BWAF-Net: Enhanced Human Promoter Identification via Biologically Weighted Attention Fusion of Transformer and Graph Attention Networks

By

Zemedkun Abebe Debela

Submitted to the School of Information Technology and Engineering

In partial fulfillment of the requirements for the degree of

Master of Science in Artificial Intelligence

Supervised by: Dr. Adane Letta Mamuye

College of Technology and Built Environment
Addis Ababa University

Addis Ababa, Ethiopia

October 10, 2025

APPROVAL PAGE

This is to certify that the thesis titled **"BWAF-Net: Enhanced Human Promoter Identification via Biologically Weighted Attention Fusion of Transformer and Graph Attention Networks"**, prepared by **Zemedkun Abebe Debela**, has been submitted in partial fulfillment of the requirements for the degree of Master of Science in Artificial Intelligence at School of Information Technology & Engineering, College of Technology and Built Environment.

Dr. Adane Letta Mamuye		
Advisor	*Signature*	*Date*
Internal Examiner	*Signature*	*Date*
External Examiner	*Signature*	*Date*
School Dean / Department Head	*Signature*	*Date*

Abstract

The identification of gene promoter regions is crucial for understanding transcriptional regulation, yet computational methods often struggle to effectively integrate the diverse biological signals involved. Existing approaches typically focus on a single data modality, such as the DNA sequence, or employ simple fusion techniques that fail to leverage explicit biological knowledge. To address these limitations, we present **BWAF-Net**, a novel multi-modal deep learning framework for the identification of human promoters.

BWAF-Net integrates three data streams: DNA sequences processed by a **Transformer** to capture long-range dependencies; gene regulatory context from 36 tissue-specific networks modeled by a **Graph Attention Network (GAT)**; and explicit domain knowledge in the form of 11 quantified biological motif counts (priors). The framework's central innovation is the **Biologically Weighted Attention Fusion (BWAF)** layer, which uses the biological priors to learn dynamic attention weights that modulate the fusion of the sequence and network representations.

Evaluated on a balanced dataset of 40,056 human promoter and non-promoter sequences, BWAF-Net achieved outstanding performance, with **99.87% accuracy**, **99.99% AUC-ROC**, and **100% precision** on the held-out test set. The proposed framework significantly outperformed a replicated state-of-the-art, sequence-only baseline as well as a series of ablated models. Our ablation studies confirm that naive feature concatenation is a suboptimal fusion strategy, validating the necessity of the intelligent BWAF mechanism. By providing a framework that is highly accurate, parameter-efficient, and interpretable, this work presents a significant advance in multi-modal AI for regulatory genomics.

Keywords: Promoter Identification, Multi-Modal Deep Learning, Attention Fusion, Transformer, Graph Attention Network (GAT), Gene Regulatory Networks, Biological Priors, Explainable AI (XAI).

Acknowledgment

First and foremost, I thank God for His guidance and strength throughout this journey. I extend my deepest gratitude to my advisors, **Dr. Adane Letta Mamuye, Dr. Fantahun Bogale Gereme, Dr. Beakal Gizachew Assefa, and Dr. Natnael Argaw Wondimu**, for their invaluable support, insights, and mentorship. I am also grateful to my classmates for their encouragement and discussions that enriched my research. Lastly, I sincerely thank my family for their unwavering love, patience, and support.

Dedication

To my family...

To my academic advisor, Dr. Adane Letta Mamuye....

And to myself

Table of Contents

List of Acronyms

Acc	Accuracy
AI	Artificial Intelligence
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
BERT	Bidirectional Encoder Representations from Transformers
Bi-GRU	Bidirectional Gated Recurrent Unit
bp	Base Pairs
BRE	TFIIB Recognition Element
BWAF	Biologically Weighted Attention Fusion
CAGE	Cap Analysis of Gene Expression
ChIP-seq	Chromatin Immunoprecipitation Sequencing
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DL	Deep Learning
DNA	Deoxyribonucleic Acid
DSR	Design Science Research
ENSG	Ensembl Gene ID Prefix
EPI	Enhancer-Promoter Interaction
F1	F1-Score
FN	False Negative
FP	False Positive

GAT	Graph Attention Network
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GRN	Gene Regulatory Network
GRU	Gated Recurrent Unit
GTEx	Genotype-Tissue Expression
GTF	Gene Transfer Format
HMM	Hidden Markov Model
Inr	Initiator Element
LCCRF	Local-context Conditional Random Fields
LIONESS	Linear Interpolation to Obtain Network Estimates for Single Samples
LSTM	Long Short-Term Memory
MHA	Multi-Head Self-Attention
ML	Machine Learning
MLP	Multi-Layer Perceptron
mRNA	Messenger RNA
PANDA	Passing Attributes between Networks for Data Assimilation
PPI	Protein-Protein Interaction
Prec	Precision
PSSM	Position-Specific Scoring Matrix
PWM	Position Weight Matrix
RAM	Random Access Memory
RBF	Radial Basis Function
Rec	Recall
RF	Random Forest

RNA	Ribonucleic Acid
RNN	Recurrent Neural Network
SHAP	SHapley Additive exPlanations
Spec	Specificity
SVM	Support Vector Machine
TF	Transcription Factor
TFBS	Transcription Factor Binding Site
TN	True Negative
TP	True Positive
TSS	Transcription Start Site
XAI	Explainable Artificial Intelligence
XGBoost	eXtreme Gradient Boosting

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Gene regulation is a fundamental biological process that orchestrates the precise expression of genes, defining cellular identity, function, and response to stimuli. At the heart of this process lie **promoters**, specific DNA regions that serve as the primary control points for initiating gene transcription [1–6]. The accurate identification and characterization of these promoters are therefore essential for deciphering the complex logic of gene regulatory networks, understanding developmental pathways, and elucidating the molecular basis of human diseases [7–12]. The function of a promoter is not determined by its DNA sequence in isolation but is profoundly influenced by a multi-layered regulatory landscape, including the three-dimensional chromatin architecture, the binding of transcription factors (TFs), and tissue-specific interactions within gene regulatory networks (GRNs) [5, 13–18].

Historically, understanding promoter function has been driven by a combination of experimental and computational technologies. Experimental techniques such as chromatin immunoprecipitation sequencing (ChIP-seq) have been instrumental in mapping TF binding sites, while methods like Cap Analysis of Gene Expression (CAGE) provide precise locations of transcription start sites (TSSs) [19–21]. Computationally, early efforts to predict promoters relied on identifying conserved sequence motifs, such as the TATA box, through statistical methods like Position Weight Matrices (PWMs) and Hidden Markov Models (HMMs) [21]. While foundational, these approaches often struggled with the sequence variability and contextual dependencies inherent in complex genomes.

The advent of high-throughput sequencing has generated a wealth of multi-modal biological data, offering an unprecedented opportunity to model this complexity. In response, the field of computational genomics has increasingly

turned to Artificial Intelligence (AI), particularly deep learning [22–26]. Advanced architectures like Transformers and Graph Attention Networks (GATs) have demonstrated significant success in learning intricate patterns from sequence and network data, respectively [24, 27–31]. Transformers, with their self-attention mechanism, excel at capturing long-range dependencies in DNA [22, 29–32], while GATs are adept at modeling the relational information in biological networks [28]. These technologies form the technical foundation upon which modern, more sophisticated promoter prediction models are built.

1.2 Motivation

our motivation for this research stems from a recognized gap between the capabilities of modern AI and the complexities of biological systems. While observing the remarkable success of deep learning models in various domains, We noted that in genomics, many state-of-the-art approaches still operate on a single data modality, typically the DNA sequence. This seemed insufficient, as biological reality is inherently multi-modal; a promoter's function is not just a property of its sequence but is deeply embedded in its cellular and network context. We was particularly interested in the challenge of integrating the linear, syntactic information of DNA with the relational, contextual information of gene regulatory networks.

Furthermore, We observed that many deep learning applications discard decades of accumulated biological knowledge in favor of a purely data-driven, end-to-end learning approach. This felt like a missed opportunity. Our work is driven by the conviction that a more powerful and trustworthy model could be built by explicitly incorporating this hard-won biological knowledge such as the roles of specific regulatory motifs into the model's architecture. We wanted to design a framework that learns from data but is also guided by established biological principles. This desire to bridge the gap between purely data-driven AI and biologically-informed modeling, specifically for the crucial task of promoter identification, is the core motivation for this thesis.

1.3 Problem Statement

The accurate identification of human gene promoters is a critical yet unsolved challenge in computational genomics. In recent years, significant efforts have been made using advanced deep learning models to address this problem. Transformer-based architectures, such as 'msBERT-Promoter', have achieved high accuracy by leveraging powerful self-attention mechanisms to learn from DNA sequences alone [9]. However, this single-modality focus represents a key technical challenge, as

it inherently ignores the rich contextual information provided by gene regulatory networks, which are known to be crucial for tissue-specific gene expression[33–35]. On the other hand, Graph Neural Networks (GNNs) have been successfully applied to model these biological networks for tasks like cancer driver gene identification, but they often lack a sophisticated mechanism for integrating fine-grained sequence features [28, 34].

Attempts to create multi-modal models often face a second technical challenge: suboptimal fusion. The most common strategy is simple feature concatenation, where learned representations from different branches are merged and passed to a classifier. As will be demonstrated empirically in Chapter 4 of this thesis, this naive approach can paradoxically degrade performance, as the strong signal from low-dimensional, high-value features (like biological priors) can be diluted by higher-dimensional, more complex features from deep learning branches. A third challenge, therefore, is the lack of a fusion methodology that can intelligently and dynamically weigh the contributions of different data modalities. Specifically, there is a significant research gap in methods that use explicit, quantitative biological knowledge (such as motif counts) not just as another input feature, but as an active controller to guide the fusion process itself. This limitation results in a performance ceiling for current models and hinders their biological interpretability. This thesis proposes a novel architecture designed to directly address these technical challenges by introducing a prior-guided dynamic fusion mechanism.

1.4 Research Questions

To address the problems outlined above, this research is guided by two primary questions:

1. How can a multi-modal deep learning architecture be designed to synergistically integrate distinct data types specifically DNA sequences and multi-tissue gene regulatory networks for human promoter identification?

2. To what extent does using explicit biological priors to dynamically guide the fusion of learned features improve performance and interpretability over static fusion methods?

1.5 Objectives of the Study

1.5.1 General Objective

The general objective of this research is to develop a multi-modal deep learning framework, BWAF-Net, for improving the accuracy of human promoter identification.

1.5.2 Specific Objectives

To achieve the general objective, this research pursued the following specific objectives:

1. To design and implement a novel deep learning architecture that integrates a Transformer branch for sequence analysis, a Graph Attention Network (GAT) branch for network context, and a Biologically Weighted Attention Fusion (BWAF) layer guided by explicit biological priors.

2. To empirically evaluate the performance of the proposed BWAF-Net model against a state-of-the-art baseline and a series of ablated models to validate the contribution of each architectural component and the novel fusion mechanism.

3. To analyze the trained model to gain interpretable insights into its decision-making process, linking its learned fusion strategy back to known biological principles.

1.6 Contribution of the Study

This thesis presents contributions to the fields of Artificial Intelligence and Computational Genomics. The primary methodological contribution is the implementation of the **Biologically Weighted Attention Fusion (BWAF)** mechanism. This represents an architectural paradigm for multi-modal learning where explicit domain knowledge is used as an active controller of information flow between specialized deep learning modules.

For the field of computational genomics, this work contributes **BWAF-Net**, a computational framework for human promoter identification. The model achieves an improved performance (99.87% accuracy and 100% precision on the test set) while being significantly more parameter-efficient than comparable state-of-the-art ensemble models. Furthermore, this thesis provides empirical evidence on the nature of multi-modal fusion in genomics. The ablation studies reveal that

while biological priors are highly predictive, naive fusion strategies like simple concatenation can degrade performance. The success of the BWAF mechanism provides a validated solution to this challenge, demonstrating that an intelligent, prior-guided fusion strategy is essential for unlocking the full synergistic potential of multi-modal data.

1.7 Scope and Delimitation

1.7.1 Scope

The scope of this study is to develop a deep learning framework for promoter prediction within the human genome. The model integrates three specific data types:

- **Sequence Data:** 2000bp DNA sequences upstream of the TSS of protein-coding genes from the GRCh38 human reference genome.
- **Network Data:** 36 normal human tissue-specific GRNs sourced from the GRAND database.
- **Biological Priors:** Quantitative counts of 11 well-characterized regulatory DNA motifs.

The study utilizes publicly available human genomic and regulatory datasets and focuses on developing an interpretable, multi-modal architecture for binary classification (promoter vs. non-promoter).

1.7.2 Delimitations

The research is confined to the human genome and focuses exclusively on promoters of protein-coding genes located on the primary chromosomes (1–22, X, Y). Other regulatory elements such as enhancers, silencers, or insulators are not considered. Additionally, only the three specified data types were used; other omics data like proteomics or metabolomics are excluded. The study is computational in nature, and no experimental (wet-lab) validation of the model's predictions is conducted. The model's applicability to other species or to predicting promoter strength remains outside the present scope.

1.8 Thesis Outline

The remainder of this thesis is organized as follows. **Chapter 2** provides a comprehensive review of the literature, covering the biological foundations of promoters

and the evolution of computational prediction methods, thereby establishing the research gap this thesis addresses. **Chapter 3** details the complete methodology, including the Design Science Research approach, the curation and preprocessing of all data modalities, the mathematical formulation of the BWAF architecture, and the full experimental design. **Chapter 4** presents and analyzes the experimental results, detailing the performance of the full BWAF model, its comparison against the replicated state-of-the-art baseline, and the insights gained from the ablation studies and interpretability analysis. Finally, **Chapter 5** summarizes the key findings and contributions of the thesis and proposes promising directions for future research.

Chapter 2

Literature Review

This chapter establishes the theoretical and empirical foundation for the thesis by providing a comprehensive review of the scientific literature. We begin by defining the biological significance of promoters to contextualize the prediction challenge, then trace the evolution of computational methods from early statistical approaches to classical machine learning. The review then transitions to the modern deep learning era, surveying the impact of foundational architectures like Convolutional and Recurrent Neural Networks before focusing on state-of-the-art models, including Transformers for sequence analysis and Graph Attention Networks (GATs) for modeling regulatory context. Finally, we critically analyze the frontier of multi-modal data fusion and the integration of biological priors into AI models. This culminates in a precise articulation of the specific research gap this thesis addresses: the lack of a framework that uses quantitative biological knowledge to dynamically guide the fusion of learned representations from distinct deep learning branches, thereby justifying the need for the novel Biologically Weighted Attention Fusion (BWAF) model.

2.1 Biological Foundations of Gene Regulation

To understand the challenge of promoter identification, it is essential to first review the fundamental principles of molecular biology that govern how genetic information is transcribed and expressed. This section provides a background on the key biological molecules and the central process of gene transcription.

2.1.1 Key Biological Molecules and the Genome

The blueprint of life is stored within **Deoxyribonucleic Acid (DNA)**, a double-helix molecule composed of a sequence of four chemical bases, or **nucleotides**: Adenine (A), Guanine (G), Cytosine (C), and Thymine (T). The entire set of

DNA instructions in an organism is known as its **genome** [36, 37]. Genomics is the field dedicated to studying the structure, function, evolution, and mapping of genomes[38].

Specific segments of DNA that contain the instructions for building functional products are called **genes**. These products are typically **proteins**, which are complex molecules that perform a vast array of tasks within a cell, or functional **Ribonucleic Acid (RNA)** molecules. The journey from a gene's DNA sequence to a functional protein is the core of gene expression and follows the central dogma of molecular biology: DNA → RNA → Protein[39–43].

2.1.2 Gene Transcription and Expression

Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product. The first and most highly regulated step of this process is **transcription**, where a specific segment of DNA is copied into an RNA molecule by an enzyme called RNA Polymerase[44, 45].

The transcription process, as illustrated in Figure 2.1, is initiated at a specific region of DNA upstream of a gene known as the **promoter**. This region acts as a binding site for RNA Polymerase and a suite of proteins called **transcription factors (TFs)**. TFs recognize and bind to short, specific DNA sequences within the promoter called **motifs** or transcription factor binding sites (TFBSs). The coordinated binding of these TFs helps to recruit RNA Polymerase to the promoter and initiate transcription at a precise location known as the **Transcription Start Site (TSS)**. Once initiated, the RNA Polymerase moves along the DNA, synthesizing a complementary RNA strand. This RNA molecule is then processed and, if it codes for a protein (as messenger RNA or mRNA), is translated into a protein by the cell's machinery (ribosomes)[4, 16, 46].

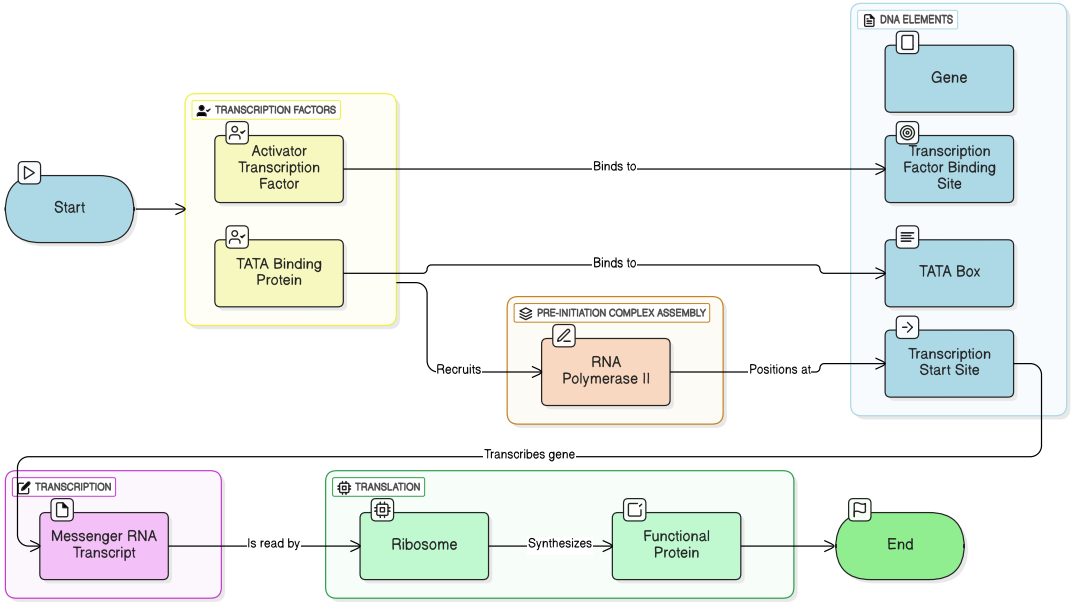

Figure 2.1: A schematic of eukaryotic gene transcription initiation. Transcription factors (TFs) bind to specific DNA motifs within the promoter region. This complex recruits RNA Polymerase II, which then initiates the synthesis of an RNA transcript from the DNA template starting at the Transcription Start Site (TSS)[46].

The regulation of this process is incredibly complex. The presence, combination, and spacing of different motifs within a promoter determine which TFs can bind, how strongly they bind, and ultimately, the rate of transcription[47]. This regulatory grammar is what our computational model aims to learn.

2.2 Traditional Computational Methods for Promoter Prediction

Before the widespread adoption of machine learning, computational promoter prediction was dominated by methods that attempted to model the known biological features of promoters directly through rule-based or statistical approaches. These early methods were foundational but were ultimately limited by their inability to capture the full complexity and variability of regulatory sequences[48].

