christopher A. Thomas, Henry Brinkerhoff, Jonathan M. Craig, Shuichi Hoshika, Desislava Mihaylova, Akira M. Pfeffer, Michaela C. Franzi, Sarah J. Abell, Jessica D. Carrasco, Jens H. Gundlach, Steven A. Benner, and Andrew H. Laszlo. Sequencing a dna analog composed of artificial bases. *Nature Communications*, 16(7240), 2025.
- [56] Tsutomu Suzuki. The expanding world of trna modifications and their disease relevance. *Nature Reviews Molecular Cell Biology*, 22, 2021.
- [57] Cora Palanca-Ballester, Aitor Rodriguez-Casanova, Susana Torres, Silvia Calabuig-Fariñas, Francisco Exposito, Diego Serrano, Esther Redin, Karmele Valencia, Eloisa Jantus-Lewintre, Angel Diaz-Lagares, Luis Montuenga, Juan Sandoval, and Alfonso Calvo. Cancer epigenetic biomarkers in liquid biopsy for high incidence malignancies. *Cancers*, 13(12), 2021.
- [58] Yorke Zhang, Brian M. Lamb, Aaron W. Feldman, Anne Xiaozhou Zhou, Thomas Lavergne, Lingjun Li, and Floyd E. Romesberg. A semisynthetic organism engineered for the stable expansion of the genetic alphabet. *Proceedings of the National Academy of Sciences*, 114(6), 2017.
- [59] Yorke Zhang, Jerod L. Ptacin, Emil C. Fischer, Hans R. Aerni, Carolina E. Caffaro, Kristine San Jose, Aaron W. Feldman, Court R. Turner, and Floyd E. Romesberg. A semi-synthetic organism that stores and retrieves increased genetic information. *Nature*, 551(7682), 2017.
- [60] Rie Yamashige, Michiko Kimoto, Yusuke Takezawa, Akira Sato, Tsuneo Mitsui, Shigeyuki Yokoyama, and Ichiro Hirao. Highly specific unnatural base pair systems as a third base pair for pcr amplification. *Nucleic Acids Research*, 40(6), 2012.
- [61] Itaru Okamoto, Yuya Miyatake, Michiko Kimoto, and Ichiro Hirao. High fidelity, efficiency and functionalization of ds-px unnatural base pairs in pcr amplification for a genetic alphabet expansion system. *ACS Synthetic Biology*, 5(11), 2016.
- [62] Denis A Malyshev, Young Jun Seo, Phillip Ordoukhanian, and Floyd E Romesberg. Pcr with an expanded genetic alphabet. *Journal of the American Chemical Society*, 131(41), 2009.

- [63] Lingjun Li, Melissa Degardin, Thomas Lavergne, Denis A Malyshev, Kirandeep Dhani, Phillip Ordoukhanian, and Floyd E Romesberg. Natural-like replication of an unnatural base pair for the expansion of the genetic alphabet and biotechnology applications. *Journal of the American Chemical Society*, 136(3), 2014.
- [64] Zunyi Yang, Fei Chen, J. Brian Alvarado, and Steven A Benner. Amplification, mutation, and sequencing of a six-letter synthetic genetic system. *Journal of the American Chemical Society*, 133(38), 2011.
- [65] Cheng Zhang, Ranfeng Wu, Fajia Sun, Yisheng Lin, Yuan Liang, Jiongjiong Teng, Na Liu, Qi Ouyang, Long Qian, and Hao Yan. Parallel molecular data storage by printing epigenetic bits on dna. *Nature*, 634(8035), 2024.
- [66] S. Kasra Tabatabaei, Bach Pham, Chao Pan, Jingqian Liu, Shubham Chandak, Spencer A Shorkey, Alvaro G Hernandez, Aleksei Aksimentiev, Min Chen, Charles M Schroeder, and Olgica Milenkovic. Expanding the molecular alphabet of dna-based data storage systems with neural network nanopore readout processing. *Nano letters*, 22(5), 2022.
- [67] Alexandre Goyon and Kelly Zhang. Characterization of antisense oligonucleotide impurities by ion-pairing reversed-phase and anion exchange chromatography coupled to hydrophilic interaction liquid chromatography/mass spectrometry using a versatile two-dimensional liquid chromatography setup. *Analytical Chemistry*, 92(8), 2020.
- [68] Mark Matzas, Peer F Stahler, Nathalie Kefer, Nicole Siebelt, Valesca Boisguerin, Jack T Leonard, Andreas Keller, Cord F Stahler, Pamela Haberle, Baback Gharizadeh, Farbod Babrzadeh, and George M Church. High-fidelity gene synthesis by retrieval of sequence-verified dna identified using high-throughput pyrosequencing. *Nature Biotechnology*, 28(12), 2010.
- [69] Zachary A. Rollins, Alan C. Cheng, and Essam Metwally. Molprop: Molecular property prediction with multimodal language and graph fusion. *Journal of Cheminformatics*, 16(1), 2024.
- [70] Mayk Caldas Ramos, Christopher J Collison, and Andrew D White. A review of large language models and autonomous agents in chemistry. *Chemical science (Cambridge)*, 16(6), 2025.

- [71] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020.
- [72] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, pages 1877–1901, 2020.
- [73] Shaghayegh Sadeghi, Alan Bui, Ali Forooghi, Jianguo Lu, and Alioune Ngom. Can large language models understand molecules? *BMC Bioinformatics*, 25(1), 2024.
- [74] Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Paye Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12), 2022.
- [75] Kevin Maik Jablonka, Philippe Schwaller, Andres Ortega-Guerrero, and Berend Smit. Leveraging large language models for predictive chemistry. *Nature Machine Intelligence*, 6(2), 2024.
- [76] Justas Ritmejeris, Xiuqi Chen, and Cees Dekker. Single-molecule protein sequencing with nanopores. *Nature Reviews Bioengineering*, 3, 2025.
- [77] Kingsley L.-J. Wong, Mattias Tolhurst, Oren A. Fox, Safwan Diwan, Nicholas Bogard, and Jeff Ni-vala. The next generation of protein sequencing and analysis methods. *Annual review of analytical chemistry*, 19(1), 2025.
- [78] Jingqian Liu and Aleksei Aksimentiev. Molecular determinants of current blockade produced by peptide transport through a nanopore. *ACS Nanoscience Au*, 4(1), 2024.
- [79] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks, 2018.

- [80] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2016.

Chapter A

Appendix One

A.1 Appendix section 1

% Data in Train	Run 1		Run 2	
	RMSE	Pearson Correlation	RMSE	Pearson Correlation
5%	7.073411	0.850875	7.273616	0.840934
10%	2.892713	0.974804	3.029744	0.972298
15%	2.406627	0.982497	2.385879	0.98266
20%	1.93742	0.98861	1.962553	0.988284
25%	1.761766	0.99056	1.789292	0.990283
30%	1.548831	0.992754	1.51182	0.993128
35%	1.388982	0.994245	1.403452	0.994087
40%	1.262943	0.995228	1.26697	0.995198
45%	1.158834	0.995994	1.132259	0.996149
50%	1.026323	0.996902	1.057582	0.996654
55%	0.971362	0.997233	0.995794	0.997051
60%	0.914756	0.99757	0.92492	0.997505
65%	0.863655	0.997816	0.880145	0.997692
70%	0.844554	0.997878	0.8422	0.997936
75%	0.810389	0.998078	0.804096	0.998092
80%	0.772177	0.998258	0.77765	0.998245
85%	0.760691	0.998356	0.76042	0.998336
90%	0.722573	0.998495	0.728488	0.998459
95%	0.711158	0.998508	0.729184	0.998475

Table A.1: Average RMSE and Pearson Correlation values for CNN model 50-fold cross validation on canonical ONT 6mer. Run 1 and 2 are independent replicates with 50 repeats per train-test split size.

% 5mC in Train	Run 1		Run 2	
	RMSE	Pearson's Correlation	RMSE	Pearson's Correlation
1%	1.526148	0.990959	1.483316	0.991408
5%	1.074622	0.995559	1.078923	0.995524
10%	1.109373	0.995852	1.085289	0.995914
30%	0.93123	0.996947	0.951015	0.996927
50%	0.910663	0.99719	0.883279	0.997215
70%	0.863836	0.997366	0.865005	0.997398
90%	0.838019	0.997437	0.846091	0.997473

Table A.2: Average RMSE and Pearson Correlation values for CNN model 50-fold cross validation on 5mC-containing 6mer dataset. Run 1 and 2 are independent replicates with 50 repeats. Models were trained using all canonical 6mer data plus varying percentages of the 5mC-containing dataset, then evaluated on the remaining portion of the 5mC set.

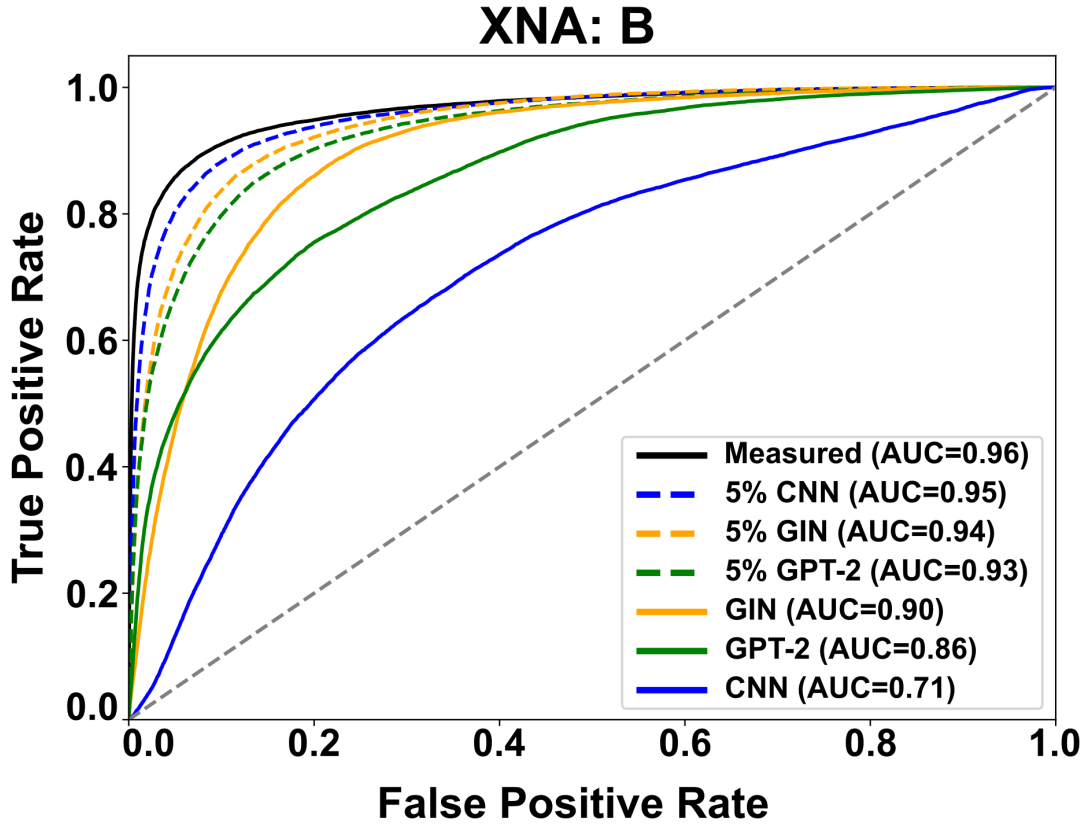


Figure A.1: Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (B) dataset was added to the model training data. Solid lines represent *de novo* prediction cases where no examples of the held out XNA base were included in the training data.