2.2.1 Rule-Based and Consensus Sequence Methods

The earliest computational approaches were essentially "handcrafted rule" systems based on direct biological observations. Researchers knew that certain DNA

motifs, such as the TATA box (consensus sequence `TATAAT`) in prokaryotes or the TATA box (consensus `TATAAAA`) and Initiator element (Inr) in eukaryotes, were frequently found at specific distances from the TSS[49].

These methods involved scanning a genome for occurrences that matched a strict **consensus sequence** or a set of predefined rules about the presence and spacing of these motifs. While effective for identifying promoters with highly conserved, canonical structures, these rule-based systems suffered from extremely high false-positive and false-negative rates. They lacked the flexibility to account for the significant sequence variability observed in functional, non-canonical promoters and could not easily integrate information from weaker, less-conserved motifs [50, 51].

2.2.2 Statistical Pattern-Matching Methods

To address the rigidity of consensus sequences, statistical methods were developed to model the variability of promoter elements probabilistically. These approaches moved from deterministic rules to scoring sequences based on their likelihood of being a functional site[49].

Position Weight Matrices (PWMs)

A significant improvement over consensus matching was the **Position Weight Matrix (PWM)**, also known as a Position-Specific Scoring Matrix (PSSM). A PWM is a matrix that stores the probability (or log-odds score) of observing each of the four nucleotides (A, C, G, T) at each position within a known motif [49, 52].

Given an alignment of N known binding site sequences of length L, the frequency matrix F is first constructed, where $F_{b,j}$ is the count of nucleotide b at position j. The probability matrix P, where $P_{b,j} = F_{b,j}/N$, is then converted into a log-odds PWM, M. The score $M_{b,j}$ for nucleotide b at position j is given by:

$$M_{b,j} = \log_2 \left(\frac{P_{b,j}}{p_b} \right) \tag{2.1}$$

where p_b is the background probability of observing nucleotide b in the genome (e.g., 0.25). To score a new sequence $S = s_1 s_2 ... s_L$, the scores are summed across its length:

$$\text{Score}(S) = \sum_{j=1}^{L} M_{s_j,j} \tag{2.2}$$

A sequence is predicted to be a promoter element if its score exceeds a predefined threshold. While PWMs account for nucleotide frequency at each position, they

make a strong simplifying assumption of positional independence, meaning they cannot model dependencies between nucleotides at different positions within the motif [53]. This limitation contributes to their high false-positive rates.

Hidden Markov Models (HMMs)

Hidden Markov Models (HMMs) were introduced to model the sequential structure of genomic regions, including promoters. An HMM is a statistical model that assumes a system is a Markov process with unobserved ("hidden") states. In genomics, these hidden states can represent different functional regions, such as "promoter," "exon," "intron," or "intergenic."[54]

An HMM is defined by a set of states, a set of emission probabilities (the probability of observing a nucleotide given a certain state), and a set of transition probabilities (the probability of moving from one state to another). For promoter prediction, one could design an HMM with states representing the core promoter, proximal elements, and flanking regions. The model can then calculate the most likely path of hidden states for a given DNA sequence using algorithms like the Viterbi algorithm. While HMMs can model dependencies between adjacent states, their ability to capture the long-range interactions common in eukaryotic promoters is limited, and they often struggle to generalize across diverse genomic contexts without complex, manually designed state architectures [55].

Advanced Statistical Models

Later methods built upon these statistical foundations. For instance, TSSFinder integrated Local-context Conditional Random Fields (LCCRFs), which are discriminative models that can handle more complex dependencies than generative models like HMMs, to better capture the positional context of core promoter elements [56]. Other approaches incorporated thermodynamic features, such as DNA duplex stability, into their statistical frameworks to improve prediction in specific organisms like archaea [57]. Despite these improvements, all of these traditional methods are fundamentally limited by their reliance on pre-defined features and statistical assumptions, a challenge that machine learning and deep learning approaches would later address.

2.3 The Rise of Machine Learning in Promoter Prediction

The limitations of statistical pattern-matching methods prompted a shift towards classical Machine Learning (ML) approaches. These models represented a sig-

nificant step forward by introducing algorithmic learning from data, but were characterized by their reliance on a two-step process: manual feature engineering followed by classification[58–60].

2.3.1 Early ML Models

Early ML approaches treated promoter prediction as a standard binary classification problem. The core of this paradigm was the process of **handcrafted feature engineering**, which involves a researcher using their domain expertise to manually design and extract a set of numerical features from the raw DNA sequence. These features, which could include nucleotide frequencies, k-mer counts, DNA physicochemical properties (like melting temperature or bendability), or motif hit scores, were compiled into a fixed-length feature vector for each sequence. This vector was then used as input for a classical classifier to learn a decision boundary between promoter and non-promoter classes[61, 62].

Support Vector Machines (SVMs)

Support Vector Machines (SVMs) were a highly popular choice for this task [62, 63]. An SVM is a supervised learning algorithm that aims to find an optimal hyperplane that best separates data points belonging to different classes in a high-dimensional space[61, 64]. For a given set of training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector and $y_i \in \{-1, 1\}$ is the class label, the SVM seeks to find a hyperplane defined by a weight vector $\mathbf{w}$ and a bias b.

The decision function is given by:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b) \tag{2.3}$$

The optimal hyperplane is the one that maximizes the margin, or the distance between the hyperplane and the nearest data points (the support vectors) from each class. This is formulated as a constrained optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \quad \text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i \tag{2.4}$$

To handle non-linearly separable data, SVMs employ the "kernel trick," which uses a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ to map the data into a higher-dimensional feature space where a linear separation is possible, without explicitly computing the transformation ϕ. Common kernels in genomics include the linear, polynomial, and Radial Basis Function (RBF) kernels.

Random Forests (RFs)

Random Forests (RFs) were another widely used method, valued for their robustness and ability to handle high-dimensional feature spaces [60, 63]. An RF is an ensemble learning method that constructs a multitude of decision trees at training time.

During training, for each of the B trees in the forest, a bootstrap sample (a random sample with replacement) is drawn from the training data. Then, at each node of the tree, a random subset of m features (where $m \ll d$) is selected, and the best split among these features is used to partition the data. This dual randomization (bagging of samples and random feature subspaces) helps to decorrelate the individual trees[62].

For a new input $\mathbf{x}$, the final prediction is made by aggregating the predictions of all B trees. For classification, this is typically done by majority vote:

$$f(\mathbf{x}) = \text{argmax}_c \sum_{b=1}^{B} I(f_b(\mathbf{x}) = c) \tag{2.5}$$

where $f_b(\mathbf{x})$ is the prediction of the b-th tree and $I(\cdot)$ is the indicator function. The strength of RFs lies in their ability to reduce the variance of individual decision trees, making them less prone to overfitting.

Despite the advances offered by models like SVMs and RFs, these early ML methods were fundamentally constrained by their reliance on the initial feature engineering step. The quality of the handcrafted features directly limited the model's performance, and these features often failed to capture complex, higher-order dependencies or long-range interactions within the DNA sequence a limitation that would later be addressed by end-to-end deep learning models [58].

2.3.2 Recent Advances in ML

More recent research continues to leverage classical ML models but often with more sophisticated feature sets or in combination with deep learning components. The focus has increasingly shifted towards improving interpretability and assessing cross-species or cross-condition generalizability. For example, `MLDSPP` [65] combines DNA structural attributes with sequence encoding in an XGBoost model, demonstrating strong performance across multiple bacterial genomes. Similarly, other studies have used ML pipelines for predicting promoter strength from synthetic libraries [66] or for identifying specific epigenetic marks like 5-methylcytosine within promoter regions [67].

A recurring theme in these recent studies is the challenge of generaliz-

ability. While models often perform well on their specific benchmark datasets, performance can degrade when applied to different species or even different tissues within the same species, as highlighted by comparative studies [68, 69]. This underscores the need for models that can learn more fundamental and transferable regulatory principles, a key motivation for the multi-modal approach proposed in this thesis. The table below summarizes several recent studies, focusing on their generalizability assessments.

Table 2.1: Summary of Machine Learning-Based Promoter Prediction Studies with a Focus on Generalizability

Study	Methodology Summary	Generalizability Assessment
MLDSPP [65]	Hybrid model combining DNA structural attributes and one-hot sequence encoding using XGBoost; incorporated explainable AI for interpretability.	Demonstrated robust performance across 12 bacterial genomes, indicating strong intra-prokaryotic transferability.
Zhao et al. (2021) [66]	Built a synthetic promoter library to train an XGBoost-based predictor for promoter strength.	Mentioned cross-context applicability, but lacked comprehensive validation on native biological datasets.
BERT-Promoter [27]	Applied BERT embeddings with SHAP-based feature selection for promoter classification.	Achieved strong performance on benchmark datasets; however, cross-species testing was not conducted.
Bhandari et al. (2021) [68]	Comparative analysis of ML and DL models, identifying CNNs as top performers.	Observed a drop in accuracy during cross-species evaluations due to differences in sequence motifs.
DeepPHiC [69]	Deep learning framework for predicting promoter-centered chromatin interactions from Hi-C data.	Generalized well across tissue types with sufficient data; struggled with sparse datasets.
Nguyen et al. (2021) [67]	Trained ML models to predict genome-wide 5mC methylation sites in promoter regions.	Performed well on internal test sets; species-specific generalization was not extensively addressed.

2.4 Deep Learning for Sequence-Based Promoter Prediction

The limitations of classical machine learning, particularly its reliance on manual feature engineering, motivated the adoption of the **deep learning (DL)** paradigm. Unlike their predecessors, deep learning models are capable of end-to-end learning, automatically discovering hierarchical feature representations directly from raw input data, such as one-hot encoded DNA sequences. This section details the progression of foundational DL architectures specifically Convolutional and Recurrent Neural Networks and culminates in the current state-of-the-art, Transformer-based models, explaining their mechanisms and applications in regulatory genomics[24, 70].

2.4.1 Foundational Architectures: CNNs and RNNs

Deep learning was first successfully applied to promoter prediction through two main architectural families: Convolutional Neural Networks (CNNs), which excel at detecting local spatial patterns, and Recurrent Neural Networks (RNNs), which are designed to model sequential dependencies[12, 71, 72].

Convolutional Neural Networks (CNNs) for Motif Detection

A Convolutional Neural Network (CNN) is a class of neural network particularly adept at identifying local, position-invariant patterns. In genomics, this capability is leveraged to automatically detect sequence motifs, akin to how a Position Weight Matrix (PWM) operates but in a more flexible and data-driven manner [71, 73, 74].

A typical CNN layer processes an input sequence (e.g., a one-hot encoded matrix of shape $L \times 4$) by sliding a set of learnable filters (or kernels) across it. Each filter is a small matrix of weights trained to recognize a specific pattern, such as a TFBS. The convolution operation at each position involves computing the dot product between the filter and the corresponding segment of the input sequence. For a 1D convolution on a sequence X with a filter W of length k, the output feature map c_i at position i is:

$$c_i = f\left(\sum_{j=0}^{k-1} W_j \cdot X_{i+j} + b\right) \tag{2.6}$$

where f is a non-linear activation function (typically ReLU) and b is a bias term. By using multiple filters, a CNN can learn to detect a rich vocabulary of motifs simultaneously. The output of a convolutional layer is often passed through a pooling layer (e.g., max pooling) to create a down-sampled representation that captures the presence of detected motifs regardless of their exact position. Models like DeepBend demonstrated this principle by using a CNN to predict DNA bendability, a structural property relevant to promoter function, directly from sequence [75].

Recurrent Neural Networks (RNNs) for Sequential Context

While CNNs are powerful for local motif detection, they have a limited receptive field. Recurrent Neural Networks (RNNs), and their more advanced variants like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, were developed to model sequential data and capture long-range dependencies[71, 74].

An RNN processes a sequence one element at a time, maintaining a hidden state h_t that serves as a memory of the preceding elements. The hidden state at timestep t is a function of the current input x_t and the previous hidden state h_{t-1}:

$$h_t = f(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{2.7}$$

where W_{hh} and W_{xh} are learned weight matrices. Standard RNNs suffer from the vanishing gradient problem, making it difficult to learn long-range dependencies. LSTMs and GRUs overcome this with a gating mechanism a series of neural networks that learn to control the flow of information, deciding what to remember from the past and what to forget. In promoter prediction, RNNs are used to model the syntax of regulatory elements, understanding that the functional impact of a motif can depend on its context and its distance from other elements [76].

Hybrid CNN-RNN Architectures

Recognizing the complementary strengths of these architectures, many successful promoter prediction models have adopted hybrid designs. In these models, a CNN layer is typically used first to extract local motif-like features from the raw sequence. The resulting feature map, which represents a higher-level abstraction of the sequence, is then fed into an RNN (often a Bidirectional LSTM or Bi-GRU) to model the sequential dependencies between these learned motifs. This synergy allows the model to learn both "what" (motifs, via CNN) and "where/in what order" (context, via RNN) the important regulatory signals are. For example, a model by Amjad et al. [77] leveraged multi-scale convolutional filters followed by Bi-GRUs to effectively capture both local and long-range patterns for promoter identification. Similarly, Zhou et al. [78] developed DeeReCT-TSS by integrating CNNs with an attention mechanism to prioritize regulatory features learned from the sequence. Table 2.2 provides a comparative summary of several such deep learning models.

2.4.2 State-of-the-Art: Transformer-Based Models

While RNNs can model long-range dependencies, they are inherently sequential and can struggle to parallelize computations. Transformer-based architectures, originally developed for natural language processing, have emerged as the state-of-the-art for sequence analysis by overcoming these limitations through the self-attention mechanism [80]. Transformers operate without an inherent locality bias, allowing every position in a sequence to directly attend to every other position, making them exceptionally well-suited for modeling the complex, long-distance

Table 2.2: Comparative Analysis of Foundational Deep Learning Models for Regulatory Element Prediction

Study	Architecture	Seq. Length	Input Rep.	Motif Discovery	Target Type
Bhandari et al. (2021) [77]	CNN + Bi-GRU	600 bp	One-hot	Yes (Convolution)	Promoter
Zeng et al. (2021) [78]	CNN + Attention	1,000 bp	One-hot + CAGE	Yes (Saliency Maps)	TSS (Promoter)
Kaur et al. (2022) [79]	Hybrid CNN-LSTM	300 bp	Multi-Feature (k-mer, word2vec, shape)	Yes (Convolution)	Enhancer
Kaur et al. (2022) [29]	Hybrid CNN + Transformer	3000 bp and 2000bp respectively	dna2vec for sequence embedding, 2-layer CNN for features, Transformer for high-level extraction.	Yes (Convolution and Transformer)	Enhancer-promoter interaction prediction

interactions between regulatory elements in genomic DNA.

A key innovation in this area is the adaptation of the BERT model for genomics, exemplified by DNABERT [81], which is pre-trained on vast amounts of DNA sequence data. This pre-training allows the model to learn a fundamental language of DNA. Subsequent models then fine-tune this pre-trained knowledge for specific downstream tasks. The `BERT-Promoter` model, for instance, fine-tunes a BERT model for promoter classification and uses SHAP for interpretability [27].

Building on this, `msBERT-Promoter` [82] established a new benchmark by introducing two key innovations. The first is a **multi-scale tokenization strategy**, where sequences are processed not as individual nucleotides but as k-mers of varying lengths (k=3 to 6). This allows the model to capture both fine-grained motifs and broader sequence patterns. The second is a **two-stage prediction framework**: the first stage identifies promoters versus non-promoters, and the second stage classifies those promoters as strong or weak. To achieve this, the authors train an **ensemble of DNABERT models**, one for each k-mer representation, and combine their predictions via soft-voting. Given its strong performance, advanced architecture, and its status as a sequence-only ensemble, we selected the first stage of this methodology as the primary state-of-the-art baseline to replicate and compare against our proposed multi-modal framework.

The power of Transformers has also been applied to related regulatory problems. In enhancer-promoter interaction (EPI) modeling, models like `EPI-Trans` [29] and `TF-EPI` [83] use Transformers to jointly model enhancer and promoter sequences, capturing the contextual information necessary for predicting physical or functional links. Similarly, `Proformer` [30] uses a specialized Transformer architecture to predict gene expression levels directly from promoter sequences. Table 2.3

provides a comparative summary of these advanced approaches.

Table 2.3: Summary of Recent Transformer-Based Models for Promoter and Enhancer Prediction

Model	**Model Type**	**Context Modeling**	**Interpretability**	**Performance Improvement**
EPI-Trans [29]	CNN + Transformer	Fused enhancer/promoter CNN outputs sent into single Transformer	Not reported	Outperforms EPI-Mind via joint attention modeling
TF-EPI [83]	Transformer + TextCNN	6-mer tokenization with long-range attention	Attention weights used to find motifs	Outperforms EPIANN and EPI-DLMH across all cell lines
BERT-Promoter [27]	Pre-trained BERT	Bidirectional encoding of sequences	SHAP explanations for strong/weak promoters	85.5% accuracy on strong promoters
TSSNote-CyaPromBERT [84]	BERT / XLNet	Context-aware modeling of prokaryotic promoters	AT-rich -10 box focus via attention	AUROC of 0.977; F1 of 0.93
Proformer [30]	Macaron Transformer	Local + global modeling via ODE-inspired layers	Not reported	Outperforms CNNs using multi-heads and mask filling
PLPMpro [85]	Prompt-Learning PLM	Not detailed	Partial (semantics captured)	Beats deep learning and PLM baselines
Chromoformer [86]	Hierarchical Transformer	Histone code + 3D interactions via attention	Attention on downstream regions	Outperforms CNNs/RNNs across gene expression tasks

2.5 Modeling Regulatory Context with Graph Attention Networks

While Transformer-based models excel at deciphering the linear grammar of DNA sequences, they are not inherently designed to model the complex, non-linear relationships that constitute biological networks. Gene regulation is not a solitary process; it occurs within a vast, interconnected system of interactions between genes, transcription factors (TFs), and other regulatory elements. To capture this critical contextual information, the field has increasingly turned to **Graph Neural Networks (GNNs)**, a class of deep learning models specifically designed to learn from graph-structured data [87]. GNNs represent biological entities (like genes) as nodes and their interactions (like regulation) as edges, allowing them to learn features based on the topological structure of the network.

Among various GNN architectures, the **Graph Attention Network (GAT)** [88] has proven particularly effective for regulatory genomics. Unlike simpler Graph Convolutional Networks (GCNs) that aggregate information from neighboring nodes uniformly, GATs introduce an attention mechanism. This allows the model to learn to assign different levels of importance to different neighbors when updating a node's representation. This is biologically highly intuitive; a

gene's function is often more strongly influenced by a few key regulatory partners than by all its neighbors equally. This ability to dynamically weigh interactions makes GATs a powerful tool for modeling the nuanced and context-dependent nature of gene regulatory networks[89–91].

The power of GATs has been demonstrated across a range of complex problems in regulatory genomics. A seminal example is **GraphReg**, which integrates 3D chromatin interaction data (e.g., from Hi-C) and epigenomic signals into a GAT to predict gene expression [92]. By modeling long-range enhancer-promoter interactions as a graph, GraphReg significantly outperformed CNN-based approaches, highlighting the importance of network context. The GAT framework has also been adapted to specifically model enhancer-promoter interaction (EPI) graphs, as seen in **GATv2EPI**, which uses a dynamic graph attention mechanism to capture complex regulatory patterns across different chromosomes and tissues [28].

Furthermore, GATs have been successfully applied to the direct prediction of TF-target gene interactions. **GraphTGI** framed this task as a link prediction problem on a TF-gene knowledge graph, using a GAT-based autoencoder to learn powerful node embeddings that incorporated both network structure and biological sequence features [93]. Expanding on this, **PPRTGI** enhanced a GNN with Personalized PageRank to improve TF-target predictions [94]. The versatility of GNNs is also evident in their application to networks derived from single-cell RNA-seq data, where models like **GNNLink** [95] and **GMFGRN** [96] infer regulatory links from dynamic gene expression states. Even methods that primarily focus on sequence have begun to incorporate graph-based thinking; for instance, Tenekeci and Tekir [97] demonstrated improved promoter identification by constructing graphs from sequence fragments and integrating them with EPI networks.