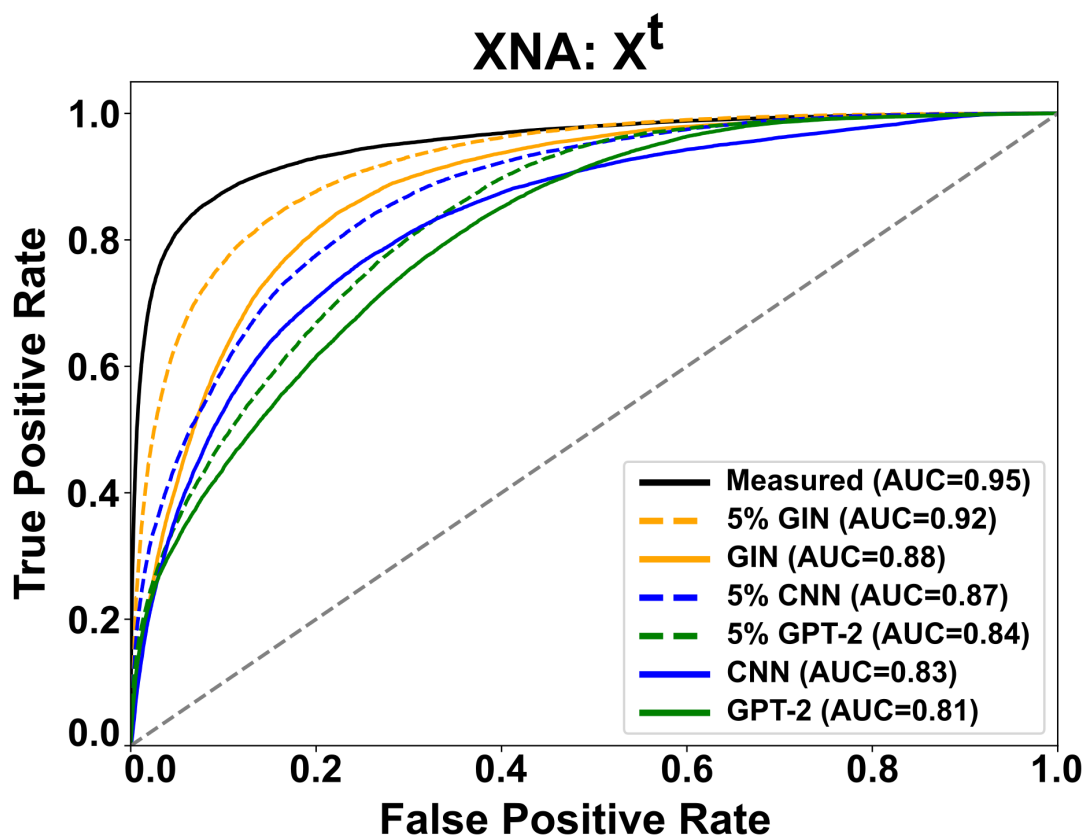


Figure A.2: Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (X^t) dataset was added to the model training data. Solid lines represent *de novo* prediction cases where no examples of the held out XNA base were included in the training data.

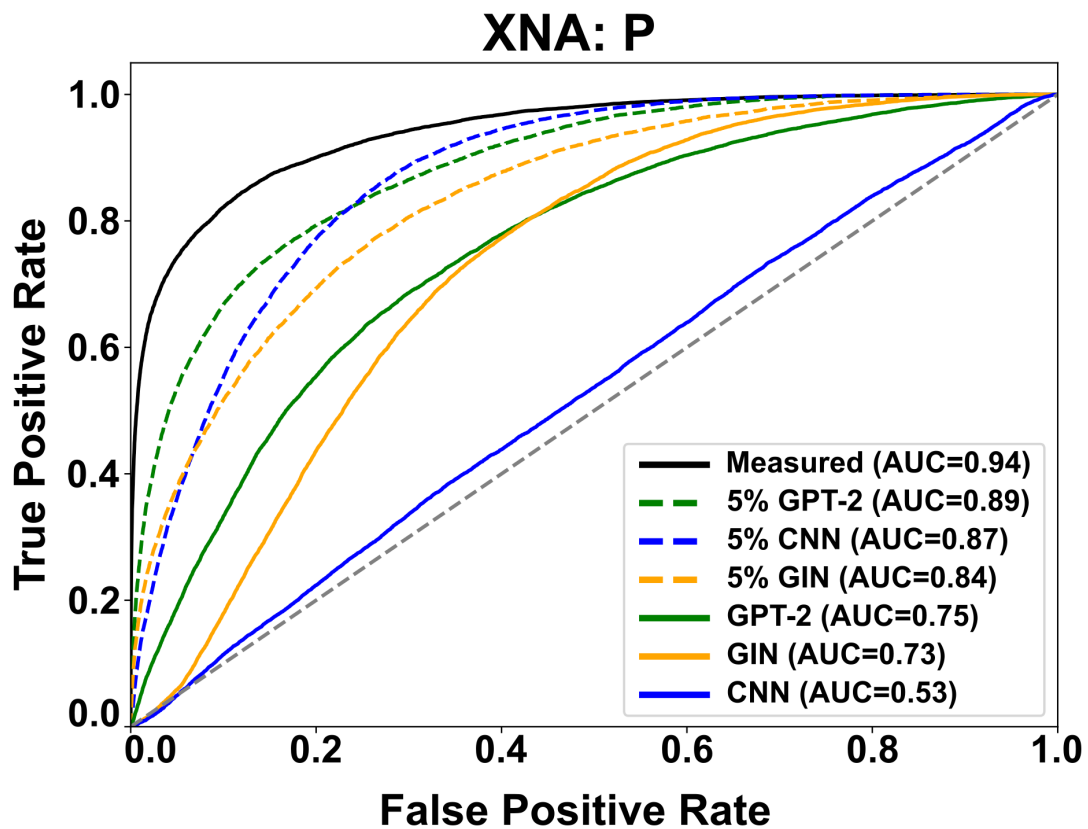


Figure A.3: Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (P) dataset was added to the model training data. Solid lines represent *de novo* prediction cases where no examples of the held out XNA base were included in the training data.

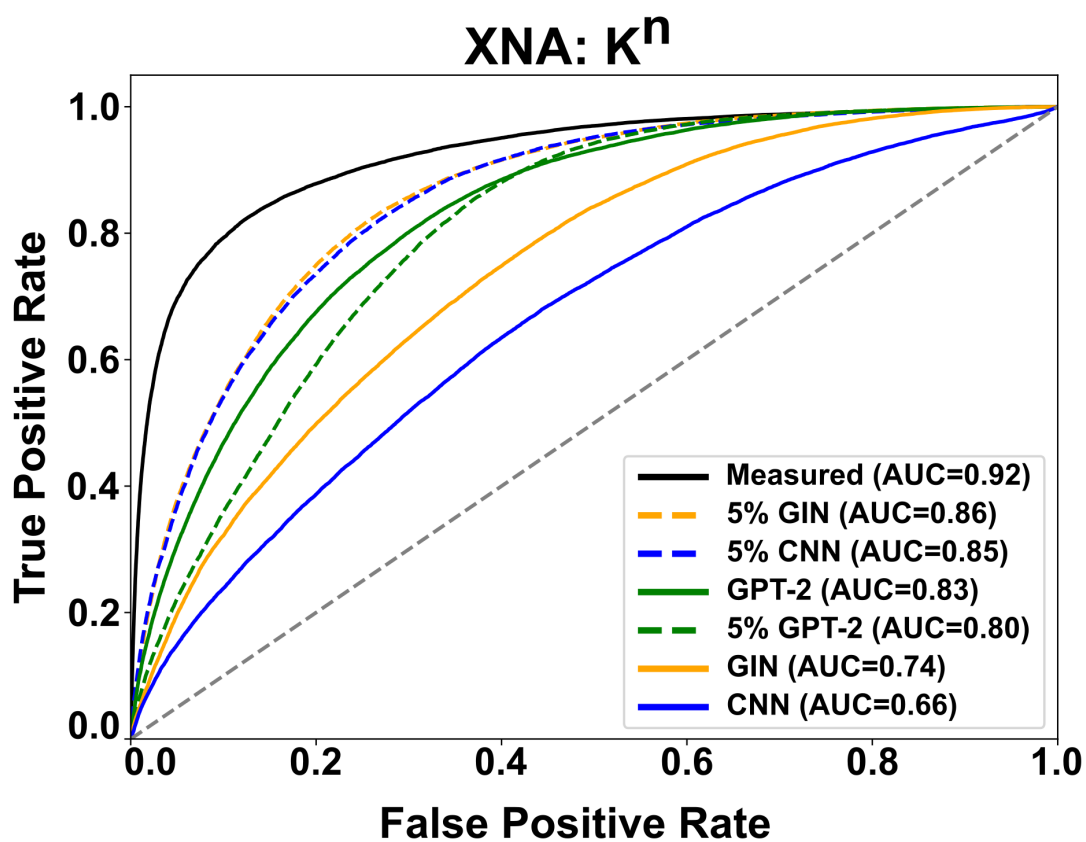


Figure A.4: Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (K^n) dataset was added to the model training data. Solid lines represent *de novo* prediction cases where no examples of the held out XNA base were included in the training data.

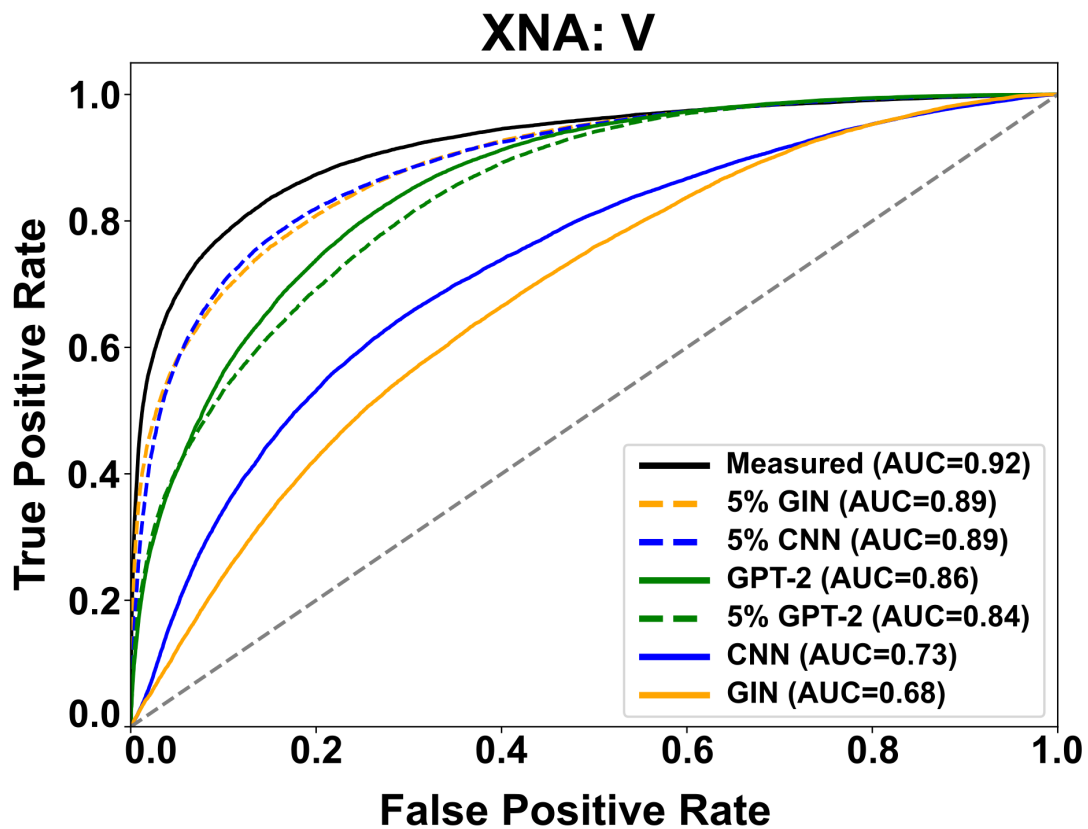


Figure A.5: Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (V) dataset was added to the model training data. Solid lines represent *de novo* prediction cases where no examples of the held out XNA base were included in the training data.

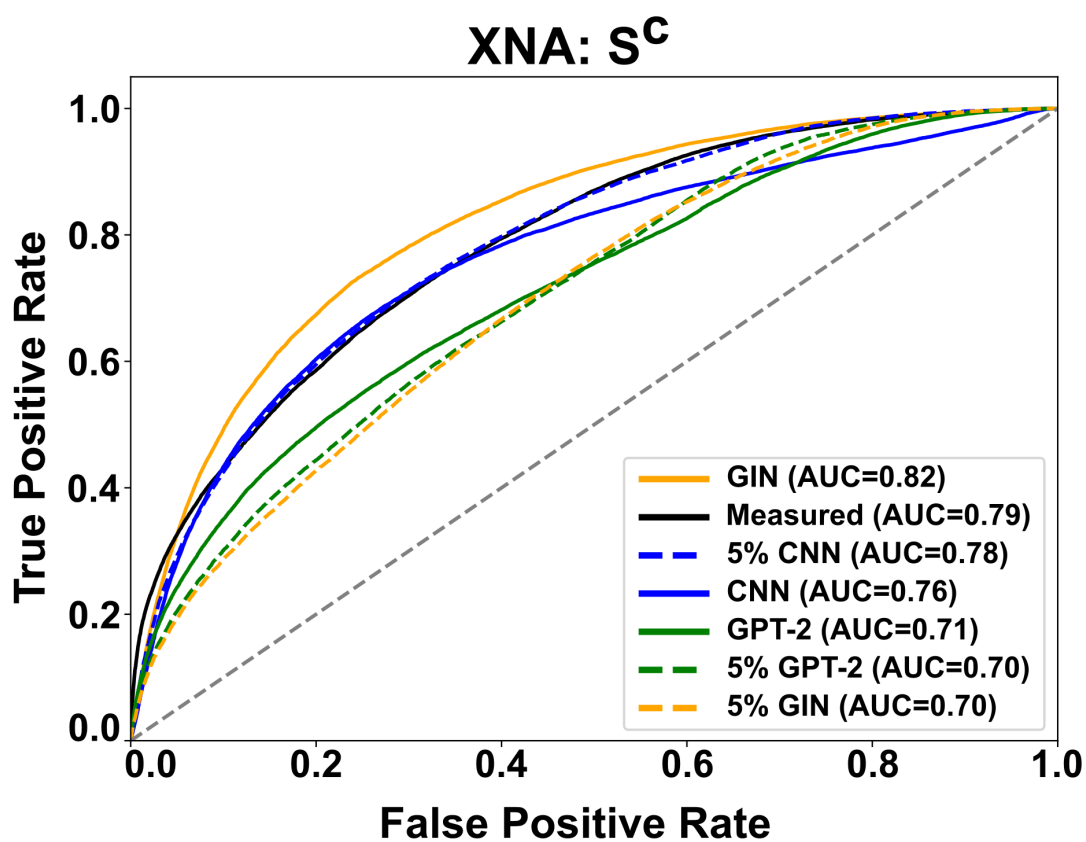


Figure A.6: Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (S^c) dataset was added to the model training data. Solid lines represent *de novo* prediction cases where no examples of the held out XNA base were included in the training data.

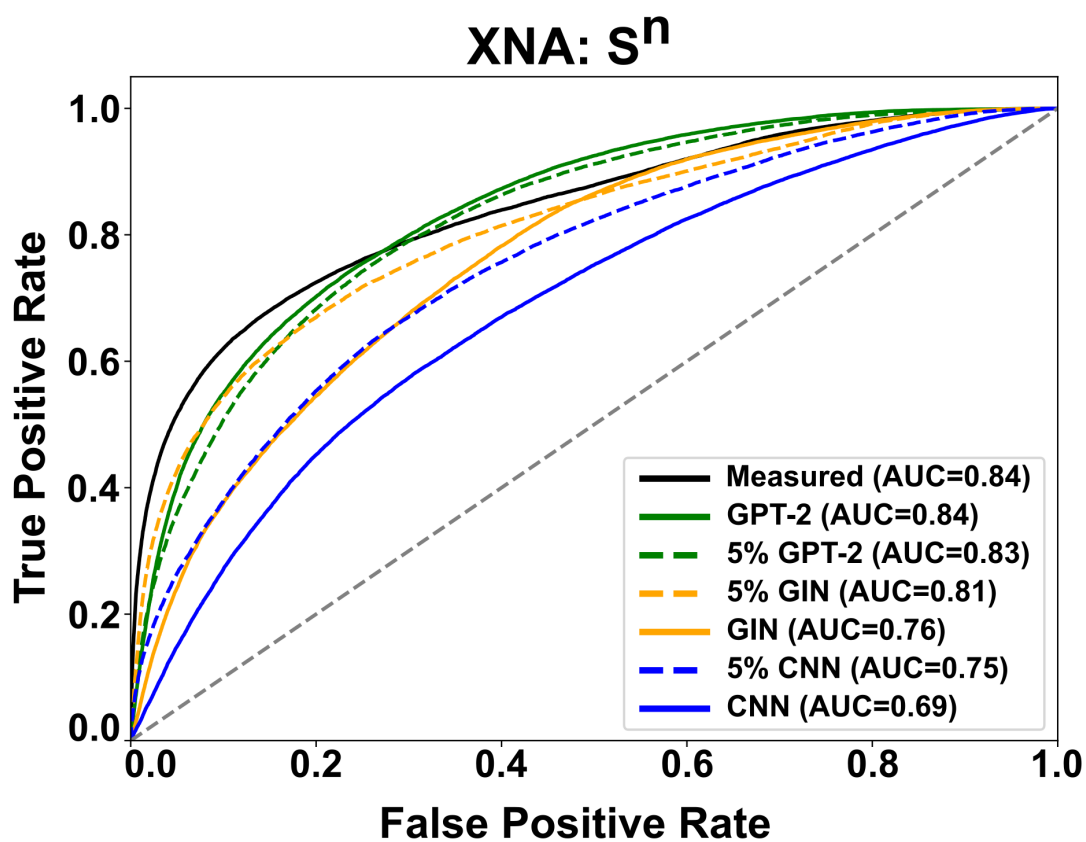


Figure A.7: Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (S^n) dataset was added to the model training data. Solid lines represent *de novo* prediction cases where no examples of the held out XNA base were included in the training data.