The demonstrated success of these diverse, state-of-the-art models in leveraging graph attention mechanisms to decode complex regulatory relationships provides a strong rationale for our architectural choices. The ability of GATs to learn context-specific node representations from tissue-specific networks is precisely what is needed to complement the sequence-focused analysis of the Transformer branch. Therefore, the inclusion of a GAT branch in our BWAF framework is a well-justified strategy to ensure our model captures the critical network context in which promoters function.

Table 2.4: Summary of Recent GNN-Based Methods for Gene Interaction and Regulatory Modeling

Model	Graph Type	Input Graph	Topology Representation
GATv2EPI [28]	GAT	Enhancer-Promoter network	Nodes are promoters/enhancers; edges denote interactions; chromosome-level subgraphs
GraphReg [92]	GAT	3D chromatin interactions	Genomic bins as nodes; E-E and E-P edges based on Hi-C/HiChIP/Micro-C/HiCAR
GraphTGI [93]	GAT Autoencoder	TF-gene network	Nodes: TFs and genes; edges: known interactions; includes sequential and chemical features
PPRTGI [94]	Personalized PageRank GNN	TF-gene network	Nodes: TFs and genes; bilinear decoding of edge scores
GNNLink [95]	GCN	scRNA-seq-derived GRN	Nodes: genes; edges: inferred links from expression data
GMFGRN [96]	GNN + Matrix Factorization	scRNA-seq-derived GRN	Nodes: genes; edges: TF-gene dependencies learned via graph embeddings
Tenekeci & Tekir [97]	GCN	Enhancer-Promoter network	Nodes: word-embedded sequences; edges: EPI interactions
AutoGAT [98]	GAT Autoencoder	TF-gene network	Knowledge graph with TFs and genes as nodes, interaction edges, with sequence and chemical node features

2.6 The Frontier: Multi-Modal Fusion and Biologically Informed AI

The recognition that promoter function is governed by more than just the primary DNA sequence has pushed the field of computational genomics towards a new frontier: **multi-modal deep learning**. This paradigm acknowledges that a more complete understanding of gene regulation can only be achieved by integrating heterogeneous but complementary data types. A comprehensive review by Stahlschmidt, Ulfenborg, and Synnergren [99] highlights this trend, detailing how fusing genomics, epigenomics, and transcriptomics can create more robust and accurate predictive models. The challenge, however, lies not just in using multiple data types, but in the sophistication of the fusion strategy itself.

2.6.1 Architectures for Multi-Modal Data Fusion

Early and straightforward approaches to multi-modal integration often rely on **simple feature concatenation**. In this strategy, features are learned independently from each modality using specialized neural network branches, and the resulting feature vectors are simply concatenated before being passed to a final classifier. For instance, the `HMPI` framework combines features from a CNN processing DNA sequence with features from a DenseNet processing structural profiles via late-stage concatenation [100]. Similarly, Kaur, Chauhan, and Aggarwal [101]

combined k-mer, word2vec, and DNA shape features by concatenating them into a unified input representation for a hybrid CNN-LSTM model. While often effective, this static fusion approach can be suboptimal, as it lacks a mechanism to dynamically weigh the importance of different modalities based on the context of a specific sample.

To address this, more advanced architectures have incorporated **attention-based fusion** mechanisms. These methods allow the model to learn the relative importance of different features or modalities in a data-driven manner. In the context of regulatory genomics, this is particularly powerful. `Chromoformer`, for example, uses a hierarchical Transformer to integrate 1D histone modification signals with 3D chromatin interaction data, using attention to learn which genomic regions and interactions are most relevant for predicting gene expression [86]. Similarly, `Proformer` introduces a dynamic modality-interaction attention mechanism to selectively weigh chromatin and sequence features, enhancing generalization across cell types [30]. In broader biomedical applications, models like `MultiSurv` have demonstrated how Transformer-based fusion heads can effectively integrate multiple patient data modalities for cancer survival prediction [102]. These attention-based methods represent a significant step beyond static concatenation by allowing for a more flexible and context-aware integration of information.

2.6.2 The Role of Biologically Informed AI

While data-driven fusion strategies are powerful, an emerging and highly promising trend is the development of **biologically informed AI**, where explicit domain knowledge is integrated to guide the model's learning process. This paradigm moves beyond letting the model discover all patterns from scratch and instead injects established biological principles to enhance accuracy, improve interpretability, and ensure that the model's decisions are biologically plausible. A comprehensive review by Conard, DenAdel, and Crawford [103] outlines the spectrum of such interpretable approaches in genomics.

This integration of prior knowledge can take several forms. One approach is to structure the model's architecture to reflect known biological relationships. `GraphReg`, for instance, uses a GAT to explicitly model 3D chromatin interactions, thereby embedding a physical prior about the genome's structure into the model [92]. Another strategy is to use biological knowledge to augment the feature space directly. This is common in promoter prediction, where features like motif counts, conservation scores, or epigenetic marks (e.g., H3K4me3) are used as inputs alongside the primary DNA sequence [85, 104].

Recent studies have emphasized the power of combining deep learning

with such priors. The work by Hartman et al. [105] in proteomics and by McGee et al. [106] in environmental health both demonstrate that models guided by biological knowledge are not only more accurate but also more reliable and interpretable. This principle of using priors to constrain and guide AI models is a powerful concept that is being explored across diverse scientific fields [107, 108].

2.7 Research Gaps and Future Directions

Identified Gaps in Current Literature

Despite substantial progress in computational biology and deep learning approaches for gene regulation, current promoter prediction models exhibit critical limitations across both methodological and biological dimensions. These limitations can be broadly categorized into six interrelated themes: (i) static fusion strategies that ignore regulatory dynamics, (ii) generalizability and overfitting in high-dimensional multimodal data, (iii) poor interpretability of black-box models, (iv) insufficient benchmarking and dataset diversity, (v) lack of biological validation, and (vi) limited modeling of long-term regulatory mechanisms.

First, many existing models rely on rigid, static fusion strategies that integrate genomic, epigenomic, and transcriptomic modalities without accounting for the dynamic, context-specific nature of gene regulation [109, 110]. Such models typically adopt early, intermediate, or late fusion methods but do not allow for adaptive integration that reflects changing chromatin states, tissue-specific transcription factor activity, or environmental stimuli [111]. As a result, they risk oversimplifying complex biological processes and limit their applicability across diverse cellular contexts [112].

Second, generalizability remains a major challenge in deep learning-based promoter prediction, especially in biomedical domains characterized by small sample sizes, missing modalities, and unbalanced feature spaces [99, 111]. Overfitting is exacerbated by high-dimensional inputs and complex architectures, and current models rarely employ adaptive, similarity-aware mechanisms to dynamically weigh the importance of each modality based on biological relevance [99].

Third, the interpretability of deep models continues to hinder their adoption and trust in genomics. While transformer-based architectures like BERT-Promoter [113] and PLPMpro [85] show strong predictive performance, they often lack mechanistic transparency and fail to provide biologically meaningful feature attributions. This obscurity is particularly problematic in promoter prediction, where understanding regulatory grammar is key to biological discovery [111].

Fourth, benchmarking remains inconsistent across promoter prediction

studies. Many models are evaluated on narrow datasets limited to specific species, tissues, or promoter types, impeding reproducibility and generalizability [55, 114]. While datasets like ReFeaFi [115] and curated benchmarks [114] have made strides, the field still lacks standardized, multi-species benchmarks under diverse regulatory conditions.

Fifth, biological validation is often missing from computational studies. Although models like ReFeaFi and synthetic promoter work by LaFleur et al. [116] incorporate in vitro and in vivo experiments, most deep models (e.g., BERT-Promoter, PLPMpro, iPromoter-CLA) do not verify predictions experimentally, leaving questions about the biological utility of their outputs [85, 113, 117].

Lastly, long-term biological dynamics such as transcriptional memory, stochastic gene expression, and chromatin remodeling are rarely modeled explicitly. Many existing models rely on static representations of chromatin accessibility and transcription factor binding, ignoring the dynamic and temporal nature of these interactions [118–120]. Furthermore, interactions involving RNA polymerase dynamics, sigma factors, cryptic promoters, and transient nuclear localization remain underexplored in current computational frameworks [116].

In summary, advancing promoter prediction requires models that integrate dynamic, biologically-informed multimodal fusion strategies; leverage interpretable architectures; utilize diverse, standardized benchmark datasets; and include experimental validation to ensure biological relevance. Bridging these gaps will enable a new generation of robust, generalizable, and biologically meaningful computational tools for understanding gene regulation.

Chapter 3

Research Methodology

3.1 Introduction

This chapter provides a comprehensive and detailed blueprint of the data, methodology, and experimental design employed in this thesis. The primary focus is the construction and validation of the central research artifact: a novel multi-modal deep learning model for human promoter region identification. The chapter is structured to logically follow the research process, from the architectural design of the proposed model to the precise protocols used for its empirical evaluation.

The chapter begins by presenting the **Proposed BWAF-Net Architecture**, the core technical contribution of this work. This section provides a granular breakdown of the framework's components, including the Transformer encoder for sequence processing, the Graph Attention Network (GAT) for learning from gene regulatory networks, and the novel Biologically Weighted Attention Fusion (BWAF) layer that integrates these modalities.

Following the architectural description, the chapter details the **Data Acquisition, Curation, and Exploration**. This section describes the sources and curation procedures for the three distinct data modalities genomic sequences, gene interaction networks, and biological priors and presents key findings from the exploratory data analysis. This is followed by a detailed account of the **Data Preprocessing Pipeline**, which outlines the critical steps taken to align, transform, and structure these heterogeneous data sources into a format suitable for the deep learning model.

The chapter then lays out the complete **Experimental Design** for the model's validation. This section defines the evaluation metrics, the baseline and ablation models used for comparison, and the specific protocols for the interpretability analysis. Finally, the entire research process is contextualized within the **Design Science Research (DSR)** paradigm, which provides the overarching

methodological framework for this study.

3.2 The Proposed BWAF-Net Architecture

The core technical contribution of this thesis is a novel multi-modal deep learning architecture, named **BWAF-Net**, engineered for the complex task of human promoter identification. The framework is built on the principle that superior predictive accuracy and biological interpretability can be achieved by synergistically integrating information from three distinct data modalities. These modalities are the linear DNA sequence, the context-specific gene regulatory network, and explicit biological prior knowledge.

3.2.1 Architectural Overview

As illustrated in Figure 3.1, BWAF-Net employs a multi-branch architecture to process its heterogeneous inputs. A **Sequence Processing** branch, implemented as a Transformer encoder, learns deep feature representations from 2000bp DNA sequences. In parallel, a **Network Processing** branch, using a Graph Attention Network (GAT), learns contextual embeddings for each gene from 36 tissue-specific gene-gene graphs.

The outputs of these two encoders, along with an 11-dimensional vector of biological priors, are then fed into the novel **Biologically Weighted Attention Fusion (BWAF) Layer**. This layer is the central component of the architecture; it uses the priors to dynamically weigh the contributions of the sequence and network features. The resulting integrated representation is then passed to a final prediction head for classification. A key aspect of this methodology for ensuring computational tractability is that the GAT branch output is pre-computed once for all genes in the dataset. During training, the model performs a simple lookup to retrieve these pre-computed features, making the end-to-end process highly efficient.

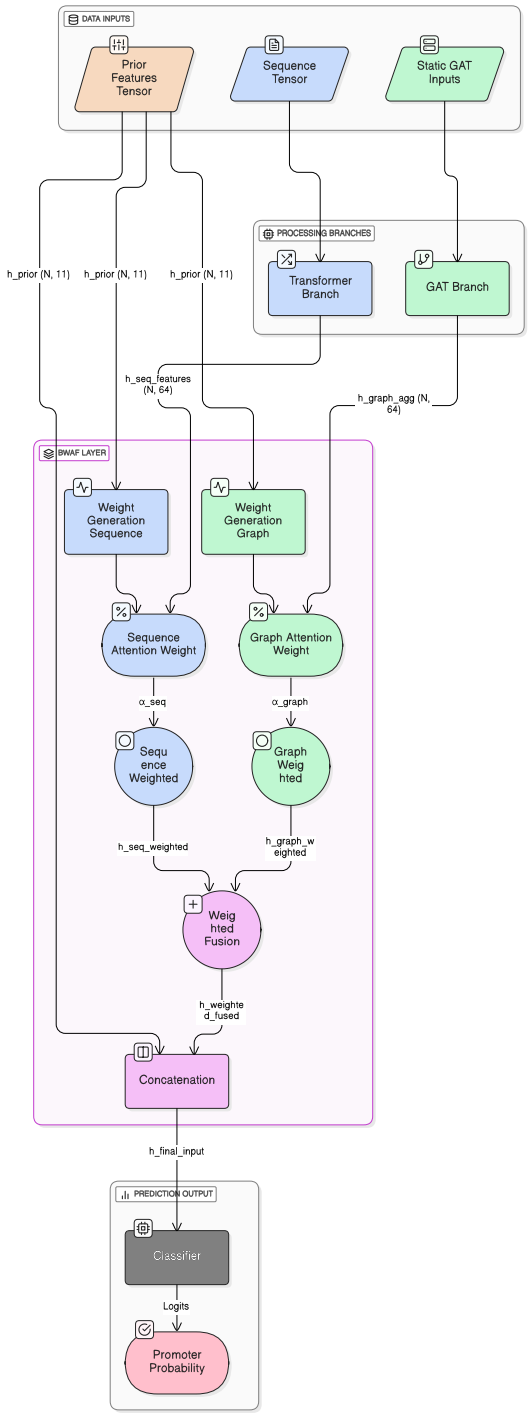

Figure 3.1: Overall Model Architecture Diagram of BWAF-Net. The framework integrates three input modalities. A Transformer processes DNA sequences, while a GAT processes gene networks. The outputs are dynamically fused by the BWAF layer, which is guided by biological priors, to produce a final prediction.

3.2.2 The Transformer Encoder

The Transformer branch is designed to learn rich feature representations from raw DNA sequences by capturing both local motifs and long-range dependencies. This is achieved using a standard Transformer encoder architecture [80]. The overall process involves embedding the sequence, processing it through attention-based encoder layers, and aggregating the output, with the specific implementation flow detailed in Algorithm 1.

The input to this branch is a batch of integer-encoded DNA sequences $S \in \mathbb{Z}^{N \times L}$, where N is the batch size and $L = 2000$. Each nucleotide is first mapped to a continuous **64-dimensional** vector using a learnable embedding layer. To inject sequence order information, a learnable positional encoding tensor $PE \in \mathbb{R}^{1 \times L \times 64}$ is added to the nucleotide embeddings.

The resulting tensor is then processed by a stack of **2 Transformer encoder layers**. Following modern best practices for training stability, a pre-LayerNorm architecture (`norm_first=True`) is used. Each layer contains a multi-head self-attention (MHA) module and a position-wise feed-forward network. The core of the MHA module is the scaled dot-product attention mechanism. For a given set of Queries (Q), Keys (K), and Values (V), the attention output is calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.1}$$

where d_k is the dimension of the keys. This operation is performed in parallel across **4 attention heads**, and their outputs are concatenated and linearly projected. The feed-forward network, which is applied after the MHA module, has an inner dimension of **256**.

After the final encoder layer, the sequence of context-aware output embeddings $H_{\text{attn}} \in \mathbb{R}^{N \times L \times 64}$ is aggregated into a single vector representation for each sequence via **masked mean pooling**. Finally, this aggregated vector is passed through a dedicated preparation layer to produce the final sequence feature vector, $h_{\text{seq\_features}} \in \mathbb{R}^{N \times 64}$, ready for the fusion stage.

Algorithm 1 Transformer Encoder Forward Pass

Input: Integer-encoded sequences $S \in \mathbb{Z}^{N \times L}$
Parameters: Embedding Layer θ_{emb}, Positional Encoding PE, Transformer Encoders θ_{enc}, Fusion Prep Layer θ_{prep}
Output: Sequence features $h_{\text{seq\_features}} \in \mathbb{R}^{N \times 64}$
$X_{\text{emb}} \leftarrow \text{Embedding}(S; \theta_{\text{emb}})$ ▷ Shape: $N \times L \times 64$
$X_{\text{pos}} \leftarrow X_{\text{emb}} + PE$ ▷ Add positional information
$M_{\text{pad}} \leftarrow (S == \text{PAD\_IDX})$ ▷ Create padding mask, Shape: $N \times L$
$H_{\text{attn}} \leftarrow \text{TransformerEncoder}(X_{\text{pos}}, \text{src\_key\_padding\_mask} = M_{\text{pad}}; \theta_{\text{enc}})$ ▷ Shape: $N \times L \times 64$
▷ Masked Mean Pooling for aggregation
$M_{\text{agg}} \leftarrow (\neg M_{\text{pad}}).\text{unsqueeze(-1).float()}$ ▷ Shape: $N \times L \times 1$
$H_{\text{sum}} \leftarrow \text{sum}(H_{\text{attn}} * M_{\text{agg}}, \text{dim} = 1)$
$N_{\text{nonpad}} \leftarrow \text{sum}(M_{\text{agg}}, \text{dim} = 1).clamp(min = 1)$ ▷ Avoid division by zero
$H_{\text{agg}} \leftarrow H_{\text{sum}} / N_{\text{nonpad}}$ ▷ Shape: $N \times 64$
$h_{\text{seq\_features}} \leftarrow \text{FusionPrepLayer}(H_{\text{agg}}; \theta_{\text{prep}})$
return $h_{\text{seq\_features}}$

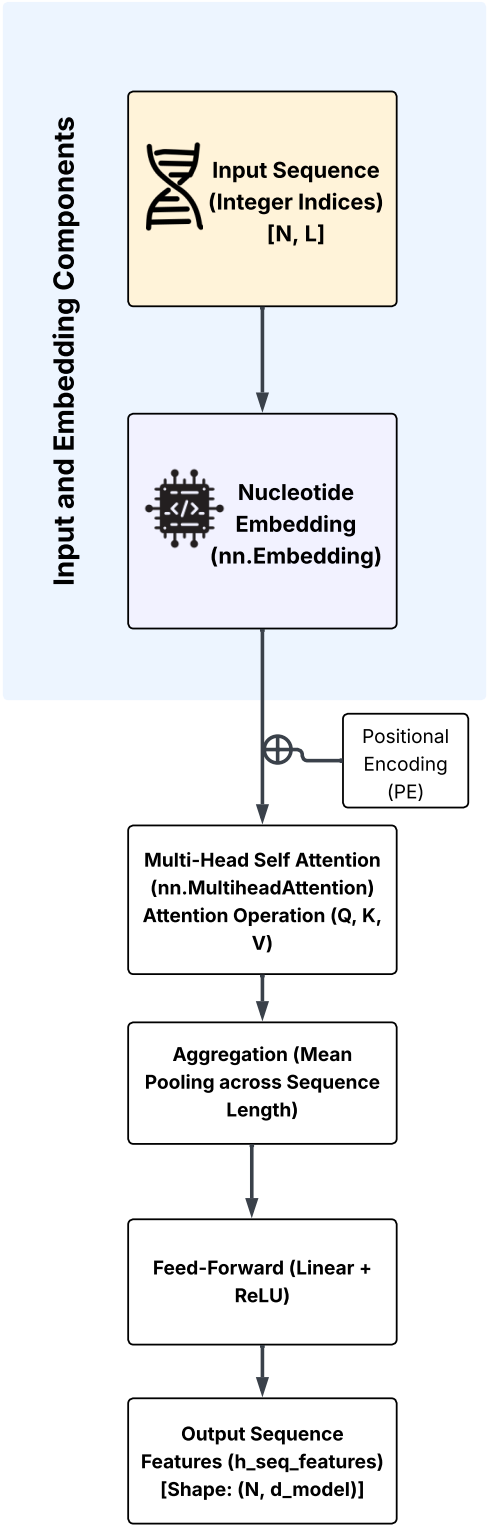

Figure 3.2: Detailed Flowchart of the Transformer Branch, illustrating the sequential processing from integer-encoded sequences to the final sequence feature vector.