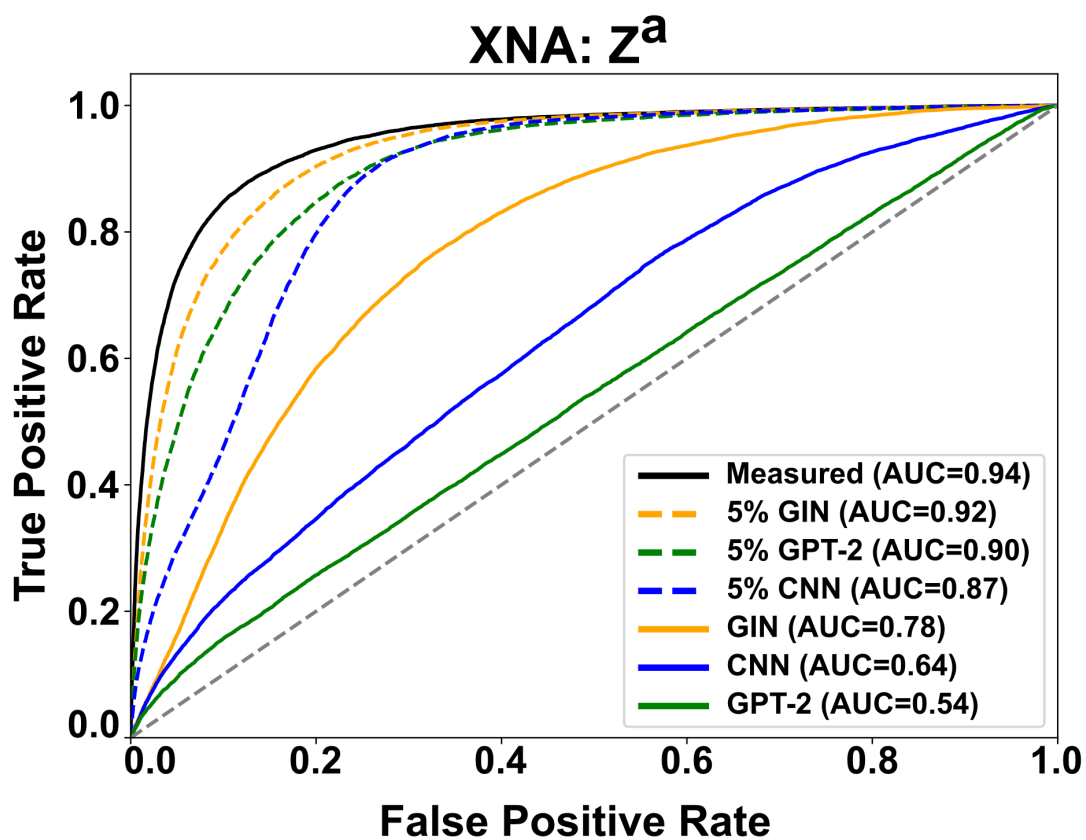


Figure A.8: Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (Z^a) dataset was added to the model training data. Solid lines represent *de novo* prediction cases where no examples of the held out XNA base were included in the training data.

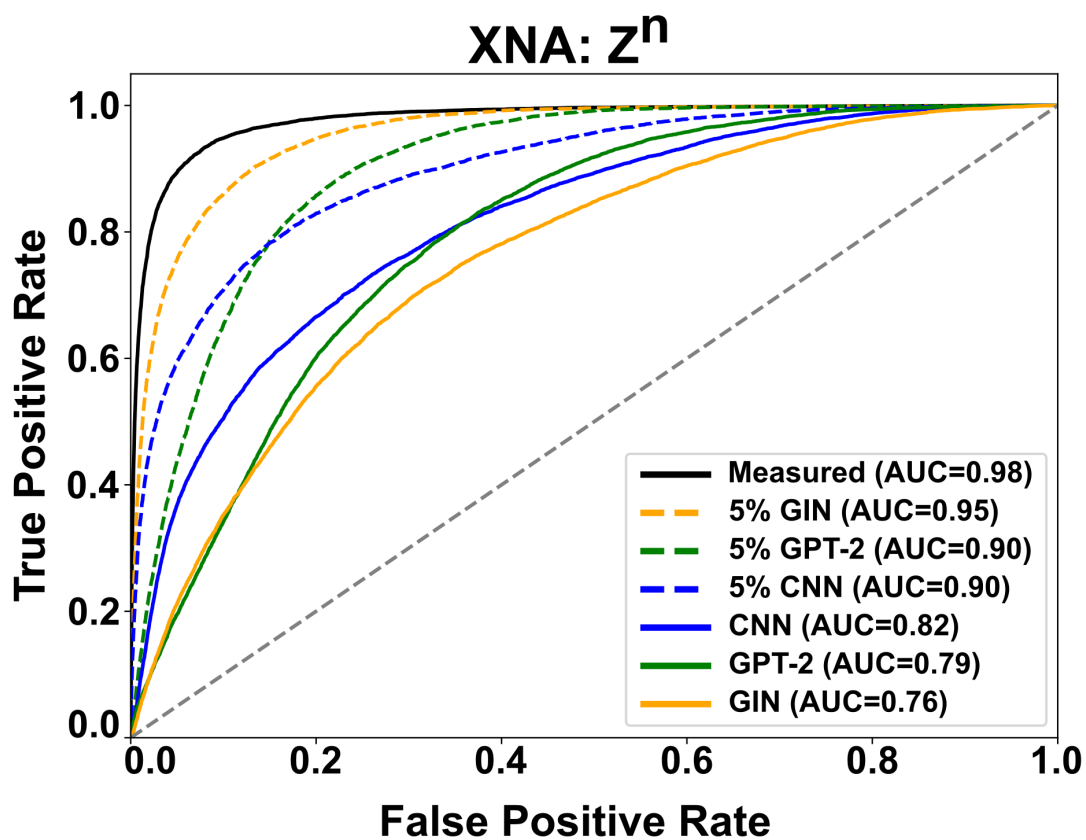


Figure A.9: Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (Z^n) dataset was added to the model training data. Solid lines represent *de novo* prediction cases where no examples of the held out XNA base were included in the training data.

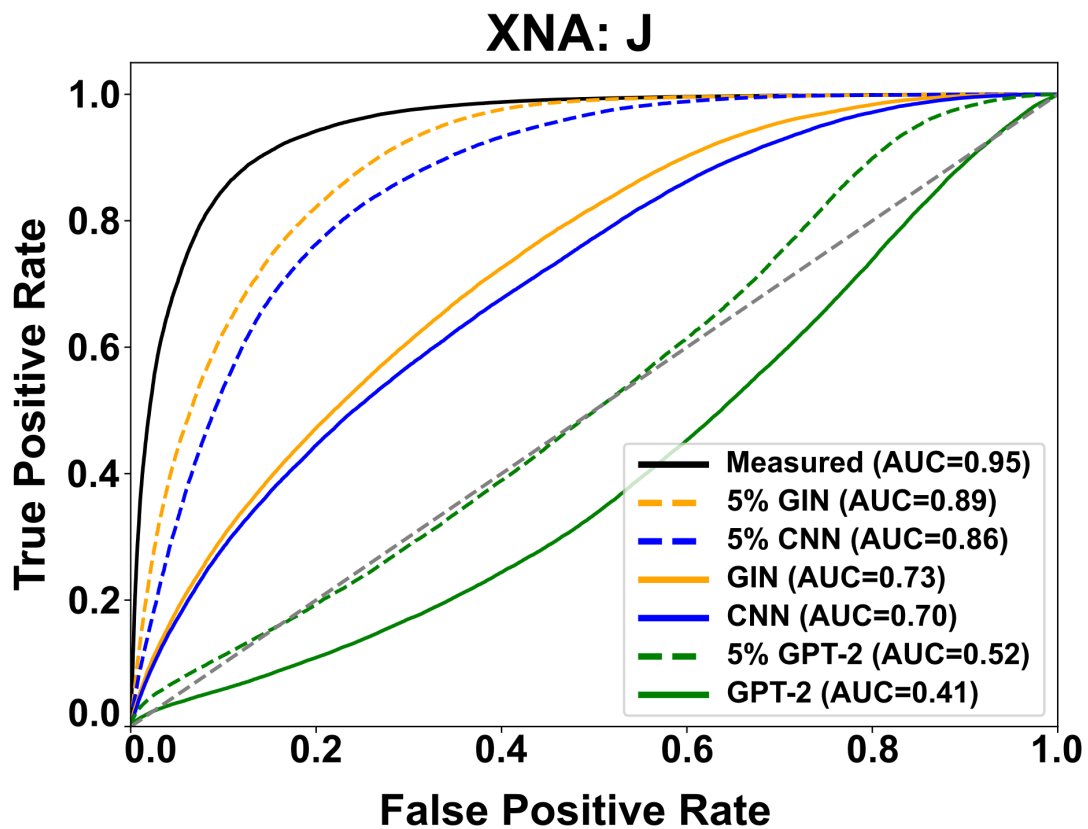


Figure A.10: Receiver operating characteristic (ROC) curves with area under the curve (AUC) values for different model-generated k-mer tables used in XNA basecalling. Models with 5% prefix (dashed lines) denote cases where 5% of the held out XNA base (J) dataset was added to the model training data. Solid lines represent *de novo* prediction cases where no examples of the held out XNA base were included in the training data.

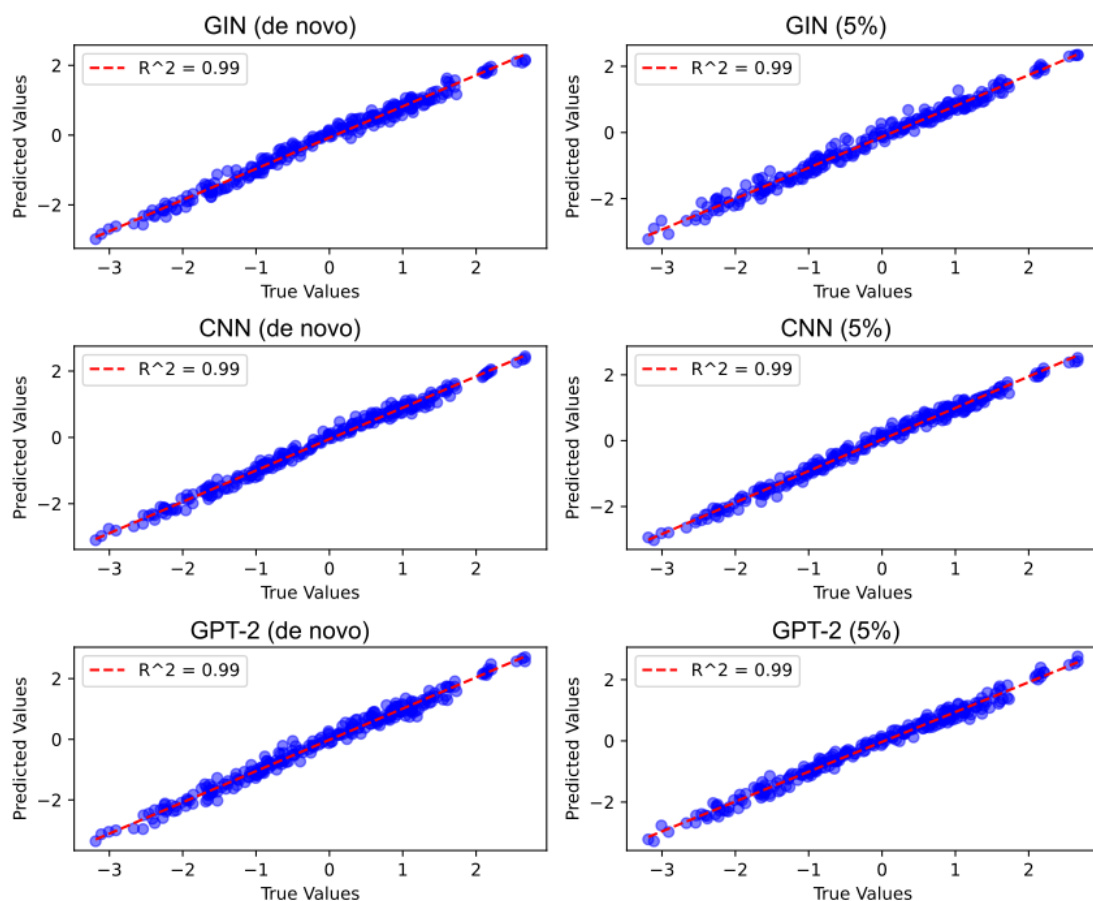


Figure A.11: Example scatter plots for canonical (ATGC) dropout experiments. Models were trained with all XNA datasets and tested on the canonical 4-mer set. For 5% cases, 5% of the held out canonical set was added to the training set.