3.2.3 The GAT Encoder

The GAT Encoder is designed to learn a representation for each gene based on its connectivity and context within 36 different tissue-specific regulatory networks. A key methodological feature for ensuring computational efficiency is that the output of this branch is **pre-computed once** for all 17,787 genes in the graph. During model training, the framework performs a simple lookup to retrieve these pre-computed feature vectors, avoiding repeated, costly graph convolutions. The entire pre-computation process is detailed in Algorithm 2.

Inputs

This branch takes two static inputs. The first is an initial node feature matrix $X_{\text{node}} \in \mathbb{R}^{N_{\text{genes}} \times N_{\text{TFs}}}$, where $N_{\text{genes}} = 17,787$ and $N_{\text{TFs}} = 644$. Each row vector in this matrix represents a gene's average normalized interaction profile across all 36 tissues. The second input is a set of 36 sparse gene-gene graph structures, $\{E_t\}_{t=1}^{36}$, where each E_t is an edge index tensor representing the co-regulatory links in a specific tissue.

Mathematical Formulation

For each of the 36 tissue-specific graphs, a Graph Attention Network (GAT) layer [88] updates the feature vector h_i for each gene node i. The update rule is based on an attention-weighted aggregation of its neighbors $\mathcal{N}(i)$. For a single attention head, the attention coefficient α_{ij} between node i and its neighbor j is computed as follows:

$$e_{ij} = \text{LeakyReLU}\left(\mathbf{a}^T[Wh_i || Wh_j]\right) \tag{3.2}$$

$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(e_{ik})} \tag{3.3}$$

where W is a learnable linear transformation, $\mathbf{a}$ is a learnable weight vector, and $||$ denotes concatenation.

In a multi-head attention mechanism, this process is repeated in parallel across K independent heads. In our implementation, we use a single GAT layer where the number of heads is set to **1** ($K = 1$) for the final output. The updated node feature vector is thus computed as:

$$h'_i = \text{ELU}\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij} W h_j\right) \tag{3.4}$$

where ELU (Exponential Linear Unit) is the non-linear activation function. This

operation produces a final gene embedding matrix $H'^{(t)} \in \mathbb{R}^{N_{\text{genes}} \times 64}$ for each tissue t.

Aggregation Across Tissues

After processing all 36 tissue graphs independently, the resulting 36 embedding matrices are stacked and aggregated via element-wise averaging. This produces a single, robust network representation for every gene, $H_{\text{graph\_all}} \in \mathbb{R}^{N_{\text{genes}} \times 64}$, which is the final pre-computed output of this branch.

Algorithm 2 GAT Encoder Forward Pass (Pre-computation)

Input: Initial node features $X_{\text{node}} \in \mathbb{R}^{N_{\text{genes}} \times N_{\text{TFs}}}$, List of 36 edge indices $\{E_t\}_{t=1}^{36}$
Parameters: GAT Layer θ_{gat}
Output: Pre-computed graph features for all genes $H_{\text{graph\_all}} \in \mathbb{R}^{N_{\text{genes}} \times 64}$
$H_{\text{tissues}} \leftarrow []$ ▷ Initialize list to store tissue-specific embeddings
for each edge index E_t in $\{E_t\}_{t=1}^{36}$ **do**
    $H_t \leftarrow \text{GATLayer}(X_{\text{node}}, E_t; \theta_{\text{gat}})$ ▷ Process graph for tissue t; Output shape: $N_{\text{genes}} \times 64$
    Append H_t to H_{tissues}
end for
$H_{\text{stacked}} \leftarrow \text{stack}(H_{\text{tissues}}, \text{dim} = 0)$ ▷ Shape: $36 \times N_{\text{genes}} \times 64$
$H_{\text{graph\_all}} \leftarrow \text{mean}(H_{\text{stacked}}, \text{dim} = 0)$ ▷ Average across tissues
return $H_{\text{graph\_all}}$

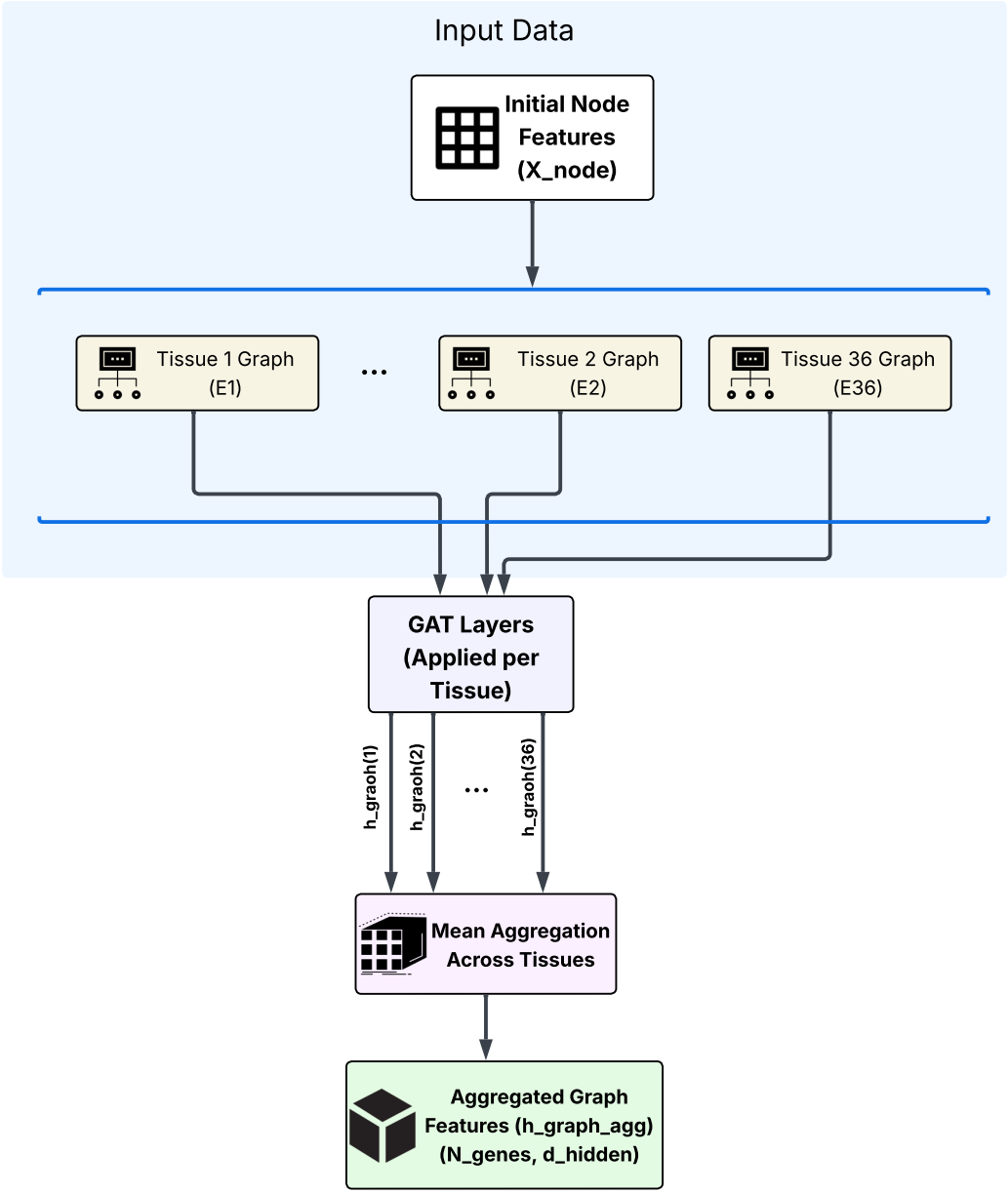

Figure 3.3: GAT Branch Processing Diagram. Initial node features and 36 tissue-specific graphs are processed by GAT layers to produce tissue-specific embeddings, which are then aggregated to form the final graph feature representation.

3.2.4 The BWAF Fusion Layer

The Biologically Weighted Attention Fusion (BWAF) layer is the central methodological contribution of this thesis. It is designed to address the challenge of multi-modal fusion by moving beyond simple concatenation. As demonstrated by the ablation studies in Chapter 4, naive fusion can be a suboptimal strategy for this problem. The intuition behind BWAF is to use the biological prior features not merely as another input to a classifier, but as an active **controller of infor-**

mation flow. This layer learns an adaptive policy to weigh the sequence-based analysis against the network-based analysis for each specific sample, guided by its known biological characteristics. The entire forward pass of this layer is detailed in Algorithm 3.

Inputs and Projections

The BWAF layer receives three inputs for a batch of size N: the sequence features $h_{\text{seq\_features}} \in \mathbb{R}^{N\times 64}$, the graph features $h_{\text{graph\_features}} \in \mathbb{R}^{N\times 64}$, and the log-transformed biological priors $h_{\text{prior}} \in \mathbb{R}^{N\times 11}$. As both the Transformer and GAT encoders were designed to output features of the same dimension, no projection layers are needed in this specific implementation; the common feature dimension d_{common} is 64.

Mathematical Formulation and Implementation

The fusion process unfolds in four main steps. First, two small, independent neural networks, termed `weight_generators`, learn to map the 11-dimensional prior vector to a scalar attention weight for each modality. Each generator is a two-layer MLP that outputs a single logit, which is then normalized to the range [0, 1] by a Sigmoid function (σ):

$$\alpha_{\text{seq}} = \sigma(\text{MLP}_{\text{seq\_wt}}(h_{\text{prior}})) \in [0,1]^{N\times 1} \tag{3.5}$$

$$\alpha_{\text{graph}} = \sigma(\text{MLP}_{\text{graph\_wt}}(h_{\text{prior}})) \in [0,1]^{N\times 1} \tag{3.6}$$

Second, these learned attention weights are broadcast and multiplied element-wise ($\odot$) with the feature vectors from their respective branches. This adaptively scales each modality's contribution based on its prior-informed importance:

$$h_{\text{seq\_weighted}} = \alpha_{\text{seq}} \odot h_{\text{seq\_features}} \tag{3.7}$$

$$h_{\text{graph\_weighted}} = \alpha_{\text{graph}} \odot h_{\text{graph\_features}} \tag{3.8}$$

Third, the modulated feature vectors are combined through summation to create a dynamically weighted fused representation:

$$h_{\text{weighted\_fused}} = h_{\text{seq\_weighted}} + h_{\text{graph\_weighted}} \tag{3.9}$$

Finally, to ensure the classifier has direct access to the original prior information, this adaptively fused representation is concatenated with the original prior vector. The resulting combined vector is passed through a LayerNorm for stabilization and then processed by a final MLP classifier to generate the output logits. This

classifier is an MLP with a hidden dimension of **128**. The complete transformation is:

$$h_{\text{final\_input}} = \text{concat}(h_{\text{weighted\_fused}}, h_{\text{prior}}) \in \mathbb{R}^{N\times(64+11)} \tag{3.10}$$

$$\text{logits} = \text{Classifier}(\text{LayerNorm}(h_{\text{final\_input}})) \in \mathbb{R}^{N\times 1} \tag{3.11}$$

Algorithm 3 BWAF Layer Forward Pass

Input: Sequence features h_{seq}, Graph features h_{graph}, Prior features h_{prior}
Parameters: Weight generators $\theta_{\text{wg\_seq}}, \theta_{\text{wg\_graph}}$, Classifier MLP θ_{clf}
Output: Logits $\hat{z} \in \mathbb{R}^{N\times 1}$
▷ Generate attention weights from priors
$\alpha_{\text{seq}} \leftarrow \sigma(\text{MLP}(h_{\text{prior}}; \theta_{\text{wg\_seq}}))$
$\alpha_{\text{graph}} \leftarrow \sigma(\text{MLP}(h_{\text{prior}}; \theta_{\text{wg\_graph}}))$
▷ Modulate and combine deep features
$h_{\text{seq\_w}} \leftarrow \alpha_{\text{seq}} \odot h_{\text{seq}}$
$h_{\text{graph\_w}} \leftarrow \alpha_{\text{graph}} \odot h_{\text{graph}}$
$h_{\text{fused}} \leftarrow h_{\text{seq\_w}} + h_{\text{graph\_w}}$
▷ Final integration and classification
$h_{\text{final}} \leftarrow \text{concat}(h_{\text{fused}}, h_{\text{prior}})$
$h_{\text{norm}} \leftarrow \text{LayerNorm}(h_{\text{final}})$
$\hat{z} \leftarrow \text{ClassifierMLP}(h_{\text{norm}}; \theta_{\text{clf}})$
return $\hat{z}$

The design of the BWAF layer represents a distinct paradigm for multimodal fusion, deliberately chosen over more conventional methods. It stands in contrast to two common approaches. The first, **static concatenation**, was shown to be suboptimal for this problem in our ablation studies (Chapter 4). This is because the high-dimensional, complex features learned by the deep encoders can overwhelm and dilute the clean, highly-discriminative signal from the low-dimensional biological priors. The second, **data-driven attention** (e.g., cross-attention), learns to weigh modalities based on patterns within the learned features themselves. While powerful, this approach lacks explicit biological guidance and risks learning spurious correlations from the dataset.

By contrast, the BWAF layer's use of priors as an explicit controller of information flow yields three key advantages. First, it injects a strong and beneficial **inductive bias** into the architecture: the scientific hypothesis that the optimal fusion strategy depends on the known biological characteristics of a promoter. Second, it creates a direct, analyzable link between domain knowledge and the model's internal behavior, making the fusion process inherently **transparent and interpretable**. Finally, by guiding the model with established rules, it makes the learning process more **efficient**. This design positions BWAF within

the principles of prior-informed machine learning, an emerging field aimed at creating more robust and trustworthy AI systems in complex scientific domains [105, 106].

3.2.5 Loss Function

For the binary classification task of identifying promoter versus non-promoter regions, the model was trained end-to-end by optimizing a carefully chosen loss function. The standard and most appropriate choice for this task is the **Binary Cross-Entropy with Logits Loss** (`BCEWithLogitsLoss`). This function is the recommended implementation in PyTorch for binary classification due to its superior numerical stability compared to using a separate Sigmoid activation layer followed by a standard 'BCELoss'. By combining the two operations, it avoids potential floating-point precision issues that can arise when dealing with values very close to 0 or 1[121, 122].

From Model Output to Probability

The final layer of the BWAF-Net's classifier outputs a single, raw, unbounded scalar value for each sample in a batch. This value is known as a **logit** ($\hat{z}_i$). The logit represents the model's confidence in the positive class (promoter) on a logarithmic scale. To convert this logit into a probability p_i that is bounded between 0 and 1, the 'BCEWithLogitsLoss' function internally applies the **Sigmoid function** (σ):

$$p_i = \sigma(\hat{z}_i) = \frac{1}{1 + e^{-\hat{z}_i}} \tag{3.12}$$

Mathematical Formulation

The Binary Cross-Entropy loss then measures the dissimilarity between the predicted probability p_i and the true binary label $y_i \in \{0, 1\}$. The loss for a single sample is computed as:

$$\mathcal{L}_{\text{BCE}}(y_i, \hat{z}_i) = -[y_i \cdot \log(\sigma(\hat{z}_i)) + (1 - y_i) \cdot \log(1 - \sigma(\hat{z}_i))] \tag{3.13}$$

This formulation has an intuitive interpretation:

- When the true label $y_i = 1$ (a promoter), the second term of the equation becomes zero. The loss simplifies to $-\log(p_i)$. The model is penalized if its predicted probability p_i for the positive class is low (i.e., far from 1).
- When the true label $y_i = 0$ (a non-promoter), the first term of the equation becomes zero. The loss simplifies to $-\log(1 - p_i)$. The model is penalized

if its predicted probability for the negative class, $1 - p_i$, is low (i.e., if p_i is high, far from 0).

The total loss for a batch is the mean of the losses for all samples within that batch. During training, the goal of the optimizer is to adjust the model's parameters (weights and biases) via backpropagation to minimize this loss function, thereby driving the model's predictions to be as close as possible to the true labels.

3.3 Data Acquisition, Curation, and Exploration

The foundation of this study is a multi-modal dataset curated to provide complementary views of gene regulation, enabling the model to learn a holistic representation of promoter activity. The data was acquired from three distinct, publicly available sources and structured into three core modalities. This section details the acquisition of these raw data sources, the exploratory analysis performed to understand their characteristics, and the curation process used to derive the final, model-ready datasets. The overall integration pipeline is illustrated in Figure 3.4.

3.3.1 Overview of Data Modalities

The three modalities were chosen to capture different scales of biological regulation. The first is **genomic sequence data**, sourced from the Ensembl GRCh38 human reference genome. This modality provides the fundamental, one-dimensional regulatory grammar encoded directly in the DNA. The second is **gene regulatory networks**, obtained from the GRAND database, which provide a higher-level, context-specific view of the regulatory landscape by modeling gene-TF interactions across 36 different human tissues. The third modality, **biological priors**, provides explicit, domain-knowledge features derived by quantifying the occurrences of known regulatory motifs within the promoter sequences themselves. Together, these three data streams provide a rich, multi-faceted representation of each potential promoter region.

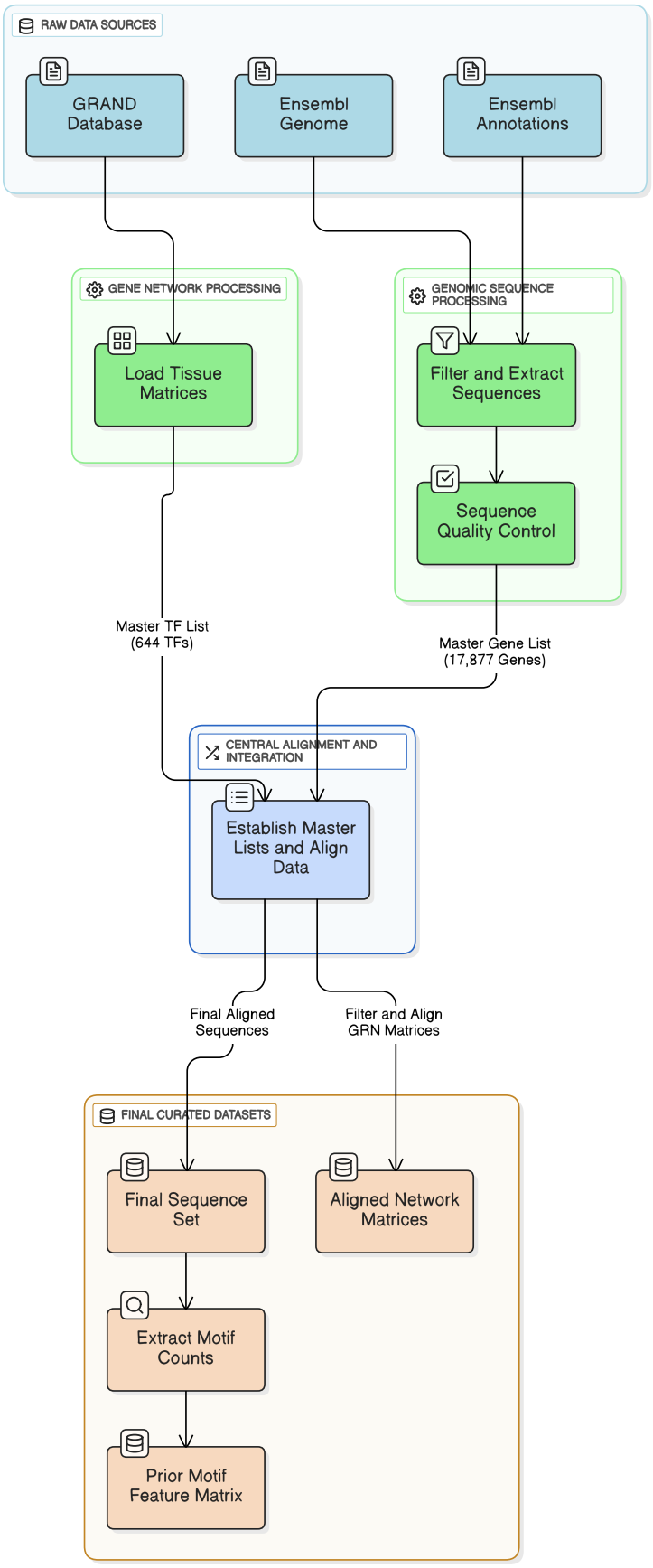

Figure 3.4: Overview of the data acquisition and curation pipeline, illustrating how the three raw data sources are processed and aligned to create the final multi-modal dataset for the BWAE model.

3.3.2 Genomic Sequence Data

The foundation of the sequence-based modality is the human genome reference assembly and its corresponding gene annotations. This section details the acquisition of these raw datasets, the exploratory analysis conducted to characterize them, and the systematic curation and extraction pipeline used to generate the final set of promoter and non-promoter sequences for the model.

Data Source and Acquisition

The primary genomic data was sourced from the human genome reference assembly **GRCh38 (hg38)** and its comprehensive gene annotations from **Ensembl release 110** [123]. The specific files were downloaded directly from the Ensembl FTP server:

- **Genome Sequence:** The primary assembly DNA sequence file, `Homo_sapiens.GRCh38.dna.primary_assembly.fa.gz`.
- **Gene Annotations:** The gene annotation file in GTF format, `Homo_sapiens.GRCh38.110.gtf.gz`.

Exploratory Analysis and Curation Rationale

Upon acquisition, an exploratory data analysis was performed to understand the composition and quality of the raw data. The genome assembly was found to contain 194 distinct contigs, comprising the 24 main human chromosomes (1-22, X, Y) and 170 additional unplaced scaffolds and alternate loci. The GTF annotation file described a total of 62,754 genes, with a diverse range of biological types, as summarized in Table 3.1.

Table 3.1: Distribution of Top Gene Biotypes in the Raw Ensembl GTF Annotation.

Gene Biotype	Count
protein_coding	**20,070**
lncRNA	18,866
processed_pseudogene	10,148
unprocessed_pseudogene	2,609
misc_RNA	2,221
snRNA	1,910
miRNA	1,879
Other	5,042
Total	**62,745**

To create a focused and biologically relevant dataset suitable for training a robust promoter prediction model, a rigorous curation strategy was adopted based on these findings. The dataset was filtered to include only **protein-coding genes** located on the **main human chromosomes**. This decision was driven by two key justifications:

1. **Focus on Protein-Coding Genes:** As shown in Table 3.1, protein-coding genes form the largest and most well-characterized class of functional genes. Their promoter regions and Transcription Start Sites (TSSs) are extensively studied and annotated, providing a reliable ground truth for model training. Non-coding genes and pseudogenes were excluded, as their promoter structures can be highly variable, less characterized, or non-functional, which could introduce significant noise and ambiguity.

2. **Exclusion of Non-Canonical Chromosomes:** Genes located on unplaced scaffolds and alternate loci were excluded. These regions are often fragmented or represent alternative haplotypes, and the gene annotations within them can be less reliable. Focusing on the main chromosomes (1-22, X, Y) ensures that the model is trained on the canonical human genome.

This stringent filtering strategy resulted in a high-quality candidate set of 20,070 protein-coding genes.

Promoter Definition and Sequence Extraction

A critical step in the methodology is the operational definition of a promoter region. For this study, a promoter is defined as the **2000 base pairs (bp) sequence immediately upstream of a gene's annotated Transcription Start Site (TSS)**. This specific 2000bp window was not chosen arbitrarily but is a deliberate decision grounded in both biological principles and established computational best practices.

The primary biological justification is that this window is sufficiently large to encompass the most critical regulatory regions. It captures the **core promoter** (typically within 50bp of the TSS), which contains essential motifs like the TATA-box and Initiator elements that are fundamental for transcription initiation [124]. It also includes the **proximal promoter** (extending up to 500bp upstream), which houses key transcription factor binding sites. Furthermore, while enhancers can be located many kilobases away, this 2000bp region is effective at capturing the influence of more local, **enhancer-like distal elements** that play a significant role in modulating gene expression [125].

From a computational standpoint, the 2000bp length aligns with the state-of-the-art in deep learning for genomics. It provides a balanced **signal-to-noise ratio**, capturing the dense regulatory zone near the TSS while minimizing the inclusion of irrelevant information from further upstream regions, which could reduce model specificity [126]. Moreover, using a fixed-length input simplifies data preprocessing and ensures the uniformity required for transformer-based models. Recent high-performance models like BERT-Promoter and msBERT-Promoter have demonstrated the effectiveness of using input sequences of a similar scale to capture the necessary long-range regulatory context [82, 113].

Following this well-supported definition, a data generation pipeline was implemented using custom Python scripts and the Biopython library [127]:

- **Positive Set (Promoters):** For each of the 20,070 candidate protein-coding genes, the 2000bp sequence upstream of its TSS was extracted. The process correctly handled gene strand orientation to ensure the true upstream region was selected.
- **Negative Set (Non-Promoters):** A corresponding negative dataset of equal size was generated. These 2000bp sequences were randomly sampled from genomic regions on the main chromosomes that were confirmed to not overlap with any defined promoter regions or annotated gene bodies.
- **Quality Control:** All extracted sequences, both positive and negative, underwent a final quality control filter. Any sequence that did not have an exact length of 2000bp or contained any ambiguous nucleotides (represented as 'N') was discarded.

This rigorous acquisition, curation, and extraction process yielded the final, clean sequence dataset used for all subsequent model training, consisting of **20,028 promoter sequences** and **20,028 non-promoter sequences**.

3.3.3 Gene Interaction Network Data

To capture the context-specific regulatory landscape for each gene, this study incorporates a modality based on Gene Regulatory Networks (GRNs). This section details the selection of the data source, the acquisition of the networks, their primary characteristics discovered through exploratory analysis, and the critical gene set alignment step required for multi-modal integration.

Data Source and Rationale

The context-specific GRN data was sourced from the **GRAND database**, a large, publicly available repository of computationally inferred networks [128].

The GRAND database was selected as the optimal source for this research for several key reasons:

- **Availability of Normal Tissue Networks:** Unlike many GRN databases that focus on disease states, GRAND provides regulatory networks for 36 normal human tissues derived from the Genotype-Tissue Expression (GTEx) project. This ensures the model learns from baseline regulatory interactions in non-pathological conditions.
- **Context-Specific Data:** The tissue-specific nature of the networks is more biologically relevant than a single, generic network, allowing the model to leverage physiologically meaningful interactions.
- **High-Quality Inference:** GRAND networks are inferred using established and robust network reconstruction algorithms such as PANDA [129] and LIONESS [130], which integrate diverse data sources to produce more accurate networks.
- **Computational Efficiency:** The availability of pre-computed GRNs allowed this research to focus on the development of the novel fusion architecture, avoiding the computationally prohibitive step of inferring 36 genome-scale networks from scratch.

Data Acquisition and Characteristics

From the GRAND database, the **36 GRNs corresponding to normal human tissues** were downloaded. A full list of these tissues is provided in Table 3.2. Exploratory data analysis on the raw network files revealed a consistent and dense structure. Each of the 36 tissue-specific networks was represented as a dense matrix of inferred interaction strengths between **644 transcription factors (TFs)** and **30,243 genes**.

The data was found to be fully dense (0% sparsity), with a numerical score present for every TF-gene pair. Analysis of the interaction scores across all 36 tissues, as shown in Figure 3.5, revealed a distribution centered around a slightly negative mean (-0.0102). This suggests a prevalence of inferred repressive interactions in the dataset, a common feature in genome-scale regulatory models.

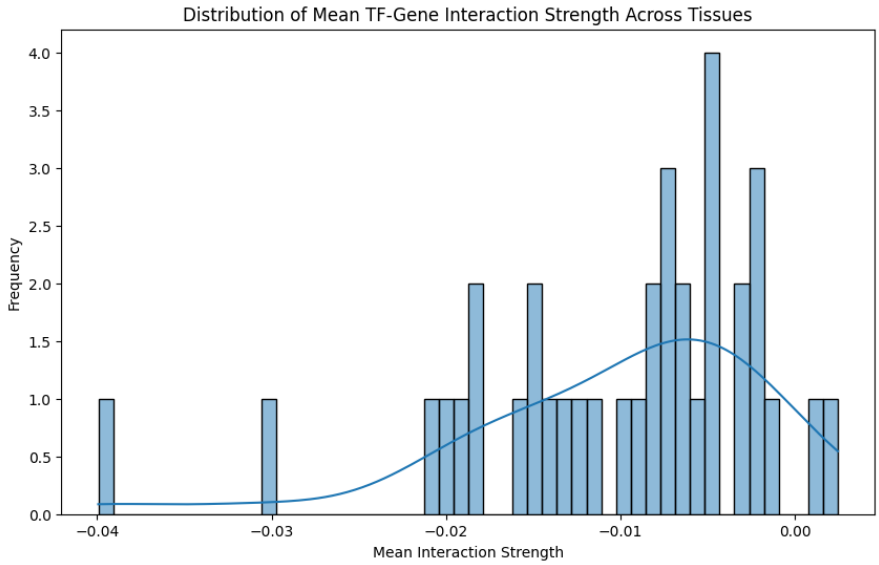

Figure 3.5: Distribution of the mean TF-gene interaction strength across all 36 normal tissue networks from the GRAND database. The distribution's slight negative skew suggests a prevalence of inferred repressive interactions in the raw data.

Gene Set Alignment

A critical curation step for this multi-modal study was to ensure that all data streams referred to the same set of biological entities. The set of 20,028 protein-coding genes from the curated sequence dataset was therefore intersected with the set of 30,243 genes present in the GRAND networks. This alignment process revealed:

- **17,787 genes** were present in both the promoter sequence dataset and the GRAND networks.
- 2,241 genes were exclusive to the promoter sequence dataset.
- 12,456 genes were exclusive to the GRAND network dataset.

To ensure data integrity, the final dataset for the model was defined by this intersection. All subsequent analyses and model training were performed exclusively on this consistently aligned set of **17,787 protein-coding genes**, for which all three data modalities were available. This step is a prerequisite for creating the master gene lists used in the data preprocessing pipeline.

Table 3.2: Normal Human Tissues Included from the GRAND Database (N=36).

Tissue Category	Tissue Name	Tissue Category	Tissue Name
Adipose	Adipose Subcutaneous	Gastrointestinal	Stomach
Adipose	Adipose Visceral	Heart	Heart Atrial Appendage
Adrenal Gland	Adrenal Gland	Heart	Heart Left Ventricle
Artery	Artery Aorta	Intestine	Intestine Terminal Ileum
Artery	Artery Coronary	Kidney	Kidney Cortex
Artery	Artery Tibial	Liver	Liver
Brain	Brain Basal Ganglia	Lung	Lung
Brain	Brain Cerebellum	Ovary	Ovary
Brain	Brain Other	Pancreas	Pancreas
Breast	Breast	Pituitary	Pituitary
Blood	Whole Blood	Prostate	Prostate
Colon	Colon Sigmoid	Salivary Gland	Minor Salivary Gland
Colon	Colon Transverse	Skeletal Muscle	Skeletal Muscle
Esophagus	Esophagus Mucosa	Skin	Skin
Esophagus	Esophagus Muscularis	Spleen	Spleen
Esophagus	Gastroesophageal Junction	Testis	Testis
Nerve	Tibial Nerve	Thyroid	Thyroid
Uterus	Uterus	Vagina	Vagina

3.3.4 Biological Prior Data

To complement the data-driven features learned by the deep learning branches, a third modality consisting of explicit, domain-knowledge-based features was curated. The rationale for this inclusion is to provide the model with an interpretable, biologically-grounded signal that can guide and enhance its learning process. These features, referred to as "biological priors," were derived by quantifying the presence of key regulatory motifs within the final set of 20,028 clean promoter sequences.

Selection of Regulatory Motifs

A set of 11 key regulatory motifs was selected based on a survey of well-characterized transcriptional regulatory elements that are fundamental to eukaryotic gene expression [131, 132]. The selection covers core promoter elements, proximal binding sites, and other functionally important sequences known to bind transcription factors and influence gene activity [133]. The specific sequence patterns used for identification were chosen to reflect the established consensus sequences reported in the literature. Table 3.3 details the selected motifs, the patterns used for their quantification, their primary biological significance, and supporting citations.

Table 3.3: Target Regulatory Motifs for Biological Prior Feature Extraction and Supporting Literature.

Motif Name	Representative Sequence(s) / Feature	Typical Function / Significance	Citation(s)
TATA Box	`TATAAA`, `TATATA`, `TATAAG`	Core promoter element (∼-30 bp), binds TBP for transcription initiation.	[131, 134]
CAAT Box	`CCAAT`	Proximal element (∼-80 bp), binds factors like NF-Y, enhances efficiency.	[132, 135]
GC Box	`GGGCGG`, `GGCGG`, `GGCAG`	Common in various promoters, binds Sp1 family transcription factors.	[132]
CpG Island	High density of `CG` dinucleotides (Count)	Found near the TSS, associated with epigenetic regulation and transcription.	[136]
BRE (TFIIB Rec. Elem.)	`GCG`, `CGG`	Flanks the TATA box, recognized by the general transcription factor TFIIB.	[131]
MRE (Metal Response Elem.)	`TGCACA`, `TGCGCC`, `TGCGCG`	Core consensus that binds MTF-1 to regulate genes in response to heavy metals.	[132]
PPE (Polypyrimidine Elem.)	`TTTTT`, `CCCCC`	T/C-rich tracts near TSS, often associated with the Initiator (Inr) element.	[131]
Octamer Motif (Oct1)	`ATTTGCAT`	Binds POU-domain transcription factors like Oct-1.	[132]
Sp1 Binding Site	`GGGCGG`	Binding site for the ubiquitous Sp1 transcription factor.	[133]
E-box (Enhancer Box)	`CANNTG` (Regex: ‘CA[ACGT]{2}TG‘)	Binds basic helix-loop-helix (bHLH) TFs, crucial for development.	[137, 138]
RFX Binding Site	`GAGACG` (example)	Binds RFX family TFs, involved in immune and developmental genes.	[135]

Quantification Process

A systematic quantification pipeline was implemented using custom Python scripts leveraging regular expressions. For each of the 20,028 promoter sequences, the scripts performed a scan to count the number of non-overlapping occurrences for each of the 11 refined motif patterns defined in Table 3.3. This process transformed each 2000bp DNA sequence into an 11-dimensional integer vector, where each element represents the raw count of a specific regulatory motif. This vector constitutes the "biological prior profile" for each promoter. This raw count data was then saved to a CSV file (`biological_prior_with_thresholds.csv`), serving as the input for the subsequent data preprocessing step, where a log transformation is applied.

3.4 Data Preprocessing Pipeline

Following the acquisition and curation of the three raw data modalities, a multi-stage preprocessing pipeline was executed to transform, align, and structure the heterogeneous data into a format suitable for the deep learning framework. This pipeline serves as the critical bridge between the curated biological datasets and the specific input requirements of the BWAF-Net architecture. Its objectives are to ensure data consistency across modalities, achieve numerical stability for model training, and convert tabular data into meaningful graph structures for the GAT encoder.

The overall workflow, illustrated in Figure 3.6, involves several key transformations. For sequence data, this includes numerical encoding. For biological priors, a variance-stabilizing log transformation is applied. For the network data, the pipeline involves both value normalization and a complex structural conversion from dense TF-gene interaction matrices to sparse, co-regulatory gene-gene graphs. The following subsections detail each of these critical steps.

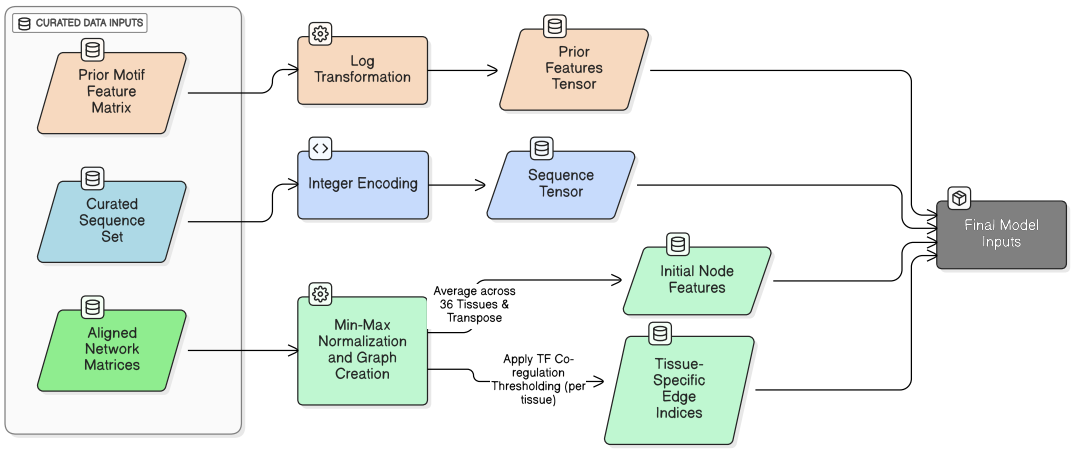

Figure 3.6: The multi-stage data preprocessing pipeline. This workflow illustrates the parallel processing of Sequence, Network, and Prior data streams, highlighting alignment to master lists, numerical transformations (encoding, log-transform, normalization), and the final graph creation step, resulting in model-ready tensors.

3.4.1 Gene and TF Alignment

A foundational step for any multi-modal analysis is ensuring that all data streams refer to the same set of biological entities in a consistent and ordered manner. This data alignment is critical for guaranteeing the integrity of the final dataset and is a prerequisite for the subsequent transformation steps. To achieve this, two master reference lists were established to serve as the single source of truth for ordering all data tensors.

First, a definitive **Master Gene List** was created. This list contains the **17,787 unique protein-coding gene IDs** (in Ensembl ENSG format) that were found to be common to both the curated promoter sequence dataset and the GRAND gene regulatory networks, as determined during the data curation phase (Section 3.3.3). This intersection ensures that every entity in the final dataset has a complete set of features across all modalities.

Concurrently, a **Master TF List** of **644 unique transcription factor (TF) IDs** was derived from the row indices of the raw GRAND database network matrices. A consistent ordering was established and enforced across all 36 tissue-specific networks.

These two master lists were then used to programmatically filter and re-index all relevant data structures. The columns of the 36 gene regulatory network matrices and the rows of the biological prior feature matrix were reordered to precisely match the sequence of the Master Gene List. This rigorous alignment guarantees that for any given sample ‘i‘ in the final dataset, the sequence data,

the prior feature vector, and the corresponding row in the pre-computed GAT output tensor all refer to the exact same biological gene. This prevents data mismatch errors during model training and is essential for the integrity of multi-omics integration strategies [139].