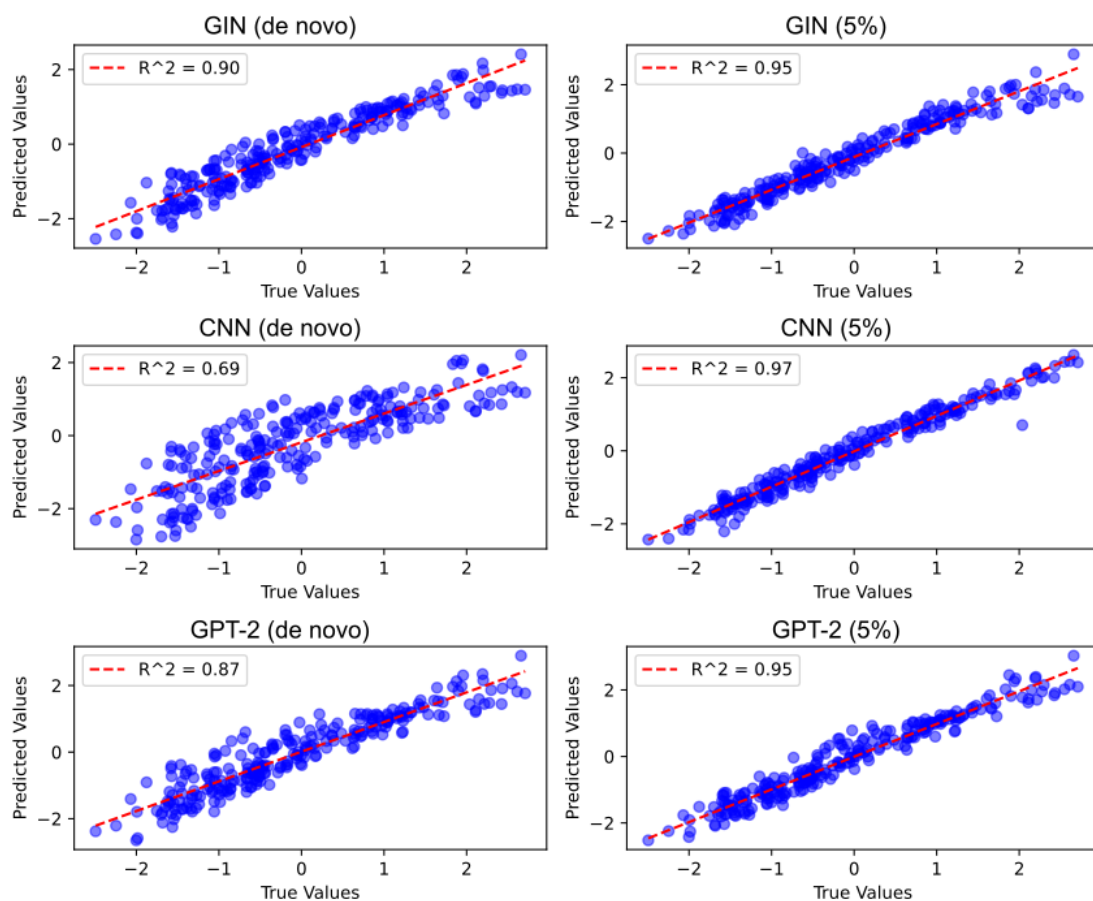


Figure A.12: Example scatter plots for XNA base B dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.

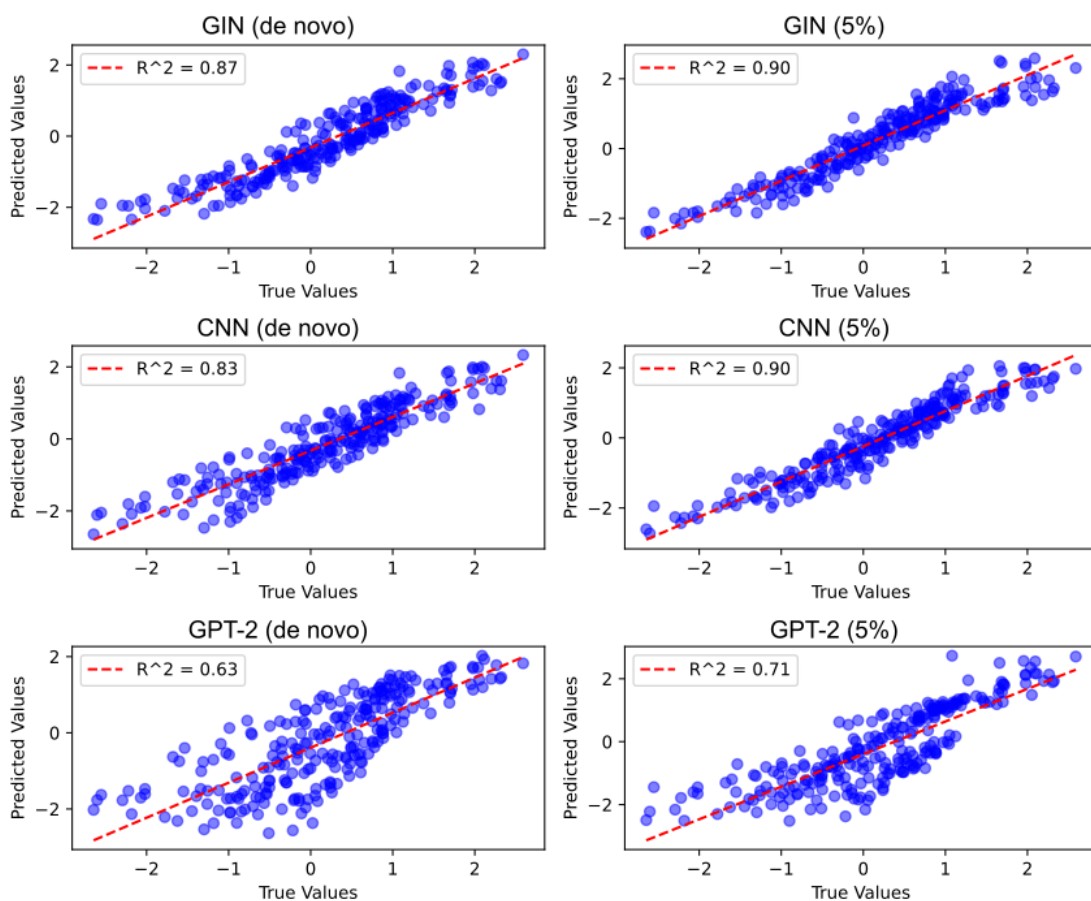


Figure A.13: Example scatter plots for XNA base J dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.

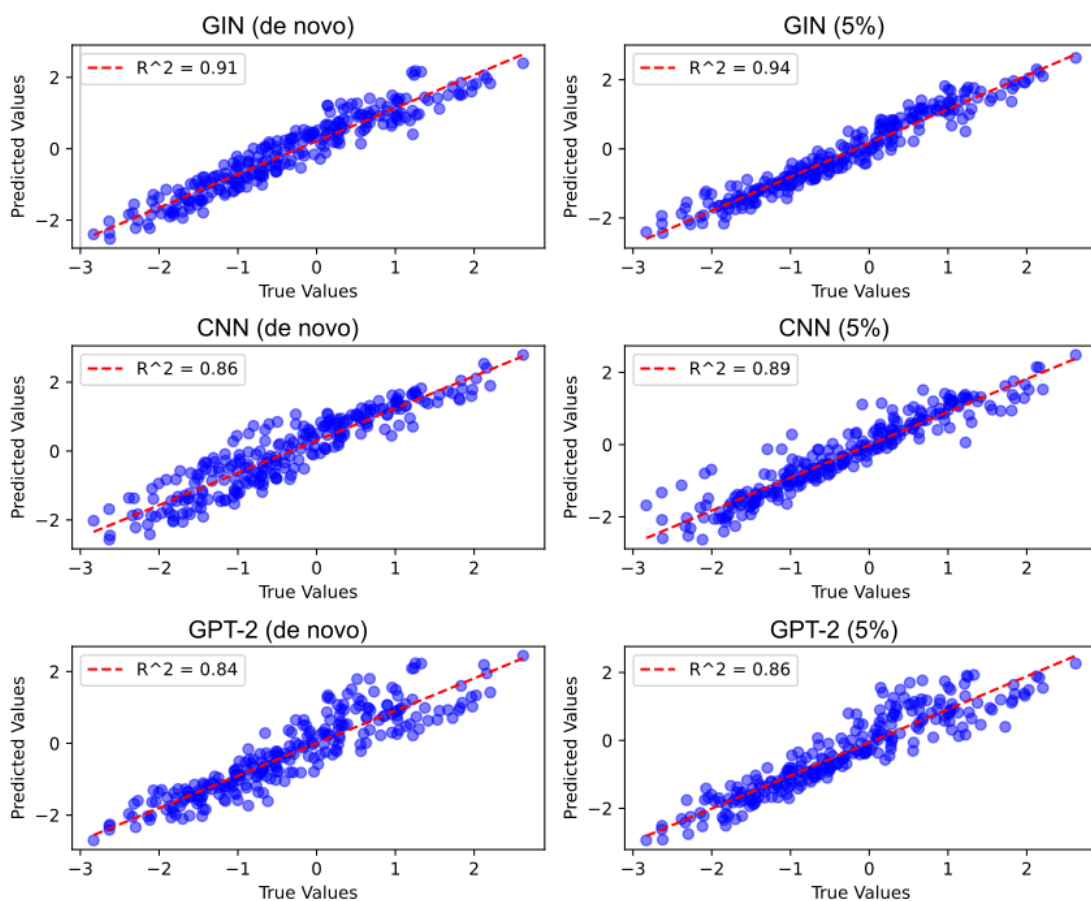


Figure A.14: Example scatter plots for XNA base X^t dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.