3.4.2 Sequence and Prior Data Transformation

After establishing the master lists, the first steps in the parallel preprocessing pipeline were to convert the raw sequence and biological prior data into numerical tensors suitable for input into the neural network.

The 2000bp DNA sequences, initially in string format, were transformed via **integer encoding**. This process maps each nucleotide to a unique integer index: A=0, T=1, C=2, and G=3. A dedicated index (4) was reserved for the padding token (`<PAD>`), which is used to ensure all sequences in a batch have a uniform length and for the masking mechanism within the Transformer's self-attention layers. This deterministic mapping results in an integer tensor of shape (N, L) for a batch of N sequences of length $L = 2000$, which is the required input format for the model's embedding layer.

Concurrently, the biological prior features required a transformation to address their statistical properties. The raw motif counts (Φ_i), as curated in the data acquisition phase, exhibited highly skewed distributions with a large dynamic range. Such distributions can destabilize model training and cause features with large counts to disproportionately influence the model's parameters. To mitigate this, a common variance-stabilizing **log transformation** was applied to the 11-dimensional feature vector for each gene i:

$$g(\Phi_i)_k = \log(1 + \Phi_{i,k}) \tag{3.14}$$

where $g(\Phi_i)_k$ is the transformed value for the k-th motif count. This transformation compresses the range of the counts while preserving their relative order, resulting in a more normally distributed and well-behaved feature set for the downstream neural network layers.

3.4.3 Network Data Normalization and Graph Creation

The gene interaction network data required the most extensive preprocessing of all three modalities. This pipeline was designed to achieve two distinct goals: first, to create a numerically stable set of initial node features for the GAT, and second, to convert the dense, computationally inferred interaction matrices into sparse, biologically meaningful graph structures suitable for graph neural network

processing.

The first step was to normalize the raw interaction scores. The scores in the 36 raw GRAND matrices have arbitrary units and vary in range from tissue to tissue. To create a consistent feature space, the scores within each of the 36 filtered and aligned tissue matrices (M) were scaled to the range [0, 1] using **Min-Max Scaling**:

$$m'_{ij} = \frac{m_{ij} - \min(M)}{\max(M) - \min(M)} \tag{3.15}$$

This normalization ensures that the GAT branch receives inputs on a common scale across all tissues, which is crucial for stable learning. These 36 normalized matrices were then used to compute the initial node features. For each of the 17,787 genes, its initial feature vector (X_{node}) was defined as the **average of its normalized interaction profiles** across all 36 tissues, resulting in a single tensor of shape $(17,787,644)$.

The second, more complex step was the **construction of sparse gene-gene graphs**. The raw GRAND matrices are dense, implying a potential interaction between every TF and every gene. However, Graph Attention Networks operate most effectively on sparse graphs that represent only the most salient biological connections. We therefore devised and implemented a novel strategy to convert the dense interaction data into a sparse, unweighted **gene-gene graph** for each tissue. The core principle is that of **TF co-regulation**: an edge is created between two genes if they are both strongly regulated by the same transcription factor in a given tissue, implying a potential functional relationship. This was implemented with a dynamic, TF-specific thresholding strategy, as detailed in Algorithm 4. For each TF, a unique threshold is calculated based on the mean and standard deviation of its interaction strengths with all genes. This adaptive approach is more robust than a single global threshold, as it accounts for the fact that different TFs have vastly different regulatory profiles.

Algorithm 4 Gene-Gene Graph Construction from TF Co-Regulation

Input: Raw Interaction Matrices $\{M_t\}_{t=1}^{36}$, Threshold factor $\beta = 2.5$, Edge cap $C_{max} \approx 1,000,000$
Output: Edge Index Tensors $\{E_t\}_{t=1}^{36}$ for each tissue graph
for each tissue t from 1 to 36 **do**
    $P_t \leftarrow \emptyset$ ▷ Initialize empty set for unique undirected gene pairs
    for each TF f from 1 to N_{TFs} **do**
        $\mathbf{v}_f \leftarrow \text{GetVectorOfAbsoluteScores}(M_t, f)$ ▷ Scores for TF f
        if $\text{count}(\mathbf{v}_f > 0) < 2$ **then continue**
        end if
        $\mu_f, \sigma_f \leftarrow \text{Mean}(\mathbf{v}_f), \text{StdDev}(\mathbf{v}_f)$
        $T_f \leftarrow \mu_f + (\beta \cdot \sigma_f)$ ▷ Calculate dynamic, TF-specific threshold
        $G_f^{\text{strong}} \leftarrow \{g \mid |\text{score}(f, g)| > T_f\}$ ▷ Identify strongly regulated genes
        if $|G_f^{\text{strong}}| \geq 2$ **then**
            $\text{Pairs}_f \leftarrow \text{GetAllUniquePairs}(G_f^{\text{strong}})$ ▷ Create a clique from the set
            $P_t \leftarrow P_t \cup \text{Pairs}_f$ ▷ Add co-regulation edges to the tissue's set
        end if
    end for
    if $|P_t| > C_{max}$ **then** ▷ Apply optional edge capping for very dense graphs
        $P_t \leftarrow \text{RandomSample}(P_t, C_{max})$
    end if
    $E_t \leftarrow \text{ConvertToDirectedEdgeIndex}(P_t)$ ▷ Create (2, NumEdges) tensor
end for
return $\{E_t\}_{t=1}^{36}$

This method transforms the dense numerical matrices into 36 sparse, unweighted graph structures (E_t), each represented by an **edge index tensor** in the format required by PyTorch Geometric. This graph creation step is a critical part of our methodology, designed to distill the most salient co-regulatory relationships from the potentially noisy, computationally inferred network data, providing a meaningful structural input for the GAT encoder.

3.5 Experimental Design

To rigorously validate the proposed BWAF framework and systematically investigate the contributions of its constituent parts, a comprehensive experimental design was established. This section serves as a complete blueprint for the empirical evaluation that will be presented in Chapter 4. It begins by detailing the standardized data partitioning strategy and the evaluation metrics used across all experiments. It then defines the suite of models under evaluation including the proposed model, a faithful replication of a state-of-the-art baseline, and a series of ablation variants. The section concludes by outlining the specific protocols for model training, evaluation, and the interpretability analysis that was performed

to validate the model's internal logic.

3.5.1 Data Partitioning and Evaluation Metrics

All experiments were conducted on the final curated dataset, which consists of 40,056 sequences and is perfectly balanced with 20,028 promoter (positive class) and 20,028 non-promoter (negative class) samples. To ensure a fair and reproducible comparison across all models, this dataset was partitioned once into three distinct, non-overlapping sets using a fixed random seed (`RANDOM_SEED=42`):

- **Training Set (70%):** 28,040 samples, used exclusively for optimizing model parameters via backpropagation.
- **Validation Set (15%):** 6,008 samples, used for hyperparameter tuning and selecting the best model checkpoint from each training run (early stopping).
- **Test Set (15%):** 6,008 samples, held out during all model development and used only once for the final, unbiased evaluation.

Following this data partitioning, the performance of each model on the binary classification task was assessed using a comprehensive suite of standard metrics. The metrics used are defined as follows, where TP, TN, FP, and FN refer to True Positives, True Negatives, False Positives, and False Negatives, respectively:

- **Accuracy (Acc):** The overall proportion of correct predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.16}$$

- **Precision (Prec):** The proportion of positive predictions that were correct.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.17}$$

- **Recall (Sensitivity, Rec):** The proportion of actual positives that were correctly identified.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.18}$$

- **Specificity (Spec):** The proportion of actual negatives that were correctly identified.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{3.19}$$

- **F1-Score (F1):** The harmonic mean of Precision and Recall.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.20}$$

- **Area Under the ROC Curve (AUC-ROC):** A threshold-independent measure of the model's ability to discriminate between the positive and negative classes.

3.5.2 Models Under Evaluation

To validate the primary hypotheses of this thesis, a suite of models was trained and evaluated on the standardized data partitions. This suite includes the proposed BWAF model, a faithful replication of a state-of-the-art baseline, and a series of carefully designed ablation models. This systematic approach allows for a direct comparison of performance and a thorough dissection of the sources of those results.

The Proposed BWAF Model

The primary model under evaluation is the full **BWAF-Net**, as detailed in Section 3.2. This model features the end-to-end integration of the Transformer encoder, the GAT encoder, and the novel Biologically Weighted Attention Fusion (BWAF) layer, utilizing all three data modalities (sequence, network, and priors).

Baseline Model: Replication of msBERT-Promoter

To contextualize the performance of the proposed BWAF model, a powerful and recent state-of-the-art method, **msBERT-Promoter** [82], was selected as the primary baseline for comparison. This model was chosen for its high reported performance and its architectural relevance as an advanced, sequence-only Transformer-based approach. As a unimodal model, it serves as the perfect control to test the central hypothesis that integrating network context and biological priors provides a significant advantage.

Justification for Dataset Modification and Replication. A direct performance comparison using the original *msBERT-Promoter* dataset and its pre-trained model was scientifically infeasible and inappropriate for three key reasons. First, the biological contexts are fundamentally different: the original study focused on short, **81bp prokaryotic (E. coli) promoter sequences**, while this thesis addresses a more complex problem involving **2000bp human eukaryotic promoters**. Second, our multi-modal BWAF model has a strict requirement for

human-specific gene regulatory network data, a modality that does not exist for the prokaryotic dataset. Testing our model on their data was therefore impossible.

Third, and most importantly, the goal of this comparison is not to benchmark performance on different biological problems, but to conduct a scientifically valid comparison of **methodological principles** on the same task. Therefore, to create the fairest possible head-to-head comparison, we faithfully replicated the core architectural and ensemble strategy of msBERT-Promoter and trained it from scratch on our own curated human promoter dataset.

Replication Methodology. The replication methodology, detailed in Algorithm 5, was designed to mirror the central principles of the baseline paper. The process involved three main stages:

1. **Multi-Scale K-mer Tokenization:** The 2000bp DNA sequences were tokenized into four distinct representations using overlapping k-mers of lengths k=3, 4, 5, and 6. This creates four parallel views of the sequence data, each capturing patterns at a different scale.

2. **Independent Transformer Training:** Four independent `KmerTransformer` models were trained from scratch, one for each k-mer representation. While the original msBERT-Promoter fine-tuned a large pretrained model (DNABERT), our replication trained smaller Transformers (2 layers, 4 attention heads) from scratch to be comparable in scale to a single branch of our proposed model and to demonstrate the power of the ensemble strategy itself.

3. **Soft-Voting Ensemble:** For evaluation on the test set, the probabilistic outputs from the four trained models were combined using a soft-voting ensemble. The final prediction for a given sequence is the arithmetic mean of the probabilities predicted by the four individual k-mer models.

This strategy ensures that any observed performance differences are directly attributable to the architectural and data modality choices of our BWAF model (i.e., multi-modal integration with guided fusion) versus the baseline's strategy (multi-scale sequence ensembling), as both are evaluated on the exact same data splits and task. This comparison deliberately contrasts a design philosophy based on information synergy (BWAF) with one based on computational scale (the ensemble).

Algorithm 5 Baseline Replication Methodology (msBERT-Promoter)

Input: Raw DNA sequences $S_{\text{train}}, S_{\text{test}}$, Labels $Y_{\text{train}}, Y_{\text{test}}$
Parameters: K-mer values $K = \{3, 4, 5, 6\}$
Output: Final ensemble predictions for the test set, P_{ensemble}
$V \leftarrow \emptyset$ ▷ Initialize map for k-mer vocabularies
for each k in K **do**
    $V_k \leftarrow \text{BuildKmerVocabulary}(S_{\text{train}}, k)$ ▷ Build vocab from training data
    Add V_k to V
end for
$M_{\text{trained}} \leftarrow \emptyset$ ▷ Initialize map for trained models
for each k in K **do**
    $D_k \leftarrow \text{TokenizeAndCreateDataset}(S_{\text{train}}, Y_{\text{train}}, k, V_k)$
    $M_k \leftarrow \text{InitializeKmerTransformer}(V_k)$
    $\text{TrainModel}(M_k, D_k)$ ▷ Train the k-mer specific model
    Add M_k to M_{trained}
end for
$P_{\text{all\_k}} \leftarrow []$ ▷ List to store predictions from each model
for each k in K **do**
    $D_{k,\text{test}} \leftarrow \text{TokenizeDataset}(S_{\text{test}}, k, V_k)$
    $P_k \leftarrow \text{PredictProbabilities}(M_{\text{trained}}[k], D_{k,\text{test}})$
    Append P_k to $P_{\text{all\_k}}$
end for
$P_{\text{ensemble}} \leftarrow \text{Average}(P_{\text{all\_k}})$ ▷ Soft-voting ensemble
return P_{ensemble}

Ablation Models

To systematically investigate the contribution of each data modality and to validate the novelty of the BWAF fusion mechanism, five ablated models were designed and implemented. Each model represents a simplification of the full architecture, allowing for a controlled assessment of its impact on performance. The architectural design of each model is illustrated in Figure 3.7. The five models are:

- **A1: Transformer + Priors:** This bi-modal model removes the GAT branch entirely. It fuses the sequence features from the Transformer encoder directly with the 11 biological priors via simple concatenation, followed by a classifier. Its performance, when compared to the full model, is used to measure the **contribution of the gene regulatory network context**.
- **A2: GAT + Priors:** This bi-modal model removes the Transformer branch. It concatenates the pre-computed GAT features for each gene with the biological priors and feeds them to a classifier. This measures the **contribution of the deep sequence features** learned by the Transformer.
- **A3: Simple Concatenation Fusion:** This model includes all three data

modalities (Sequence, Network, and Priors) but replaces the novel BWAF layer with a standard, naive fusion strategy. The features from all three branches are concatenated into a single vector and passed to a classifier. This is the most critical ablation, as comparing its performance directly against the full BWAF model quantifies the **benefit of the proposed prior-guided dynamic fusion mechanism**.

- **A4: Transformer-Only:** A unimodal baseline that uses only the Transformer branch to make predictions from sequence data, followed by a classifier. This establishes the performance of a powerful, purely data-driven sequence model and helps to contextualize the benefit of adding other modalities.

- **A5: Priors-Only:** A simple MLP baseline that uses only the 11 biological prior features as input. This model measures the **raw predictive power of the curated domain knowledge** and serves as a strong, non-deep-learning baseline.

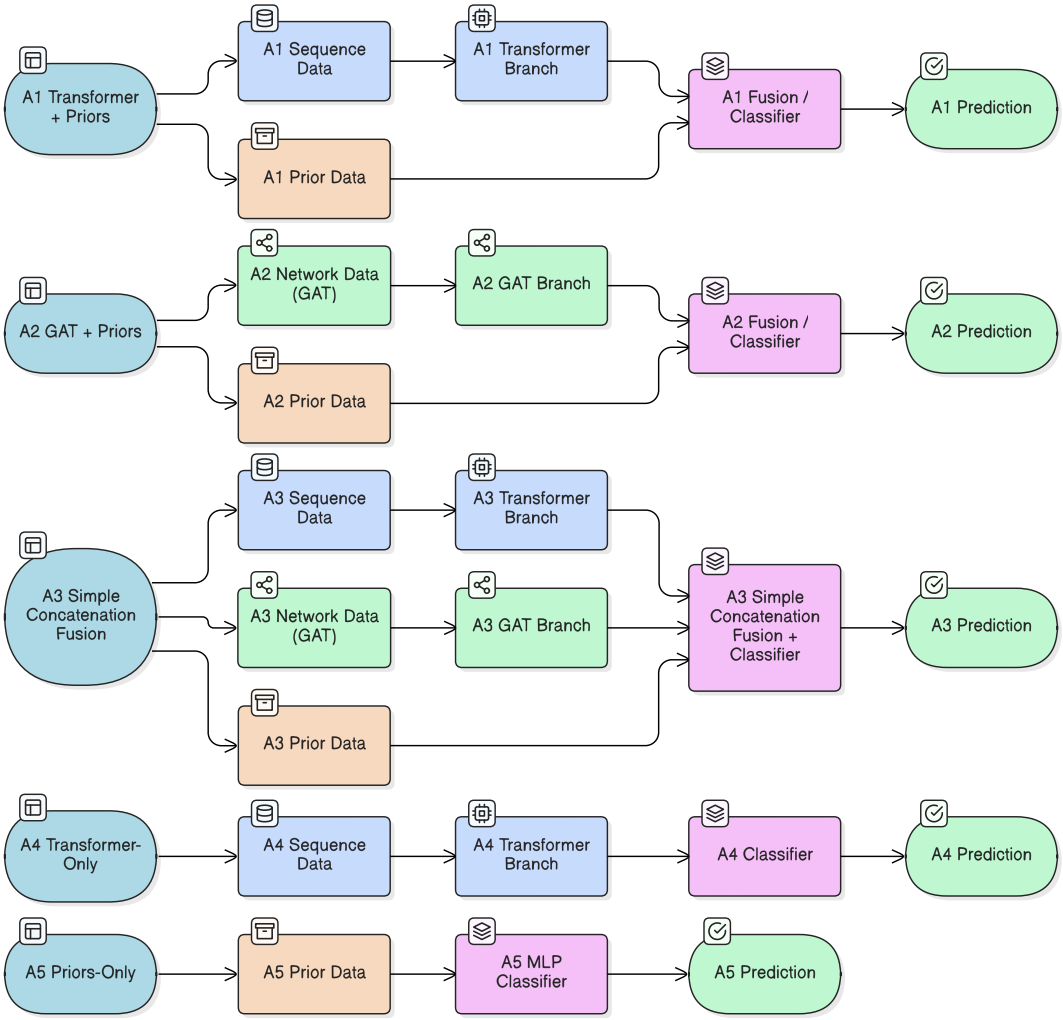

Figure 3.7: Architectural diagrams of the five models used in the ablation study. Each model is a variant of the full BWAF framework, designed to isolate the contribution of specific data modalities (Sequence, Network, Priors) and the fusion mechanism (BWAF vs. Concatenation).

3.5.3 Experimental Protocols

To ensure consistency and fairness, all models were subjected to a standardized set of experimental protocols for training, evaluation, and interpretability analysis.

Training and Evaluation Protocol

Each of the models under evaluation the full BWAF-Net, the baseline replication, and all five ablation models was trained using an identical protocol. Models were trained for a maximum of **10 epochs** using the AdamW optimizer. A 'ReduceLROnPlateau' learning rate scheduler was employed, which reduces the learning

rate if the validation loss does not improve for a set number of consecutive epochs. After each epoch, the model's performance was measured on the validation set. The model weights that yielded the **lowest validation loss** over the 10 epochs were saved as the "best model." This early stopping strategy prevents overfitting and ensures that the model selected is the one that generalizes best to unseen data.

Following the completion of training, this single best model checkpoint was loaded and evaluated exactly once on the held-out test set to generate the final, unbiased performance metrics reported in Chapter 4.

3.5.4 Interpretability Analysis Protocol

A central objective of this thesis was to develop a model that is not only accurate but also transparent. To move beyond performance metrics and understand the model's internal decision-making process, a two-pronged explainability analysis was designed. The primary goal was to validate the novel BWAF layer by answering the question: *Does the BWAF layer learn a biologically plausible fusion strategy?* This was achieved by investigating the learned mapping between the 11 biological priors and the two attention weights (α_{seq} and α_{graph}) using two complementary methods.

First, to understand the global, high-level trends, a **Spearman Correlation Analysis** was performed. This non-parametric method was chosen to measure the strength and direction of the monotonic relationship between each prior feature and the generated attention weights, without assuming a linear relationship. For all 6,008 samples in the test set, the alpha weights were extracted, and a correlation matrix was computed to reveal simple, monotonic associations.

Second, to capture more complex, non-linear relationships and to formally quantify the marginal contribution of each feature, **SHAP (SHapley Additive exPlanations)** was employed [140]. SHAP is a game-theoretic approach that can explain a model's output by computing the contribution of each feature to a prediction. It was specifically chosen for its ability to reveal not just the magnitude of a feature's impact but also its effect on individual predictions. We used the model-agnostic 'KernelExplainer' to explain the outputs of the two `weight_generator` MLPs within the BWAF layer, using the 11 prior features as inputs. SHAP values were computed for a representative background of 100 samples and a target set of 500 samples from the test set.

This dual-method approach was deliberately designed to provide a complete picture. Correlation reveals the overall trend, while SHAP reveals the local, context-dependent feature importances. This allows for the discovery of sophis-

ticated, non-linear policies learned by the model, where a feature can be highly important (high SHAP value) but have a weak or even negative overall correlation.

3.5.5 Implementation Details

All models, data processing pipelines, and analyses were implemented in Python 3.10. The deep learning architectures were built using the **PyTorch** [141] and **PyTorch Geometric** [142] libraries. All experiments were conducted on a system with an Intel Core i7 CPU and 16GB of RAM. The complete source code is provided in the supplementary materials.

The selection of hyperparameters was guided by a combination of established best practices, empirical testing, and hardware constraints. Key training hyperparameters are summarized in Table 3.4. The AdamW optimizer was chosen as it is the standard for training Transformer-based models, and BCEWithLogitsLoss was used as it is the most appropriate and numerically stable loss function for this binary classification task. A 'ReduceLROnPlateau' scheduler was employed to dynamically adjust the learning rate based on validation performance.

The number of training epochs was set to a maximum of 10. Observation of the validation loss curves during initial experiments showed that all models converged rapidly, typically reaching a minimum loss well before this limit. The use of an early stopping protocol, which saves the model checkpoint with the lowest validation loss, ensures that the selected model is the one that generalizes best, regardless of the fixed number of epochs. The batch sizes of 16 and 32 were determined to be the largest possible that could be accommodated by the available system memory for the respective model architectures.

Table 3.4: Key Training Hyperparameters.

Hyperparameter	**Value / Setting**
Universal Parameters	
Optimizer	AdamW
Weight Decay	1×10^{-5}
Loss Function	Binary Cross-Entropy with Logits Loss
Max Epochs	10
Learning Rate Scheduler	ReduceLROnPlateau (patience=3)
Model-Specific Parameters	
BWAF & Ablation Models Learning Rate	0.0005
BWAF & Ablation Models Batch Size	16
Baseline (k-mer) Models Learning Rate	0.0001
Baseline (k-mer) Models Batch Size	32

3.6 Research Methodology: Design Science Research

The overarching methodology guiding this work is the **Design Science Research (DSR)** paradigm [143]. DSR is particularly suited for this study as it is an engineering-focused approach that emphasizes the creation and evaluation of innovative artifacts such as models, methods, or systems to solve specific, real-world problems. In this context, the identified problem is the accurate identification of promoter regions by integrating heterogeneous biological data. The primary **artifact** is the novel BWAF-Net model, which combines Transformers and Graph Attention Networks under the explicit guidance of biological priors.

The DSR process is inherently iterative, and for this thesis, it was adapted into five distinct phases. This structured process ensures that the research produces a well-designed, thoroughly evaluated, and impactful contribution to the field. The phases are as follows:

- **Problem Identification and Motivation:** As detailed in the Literature Review (Chapter 2), a thorough analysis of existing promoter prediction methods was conducted to identify their limitations, particularly concerning multi-modal data integration. This phase culminated in the formulation of the research questions that guide this study.
- **Objective Definition:** Based on the identified problem, the objective was defined as the creation of a multi-modal deep learning model that leverages a novel, biologically-informed fusion strategy. The specific objectives, outlined in Section 1.5, provided a clear roadmap for the artifact's design.
- **Design and Development:** This was the core artifact-building phase of the research. It involved the detailed design, implementation, and refinement of the BWAF-Net model, including its constituent Transformer and GAT branches and the novel fusion mechanism, as specified in Section 3.2.
- **Demonstration and Evaluation:** In this phase, the developed artifact was applied to the problem and rigorously evaluated. As detailed in the Experimental Design (Section 3.5) and presented in the Results (Chapter 4), the model's performance was measured against a state-of-the-art baseline and a series of ablation studies to demonstrate its utility and superiority.
- **Knowledge Contribution:** The final phase involves communicating the research findings. This thesis makes several distinct contributions: (1) The **BWAF-Net model** itself, as a validated and effective artifact for promoter

prediction; (2) The **prior-guided fusion mechanism**, a novel methodological contribution to the field of multi-modal AI; (3) **Empirical evidence** from the comparative and ablation studies that demonstrates the superiority of this guided-fusion approach over naive concatenation and unimodal baselines; and (4) **New insights** from the interpretability analysis that confirm the model learns a biologically plausible fusion strategy.

A visual summary of this adapted DSR process is provided in Figure 3.8.

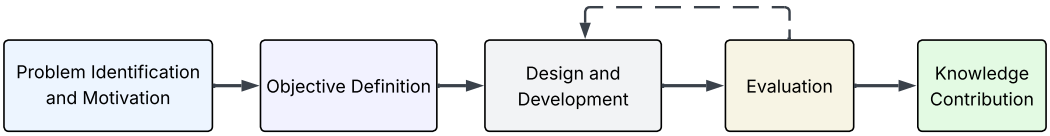

Figure 3.8: Flowchart illustrating the Design Science Research (DSR) methodology as adapted for this thesis, based on the framework by De Sordi [143].

Chapter 4

Results and Discussion

4.1 Introduction

This chapter presents the comprehensive empirical evaluation of the Biologically Weighted Attention Fusion (BWAF) framework, following the methodology and experimental design detailed in Chapter 3. The chapter is structured to first present the experimental findings objectively in the *Performance Results* section, followed by a unified *Discussion* of their implications. The results are detailed systematically, beginning with the performance of the full BWAF model, followed by a comparative analysis against the replicated baseline, the findings from the ablation studies, and the outcomes of the interpretability analysis. The chapter then culminates in a discussion that synthesizes these findings, validates the core hypotheses, and discusses the broader contributions of this work.

4.2 Performance Results

4.2.1 BWAF Model Performance

The performance of the proposed Biologically Weighted Attention Fusion (BWAF) model was established by evaluating the fully trained model on the held-out test set, which comprises 6,008 samples. The quantitative performance, predictive reliability, and training dynamics were assessed to provide a comprehensive benchmark of its effectiveness.

The model's performance on standard binary classification metrics is summarized in Table 4.1. The model achieved an accuracy of 99.87%, correctly classifying 5,992 out of 6,008 samples. The AUC-ROC of 99.99% indicates a strong separation between the promoter and non-promoter classes in the model's learned feature space.

Table 4.1: Performance metrics of the full BWAF model on the held-out test set.

Model	Accuracy	AUC-ROC	F1-Score	Precision	Recall	Specificity
BWAF (Full)	**99.87%**	**99.99%**	**99.87%**	**100.00%**	**99.73%**	**100.00%**

An analysis of the model's error profile provides further insight into its predictive reliability. The confusion matrix, presented in Figure 4.1, reveals two key characteristics of the model's performance. First, the model achieved 100.00% precision and 100.00% specificity, corresponding to zero false positives (FP = 0). Second, the model correctly identified 3,010 of the 3,018 true promoters, resulting in a recall of 99.73% with only eight false negatives (FN = 8).

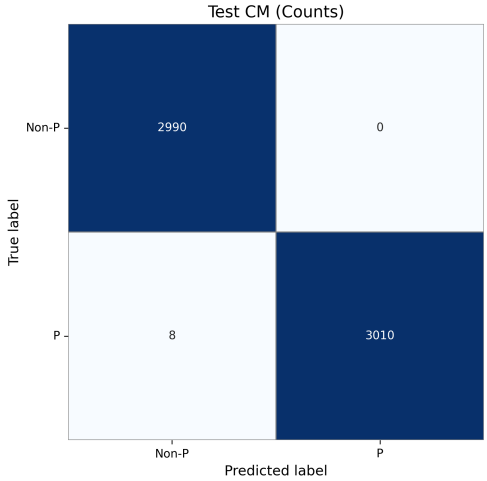

Figure 4.1: Confusion matrix of the full BWAF model on the test set (N=6,008). The model achieved 100% precision with zero false positives.

The model's learning behavior and generalization capability were assessed by analyzing the training and validation loss curves over 10 epochs, as shown in Figure 4.2. The curves demonstrate a stable convergence, with both training and validation losses decreasing consistently. The minimal gap between the two curves suggests effective generalization to unseen data with low risk of overfitting. The model checkpoint from epoch 8 was selected for final evaluation, corresponding to the minimum validation loss of 0.0060. These training dynamics are consistent with an efficient and robust learning process.

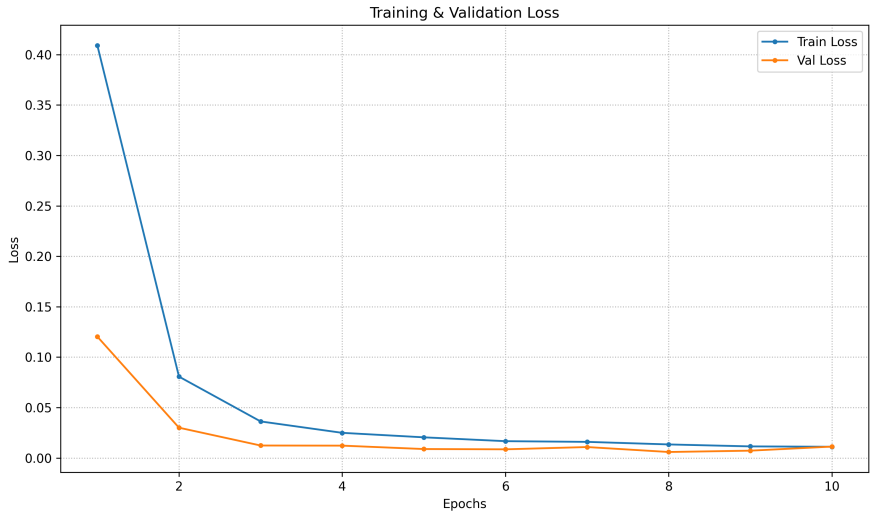

Figure 4.2: Training and validation loss curves for the full BWAF model over 10 epochs. The best model checkpoint, based on the minimum validation loss, was saved at epoch 8.

4.2.2 Comparative Analysis against Baseline

To contextualize its performance, the BWAF model was benchmarked against the replicated state-of-the-art baseline, msBERT-Promoter. The comparison focused on both predictive accuracy on the identical held-out test set and architectural efficiency.

Table 4.2 presents the head-to-head performance comparison. For additional context, the performance metrics reported in the original msBERT-Promoter paper on their prokaryotic dataset are also included. The results indicate a substantial performance difference between the BWAF model and the baseline replication on our dataset across all metrics.

Table 4.2: Performance Comparison: BWAF vs. Replicated Baseline (on our dataset) and Reported SOTA (on their dataset).

Model	Accuracy (%)	AUC-ROC (%)	F1-Score (%)	Precision (%)	Recall (Sn, %)	Specificity (Sp, %)
Our BWAF Model (on our dataset)	**99.87**	**99.99**	**99.87**	**100.00**	**99.73**	**100.00**
msBERT-Promoter (Replication on our dataset)	78.33	85.39	75.33	87.96	65.87	90.90
msBERT-Promoter (Reported by Li et al., 2024)	*96.20*	*99.40*	–	–	*97.30*	*95.10*

The difference in predictive reliability is further illustrated by the confusion matrices in Figure 4.3 and the raw error counts in Table 4.3. The BWAF model made only 8 errors, all of which were False Negatives, while the baseline made a total of 1,302 errors.

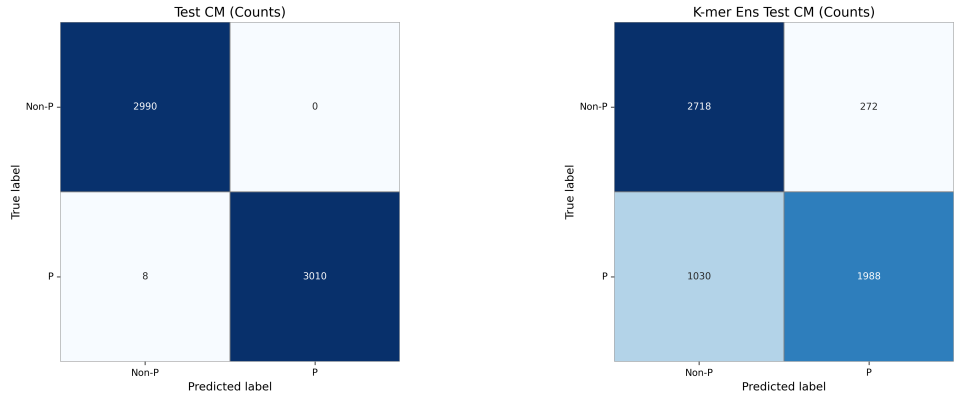

Figure 4.3: Confusion matrices on the test set for the BWAF model (left) and the replicated msBERT-Promoter baseline (right).

Table 4.3: Error Analysis: False Positives (FP) and False Negatives (FN) on the Test Set.

Model	**False Positives (FP)**	**False Negatives (FN)**
Our BWAF Model	**0**	**8**
msBERT-Promoter (Replication)	272	1030

In addition to predictive performance, the architectural efficiency of the two models was compared. Table 4.4 details the total number of trainable parameters, training time, and inference complexity for the single BWAF model versus the four-model baseline ensemble.

Table 4.4: Efficiency Comparison of the BWAF Model vs. the msBERT-Promoter Replication.

Parameter / Metric	**Our BWAF Model**	**msBERT-Promoter (Replication)**
Architecture	Single, Integrated Multi-Modal	Ensemble of 4 Separate Models
Total Trainable Parameters	~292,601	~937,220
Total Training Time (CPU)	~79 hours	~253 hours
Inference Complexity	One forward pass	Four forward passes + Averaging

4.2.3 Ablation Study Results

To understand the individual and synergistic contributions of the model's core components, a series of ablation studies were conducted. As detailed in Chapter 3, components of the full model were systematically removed or replaced, and each

ablated variant was trained and evaluated on the identical data splits to allow for a direct assessment of its impact.

The comprehensive results of these studies are summarized in Table 4.5. This table presents the performance of all five ablated models (A1-A5) compared against the full BWAF model. A visual summary of the performance hierarchy is provided in the bar chart in Figure 4.4. Additionally, the training and validation loss curves for two key baselines, A4 (Transformer-Only) and A5 (Priors-Only), are shown in Figure 4.5 to illustrate their learning dynamics.

Table 4.5: Full Results of Ablation Studies on the Test Set. Performance metrics are compared for the full BWAF model and all five ablated variants.

Model	Accuracy (%)	AUC-ROC (%)	F1-Score (%)	Precision (%)	Recall (%)	FN Count	FP Count
Our BWAF Model (Full)	**99.87**	**99.99**	**99.87**	**100.00**	**99.73**	**8**	**0**
A5: Priors-Only (MLP)	97.66	99.69	97.62	100.00	95.36	140	0
A1: Transformer+Priors	94.19	98.76	94.20	94.37	94.03	180	169
A2: GAT+Priors	94.04	98.50	93.69	100.00	88.13	358	0
A3: Simple Concat Fusion	93.70	98.87	93.91	91.35	96.62	102	276
A4: Transformer-Only	71.23	78.15	68.50	76.13	62.25	1139	589

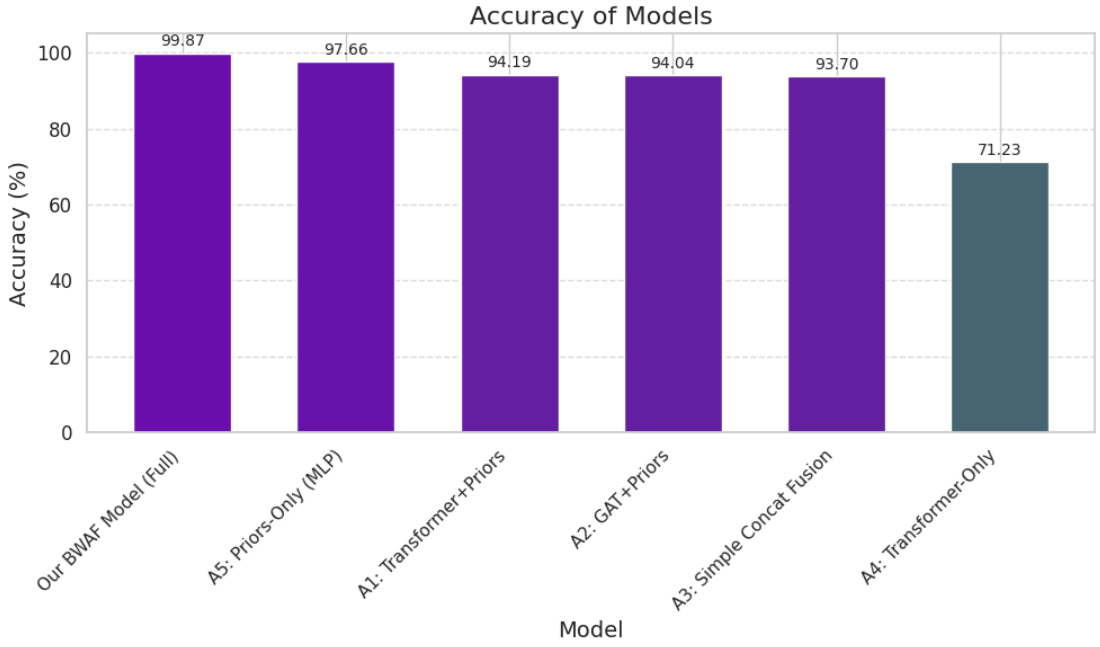

Figure 4.4: Performance comparison of the full BWAF model against all ablation variants on the test set, illustrating the performance hierarchy.

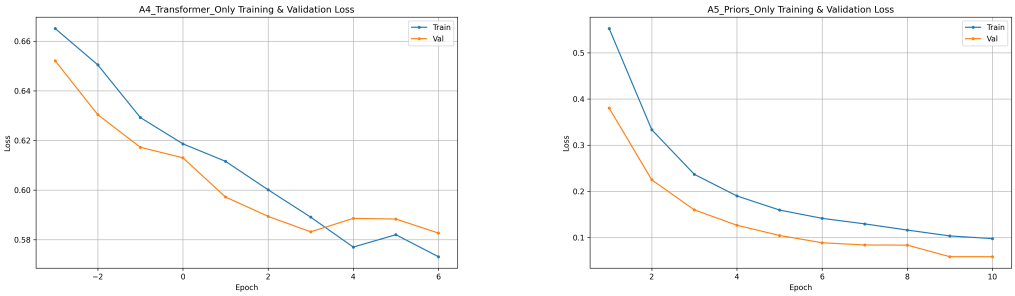

Figure 4.5: Training and validation loss curves for (a) A4: Transformer-Only and (b) A5: Priors-Only, showing the difference in convergence and final loss.

4.2.4 Interpretability (XAI) Results

A central objective of this research was to develop a model whose internal decision-making is transparent. To investigate whether the BWAF layer learned a biologically plausible fusion strategy, the two-pronged analysis protocol defined in Chapter 3 was executed.