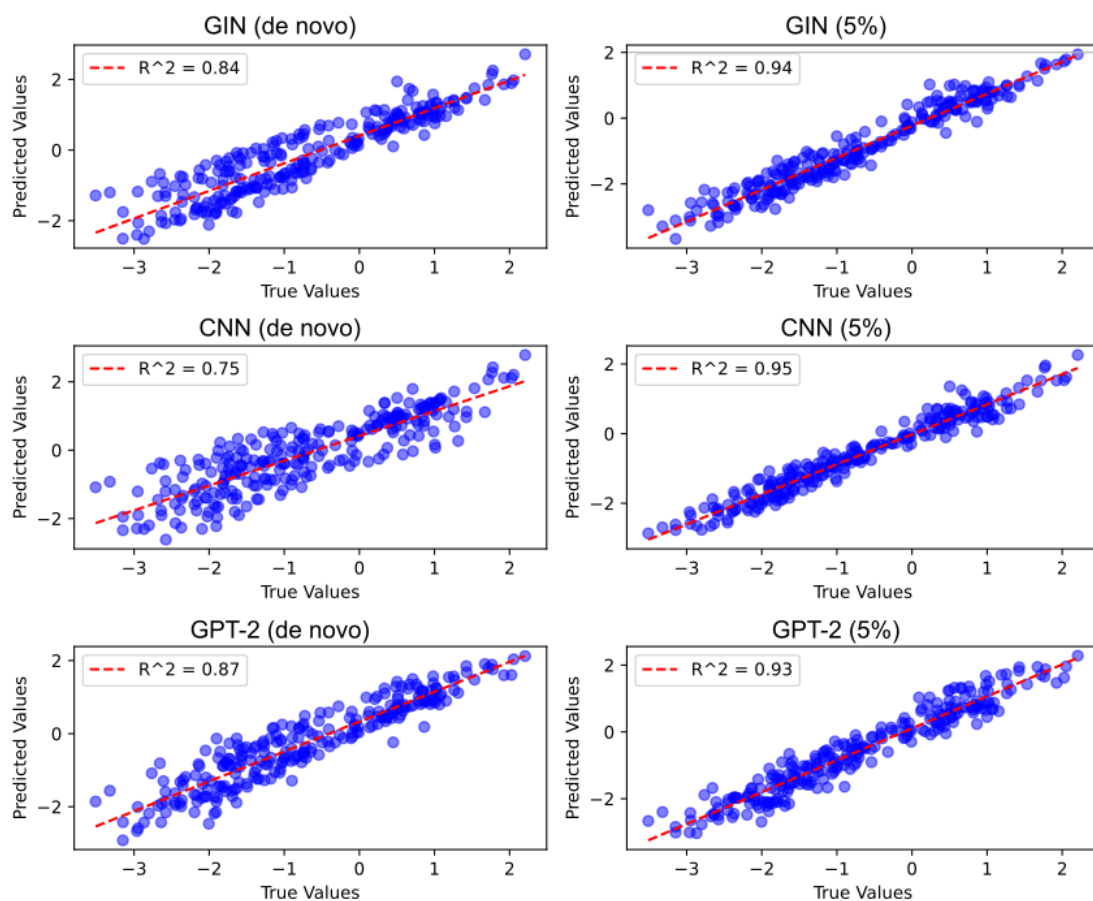


Figure A.15: Example scatter plots for XNA base P dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.

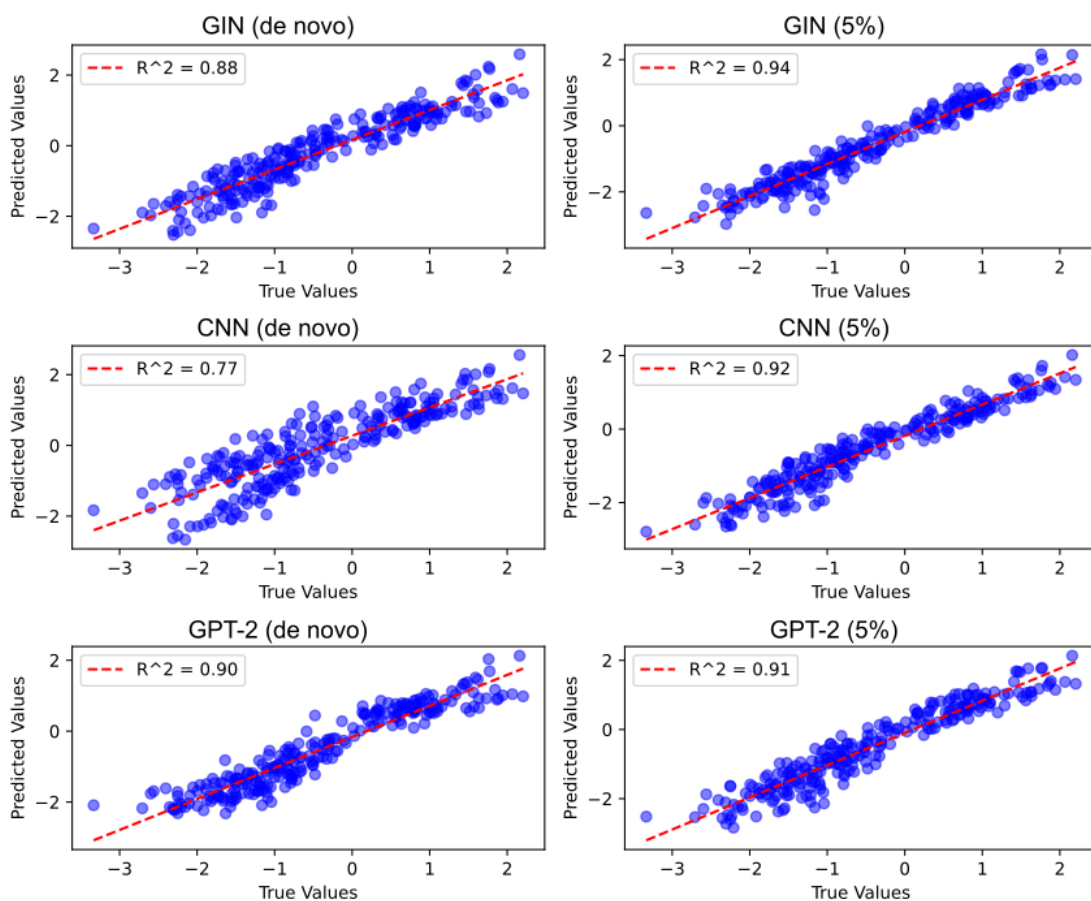


Figure A.16: Example scatter plots for XNA base K^n dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.

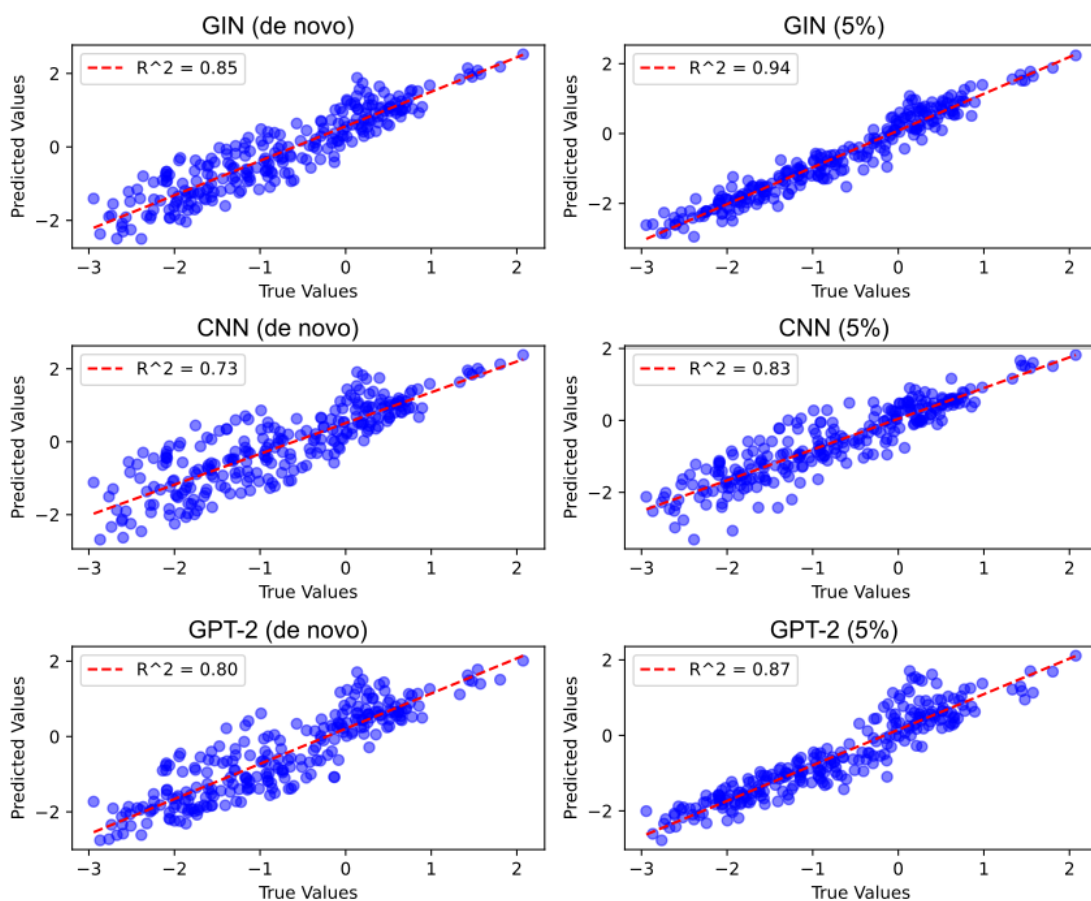


Figure A.17: Example scatter plots for XNA base V dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.

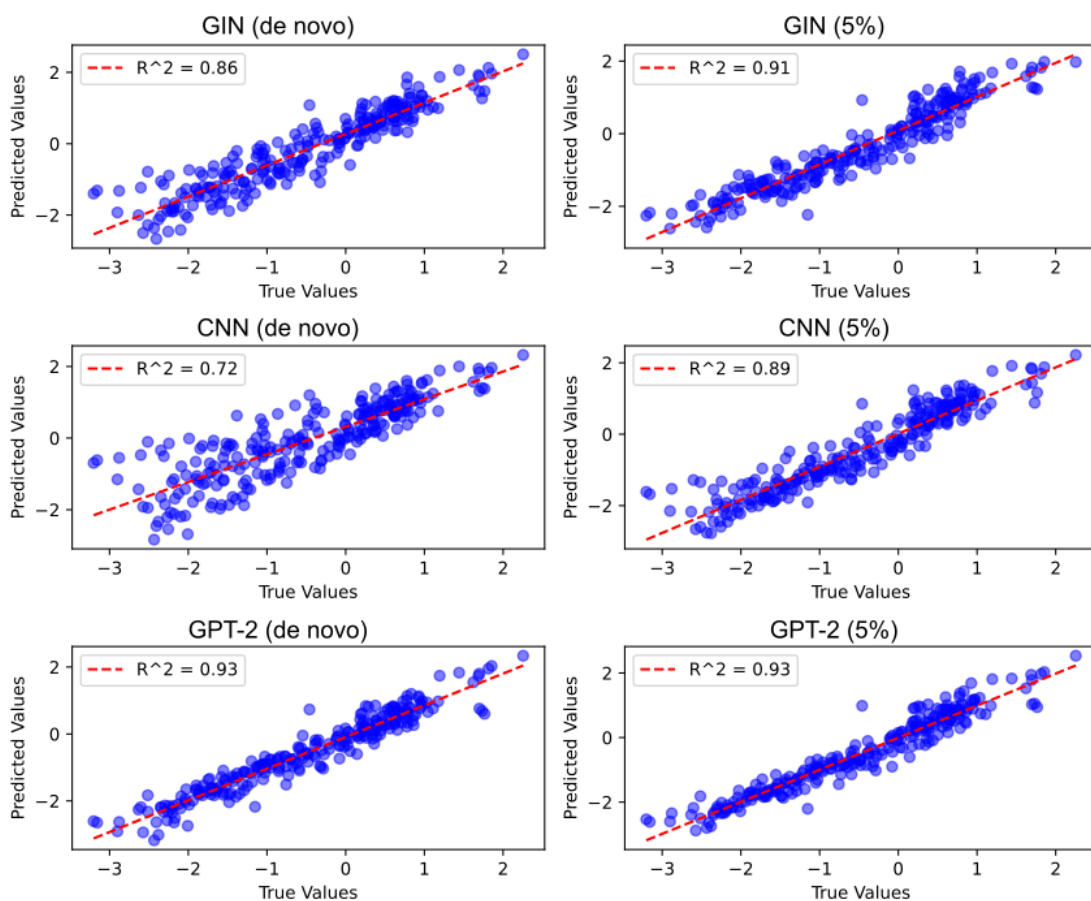


Figure A.18: Example scatter plots for XNA base S^c dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.

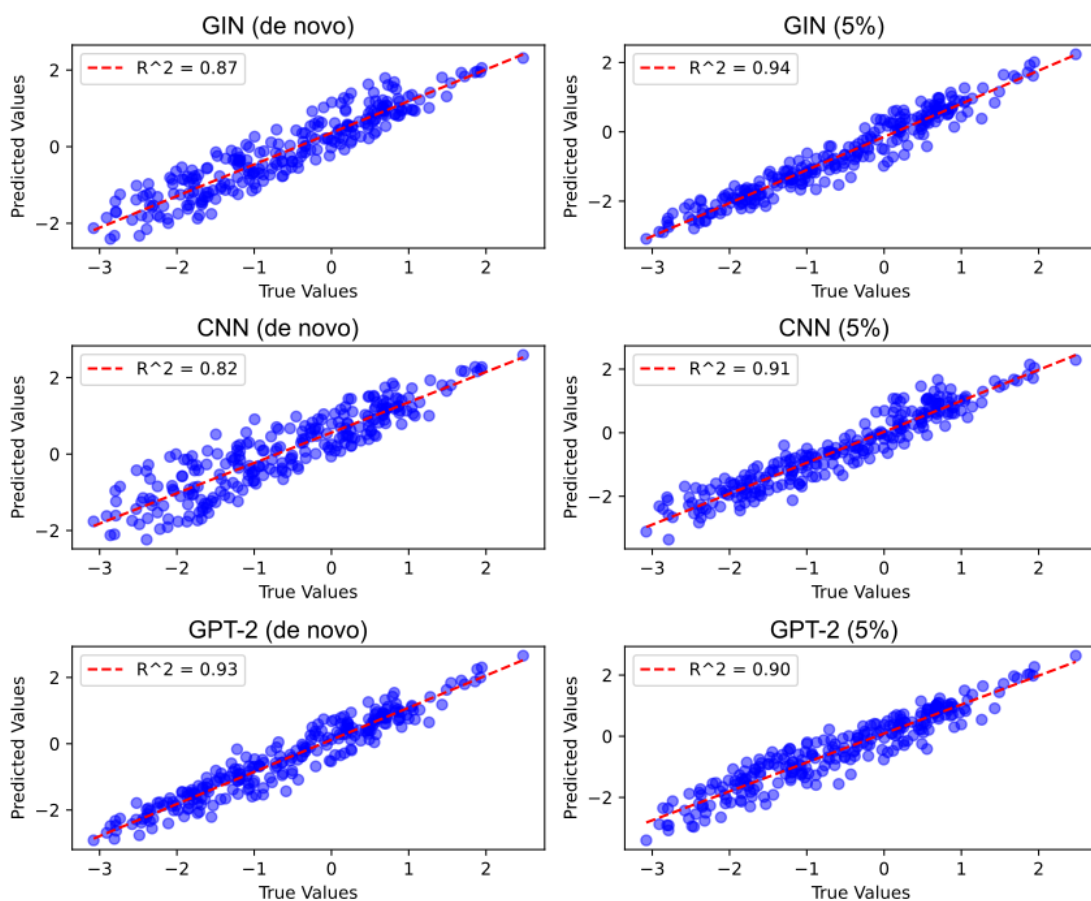


Figure A.19: Example scatter plots for XNA base S^n dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.

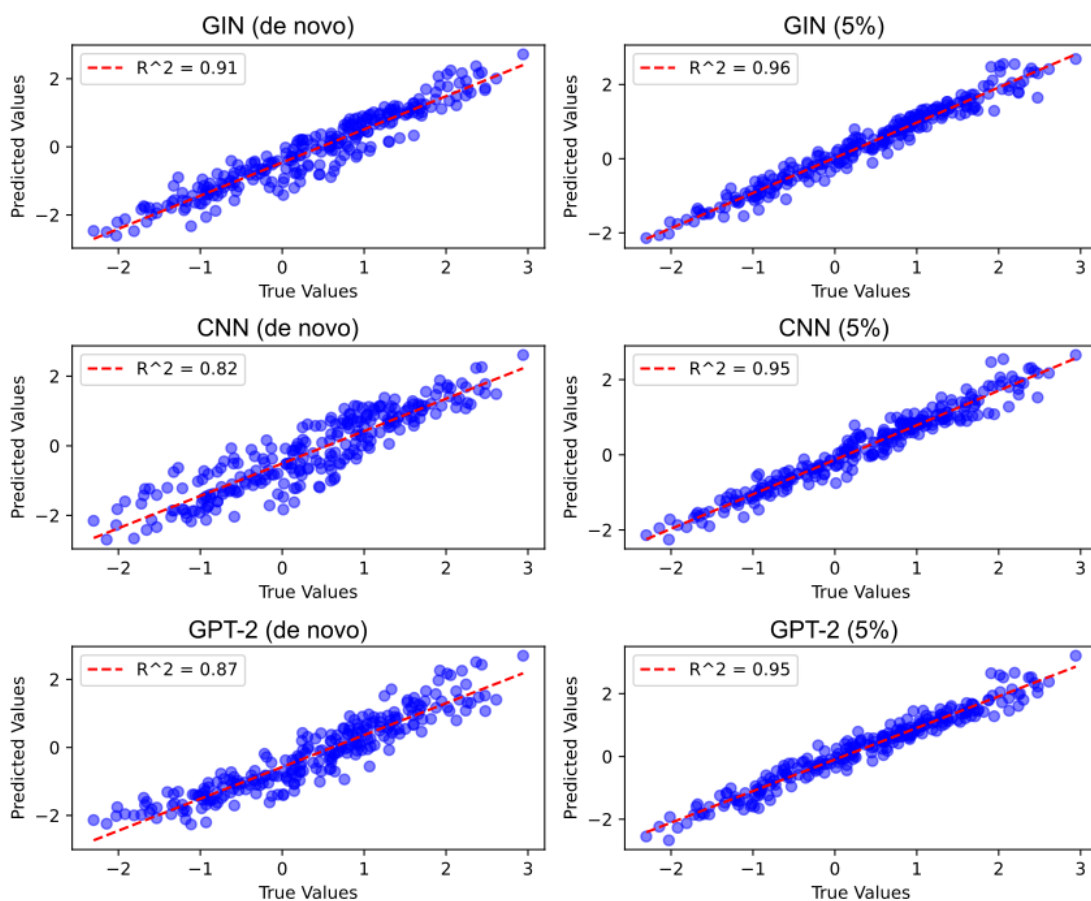


Figure A.20: Example scatter plots for XNA base Z^a dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.

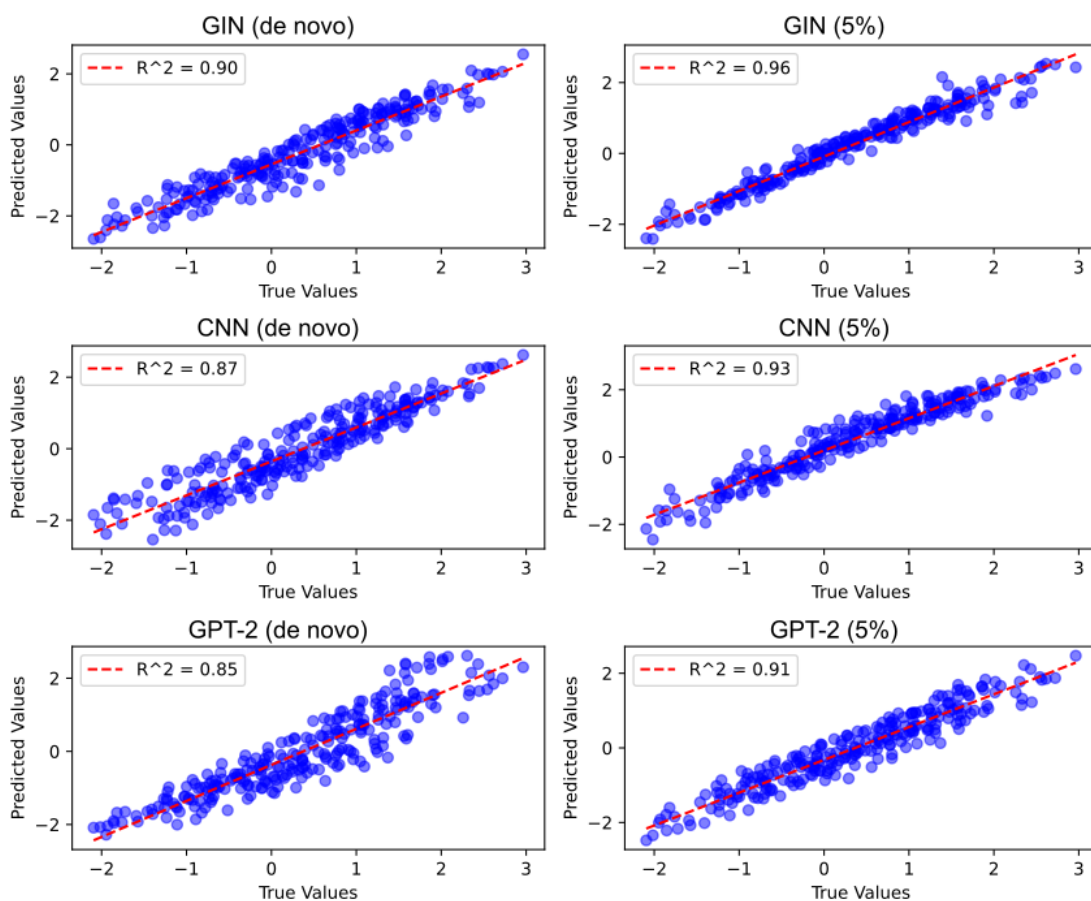


Figure A.21: Example scatter plots for XNA base Z^n dropout experiments. Models were trained with the canonical 4-mer dataset and all other XNA datasets. For 5% cases, 5% of the held out base was added to the training set.

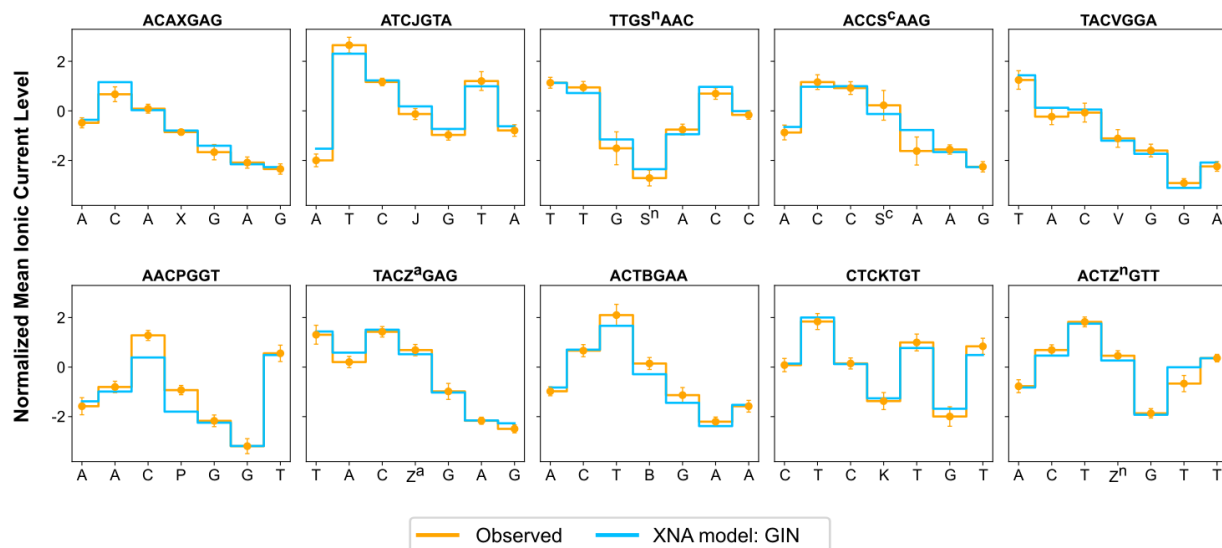


Figure A.22: Example plots overlaying model-predicted k-mer mean current levels (blue) on measured current levels (orange) for representative heptamers/bases. Error bars indicate the standard deviation of the measured normalized signal level. Model-predicted levels shown here were generated by the GIN architecture trained under few-shot conditions (i.e. trained with 5% of each held-out XNA dataset added to the training data.)

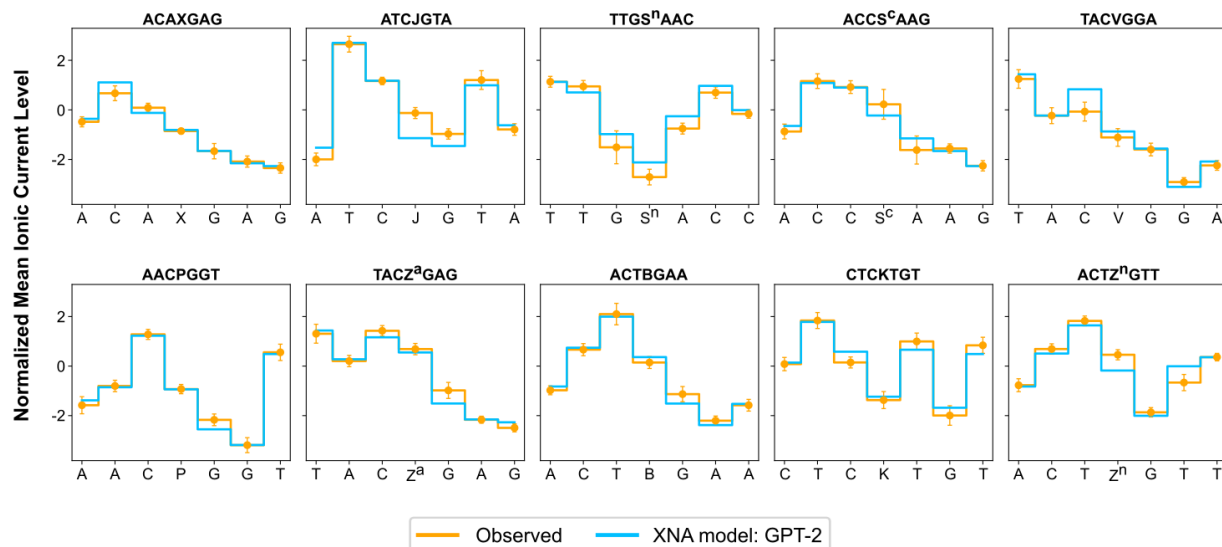


Figure A.23: Example plots overlaying model-predicted k-mer mean current levels (blue) on measured current levels (orange) for representative heptamers/bases. Error bars indicate the standard deviation of the measured normalized signal level. Model-predicted levels shown here were generated by the GPT-2 architecture trained under few-shot conditions (i.e. trained with 5% of each held-out XNA dataset added to the training data.)

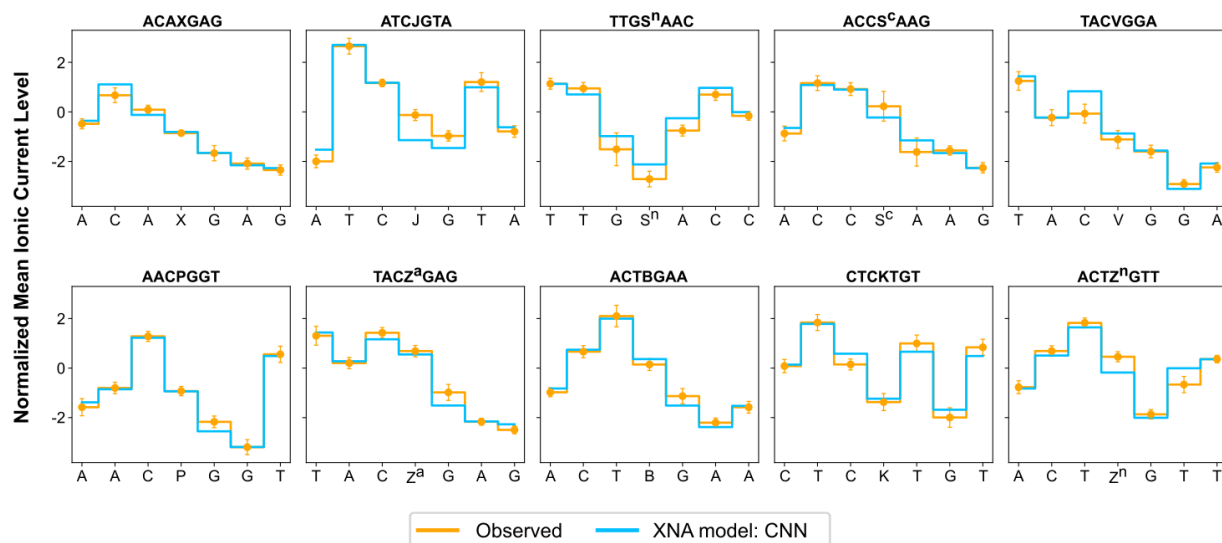


Figure A.24: Example plots overlaying model-predicted k-mer mean current levels (blue) on measured current levels (orange) for representative heptamers/bases. Error bars indicate the standard deviation of the measured normalized signal level. Model-predicted levels shown here were generated by the CNN architecture trained under few-shot conditions (i.e. trained with 5% of each held-out XNA dataset added to the training data.)