First, SHAP values were calculated to formally quantify the importance of each of the 11 biological priors in generating the attention weights (α_{seq} and α_{graph}). The SHAP summary plots in Figure 4.6 visualize the magnitude and direction of each feature's impact. The precise quantitative ranking of each feature's mean absolute SHAP value is provided in Table 4.6. The results show a clear divergence in the features that drive each weight: the TATA Box and CAAT Box are the most impactful features for the sequence weight, while the GC Box and Sp1 Binding Site are most impactful for the graph weight.

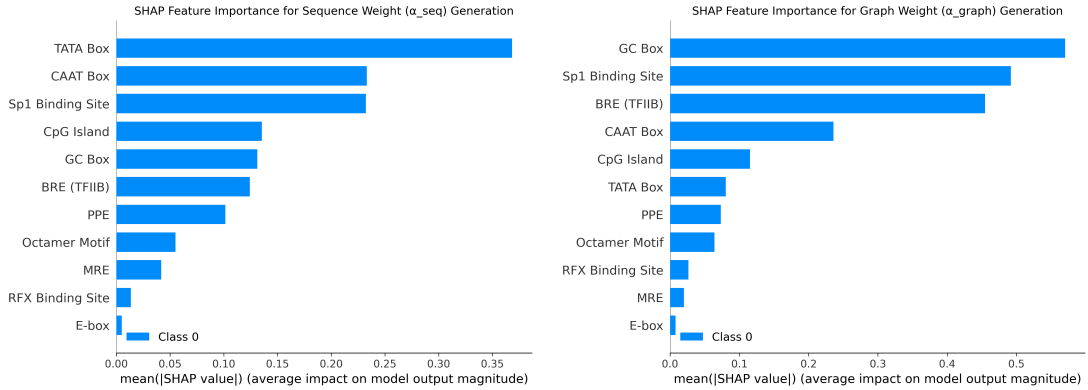

Figure 4.6: SHAP feature importance plots. The length of the bar indicates the mean absolute SHAP value (overall impact) of each biological prior feature on the learned attention weights for the sequence branch (α_{seq}, left) and the graph branch (α_{graph}, right).

Table 4.6: Ranked Feature Importance (Mean Absolute SHAP Value) for the Generation of BWAF Attention Weights.

Drivers of Sequence Weight (α_{seq})		**Drivers of Graph Weight (α_{graph})**	
Feature	**Mean Abs SHAP**	**Feature**	**Mean Abs SHAP**
TATA Box	0.368	GC Box	0.570
CAAT Box	0.233	Sp1 Binding Site	0.492
Sp1 Binding Site	0.232	BRE (TFIIB)	0.455
CpG Island	0.135	CAAT Box	0.236
GC Box	0.131	CpG Island	0.115
BRE (TFIIB)	0.124	TATA Box	0.080
PPE	0.101	PPE	0.073
Octamer Motif	0.055	Octamer Motif	0.064
MRE	0.042	RFX Binding Site	0.026
RFX Binding Site	0.013	MRE	0.020
E-box	0.005	E-box	0.008

Second, to complement the SHAP analysis, the global monotonic trends between the priors and the learned alpha weights were investigated using Spearman correlation. The resulting correlation matrix, visualized as a heatmap in Figure 4.7, reveals several strong relationships. For instance, there are strong positive correlations between the GC Box and α_{graph} (0.91) and strong negative correlations between the TATA Box and α_{seq} (-0.61).

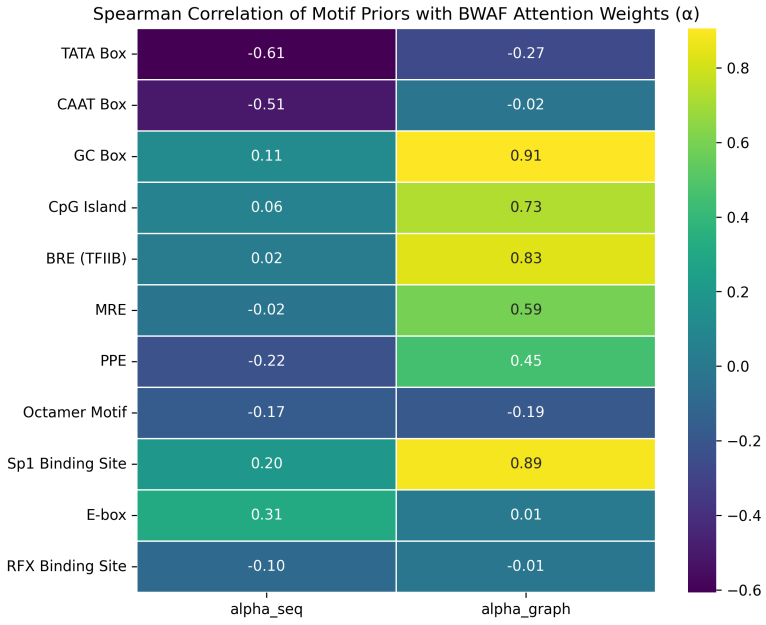

Figure 4.7: Spearman correlation heatmap between the 11 log-transformed biological prior features and the two learned alpha weights (α_{seq} and α_{graph}) across the test set.

The overall behavior of the learned weights is shown in Figure 4.8. For

true promoter samples (Label=1), the model consistently assigns higher weights to both modalities compared to non-promoter samples (Label=0), for which both weights are suppressed towards zero. Finally, an analysis of the Transformer's internal self-attention map for a sample promoter containing a TATA box revealed that the model's focus was highly distributed across the 2000bp sequence, rather than being concentrated on the single TATA box motif.

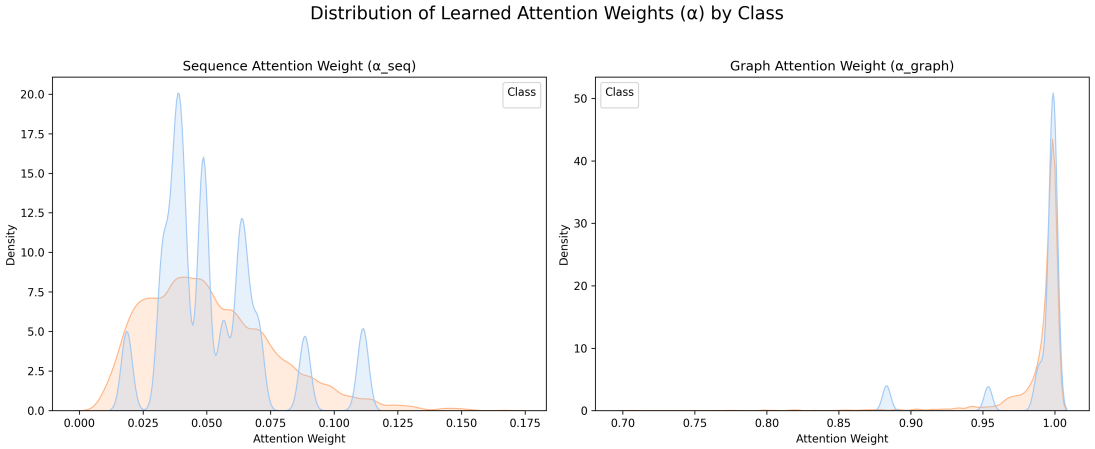

Figure 4.8: Violin plots showing the distribution of the learned attention weights α_{seq} (left) and α_{graph} (right) for true promoter (Label=1) and non-promoter (Label=0) samples in the test set.

4.3 Discussion

The experimental results presented in this chapter provide a multi-faceted validation of the proposed Biologically Weighted Attention Fusion (BWAF) framework. The findings confirm the central hypothesis that a multi-modal approach, intelligently integrating genomic sequence, network context, and explicit prior knowledge, is superior to advanced, sequence-only baselines for human promoter identification. The full BWAF model achieved 99.87% accuracy, a performance that substantially surpasses the 78.33% accuracy of the replicated `msBERT-Promoter` ensemble. This outcome is not merely a measure of predictive success but a validation of architectural philosophy. The BWAF model achieves its state-of-the-art performance while being approximately **3.2 times smaller** in parameter count (Table 4.4). This efficiency underscores a key finding: solving the problem with **information synergy** by providing the model with diverse, complementary data streams is more effective and efficient than a brute-force computational approach that relies on a single modality.

A key insight from the ablation studies is the foundational power of explicit biological priors. The simple MLP model using only 11 motif counts (A5:

Priors-Only) achieved a 97.66% accuracy, substantially outperforming the powerful Transformer-Only model (A4: 71.23%). This result highlights a potential pitfall in purely data-driven deep learning: without guidance, complex models can struggle to learn foundational biological rules that are more effectively provided as explicit domain knowledge. However, the experiments also revealed that naive integration of this knowledge is a suboptimal strategy. The Simple Concatenation model (A3), despite having access to all three data streams, performed worse than the Priors-Only baseline at 93.70 % accuracy (Table 4.5). This critical result demonstrates that simply concatenating features dilutes the strong, discriminative signal from the low-dimensional priors with higher-dimensional, more complex features from the deep learning branches. The success of the full BWAF model, the only architecture to improve upon the Priors-Only baseline, provides compelling empirical validation for its novel fusion mechanism as a necessary solution to this signal dilution problem.

The interpretability analysis confirms that this successful fusion mechanism is not a black box but operates on a foundation of learned biological principles. As quantified in Table 4.6, the SHAP analysis revealed that the model learned to heavily weight the Transformer branch's output when strong canonical sequence motifs like the TATA and CAAT boxes were present, aligning with the known importance of these elements in core promoter function [131]. Conversely, for promoters rich in motifs associated with complex regulation, such as the GC Box, the model learned to prioritize the contextual information from the GAT branch, a strategy consistent with the regulation of ubiquitously expressed housekeeping genes [132]. Furthermore, the correlation analysis (Figure 4.7) uncovered a sophisticated, non-linear policy: the model's reliance on the full sequence context (a higher α_{seq}) is greatest for promoters that *lack* strong canonical motifs. This suggests the model has learned to handle the complexity of TATA-less promoters, which are known to rely on a more dispersed and complex set of regulatory signals [124]. This level of transparency enhances the trustworthiness and scientific value of the model's predictions.

The primary contribution of this work to the field of Artificial Intelligence is the validation of the BWAF mechanism as a new paradigm for multi-modal fusion, where domain knowledge acts as an active controller of information flow. This "knowledge-as-a-controller" pattern injects a strong inductive bias, enhances interpretability, and improves sample efficiency, offering a generalizable template for other scientific domains. For computational genomics, this thesis contributes a state-of-the-art framework that sets a new performance benchmark for human promoter identification. By holistically integrating deep sequence representations, multi-context network data, and explicit motif priors, the BWAF model provides

a highly reliable tool for genome-wide screening.

While this study achieved its primary objectives, it is important to acknowledge its limitations. The model's performance is dependent on the quality of the computationally inferred gene regulatory networks used in this work. The analysis was also intentionally limited to protein-coding genes to establish a focused baseline, and the model provides a static prediction for a given sequence. Finally, as with any machine learning model, performance has only been validated on the specific dataset curated for this study. Acknowledging these limitations is crucial for contextualizing the findings and framing the contributions of the robust and interpretable framework established in this thesis.

Chapter 5

Conclusion and Recommendation

5.1 Conclusion

The accurate identification of gene promoter regions is a foundational challenge in genomics, pivotal to understanding the complex orchestration of gene expression. This thesis addressed the limitations of existing computational methods, which often struggle to effectively integrate the diverse signals that govern promoter activity. We hypothesized that a multi-modal deep learning framework, designed to synergistically combine DNA sequence patterns, context-specific gene regulatory networks, and explicit biological domain knowledge, would yield a more accurate and interpretable solution.

To this end, we introduced the **Biologically Weighted Attention Fusion (BWAF)** framework, a novel architecture centered on a prior-guided dynamic fusion mechanism. The BWAF model integrates a Transformer branch to capture long-range dependencies in 2000bp DNA sequences, a Graph Attention Network (GAT) branch to model the regulatory context from 36 normal human tissue networks, and a feature stream of 11 quantified biological priors.

The comprehensive experimental evaluation presented in Chapter 4 provides unequivocal support for our approach. The full BWAF model achieved state-of-the-art performance on our curated test set, with **99.87% accuracy**, **99.99% AUC-ROC**, and a perfect **precision of 100.00%**. This performance significantly surpassed that of our replication of the `msBERT-Promoter` ensemble, a powerful sequence-only baseline, demonstrating the substantial value of multi-modal integration. Furthermore, our ablation studies yielded critical insights into the nature of the prediction task. We empirically demonstrated that the 11 biological priors are exceptionally predictive on their own, yet naive fusion of these priors with deep learned features results in a suboptimal, degraded performance. The success of the full BWAF model validated our core contribution: the novel

BWAF layer, which uses priors not as passive features but as active controllers of information flow, is essential for successfully integrating all three data streams. Our interpretability analysis confirmed that this learned fusion policy is not a black box but is grounded in biologically plausible principles.

In conclusion, this thesis successfully designed, implemented, and validated a novel deep learning framework that is highly effective, computationally efficient, and interpretable. The findings validate the central hypothesis that intelligent, prior-guided fusion of complementary data modalities is a superior strategy for complex problems in regulatory genomics. The BWAF model stands as a significant contribution to the field, offering both a powerful predictive tool and a new architectural paradigm for building biologically-informed AI.

5.2 Recommendations for Future Work

The success and architecture of the BWAF framework open up numerous exciting avenues for future research. The following recommendations aim to build upon the findings of this thesis, expanding the model's capabilities, deepening its biological insights, and broadening its applicability.

Exploring Biological Frontiers via Model Outliers. With near-perfect performance, the few instances where the model's predictions diverge from the ground truth represent opportunities for new biological discovery. A detailed biological analysis of the eight false negative promoters could reveal novel or highly divergent regulatory mechanisms. Furthermore, a systematic study of the learned α_{seq} and α_{graph} weights across different gene families (e.g., tissue-specific vs. housekeeping genes) could provide genome-wide insights into dominant regulatory strategies.

Enhancing the Model with New Data Modalities. The BWAF framework is extensible and could be enriched by integrating additional sources of biological information. Epigenetic data, such as DNA methylation and histone modifications (e.g., H3K4me3, H3K27ac), could be added to the biological prior vector to guide the fusion policy based on chromatin state. Additionally, a third deep learning branch could be added to process chromatin accessibility data (e.g., from ATAC-seq).

Generalizing the Framework to New Genomic Tasks. The core architecture of BWAF can be adapted to address other fundamental problems in genomics. The framework could be modified for **enhancer prediction and enhancer-promoter linking** by using the GAT branch to model 3D chromatin interactions

from Hi-C data. Alternatively, it could be transitioned from a binary classifier to a regression model for **promoter strength prediction**, using gene expression levels (e.g., from GTEx) as the target variable.

Assessing Cross-Species Generalization and Robustness. The model's robustness and the conservation of its learned principles can be tested in new contexts. Training and testing the model on another mammalian genome, such as mouse, would be a valuable exercise in **cross-species evaluation** and comparative genomics. Investigating the model's performance when certain data modalities are missing would test its **robustness to data sparsity**, a crucial aspect for real-world applications.

Refining Network Representations. The GAT branch, a key component of the model, can be further improved. More sophisticated **graph construction methods** could be explored, such as creating bipartite TF-gene graphs directly. The GAT layers could also be extended to **incorporate edge features**, such as interaction scores, to further enrich the network representations.

References

[1] Liliana Villao-Uzho et al. “Plant promoters: their identification, characterization, and role in gene regulation”. In: *Genes* 14.6 (2023), p. 1226.

[2] Andreas Karollus, Tobias Mauermeier, and Julien Gagneur. “Current sequence-based models capture gene expression determinants in promoters but mostly ignore distal enhancers”. In: *Genome Biology* 24.1 (2023), p. 215.

[3] Emily G Brooks et al. “Plant promoters and terminators for high-precision bioengineering”. In: *BioDesign Research* 5 (2023), p. 0013.

[4] Cara Deal, Lien De Wannemaeker, and Marjan De Mey. “Towards a rational approach to promoter engineering: understanding the complexity of transcription initiation in prokaryotes”. In: *FEMS Microbiology Reviews* 48.2 (2024), fuae004.

[5] Gergely Nagy et al. “Lineage-determining transcription factor-driven promoters regulate cell type-specific macrophage gene expression”. In: *Nucleic acids research* 52.8 (2024), pp. 4234–4256.

[6] Hiroaki Ohishi et al. “Transcription-coupled changes in genomic region proximities during transcriptional bursting”. In: *Science Advances* 10.49 (2024), eadn0020.

[7] Sujay Pal and Debabrata Biswas. “Promoter-proximal regulation of gene transcription: Key factors involved and emerging role of general transcription factors in assisting productive elongation”. In: *Gene* 878 (2023), p. 147571.

[8] Ali Raza et al. “iPro-TCN: prediction of DNA promoters recognition and their strength using temporal convolutional network”. In: *IEEE Access* 11 (2023), pp. 66113–66121.

[9] Yazi Li et al. “msBERT-Promoter: a multi-scale ensemble predictor based on BERT pre-trained model for the two-stage prediction of DNA promoters and their strengths”. In: *BMC biology* 22.1 (2024), p. 126.

[10] Vikram Agarwal et al. “Massively parallel characterization of transcriptional regulatory elements”. In: *Nature* 639.8054 (2025), pp. 411–420.

[11] Kseniia Dudnyk et al. “Sequence basis of transcription initiation in the human genome”. In: *Science* 384.6694 (2024), eadj0116.

[12] Pengcheng Zhang et al. “Deep flanking sequence engineering for efficient promoter design using DeepSEED”. In: *Nature communications* 14.1 (2023), p. 6309.

[13] George Hunt et al. “Tissue-specific RNA Polymerase II promoter-proximal pause release and burst kinetics in a Drosophila embryonic patterning network”. In: *Genome Biology* 25.1 (2024), p. 2.

[14] M Arnold and KR Stengel. *Emerging insights into enhancer biology and function. Transcription 14: 68–87*. 2023.

[15] Ying Huang et al. “HSFA1a modulates plant heat stress responses and alters the 3D chromatin organization of enhancer-promoter interactions”. In: *Nature communications* 14.1 (2023), p. 469.

[16] Ilias Georgakopoulos-Soares et al. “Transcription factor binding site orientation and order are major drivers of gene regulatory activity”. In: *Nature communications* 14.1 (2023), p. 2333.

[17] Michael JG Milevskiy et al. “Three-dimensional genome architecture coordinates key regulators of lineage specification in mammary epithelial cells”. In: *Cell Genomics* 3.11 (2023).

[18] Isabel Esain-Garcia et al. “G-quadruplex DNA structure is a positive regulator of MYC transcription”. In: *Proceedings of the National Academy of Sciences* 121.7 (2024), e2320240121.

[19] Omkar Chandra et al. “Patterns of transcription factor binding and epigenome at promoters allow interpretable predictability of multiple functions of non-coding and coding genes”. In: *Computational and Structural Biotechnology Journal* 21 (2023), pp. 3590–3603.

[20] Simon Pavlu et al. “Core promoterome of barley embryo”. In: *Computational and Structural Biotechnology Journal* 23 (2024), pp. 264–277.

[21] Sarvesh Nikumbh and Boris Lenhard. “Identifying promoter sequence architectures via a chunking-based algorithm using non-negative matrix factorisation”. In: *PLOS Computational Biology* 19.11 (2023), e1011491.

[22] Shujun He et al. “Nucleic transformer: Classifying dna sequences with self-attention and convolutions”. In: *ACS Synthetic Biology* 12.11 (2023), pp. 3205–3214.

[23] Dmitry Penzar et al. “LegNet: a best-in-class deep learning model for short DNA regulatory regions”. In: *Bioinformatics* 39.8 (2023), btad457.

[24] Abdul Muntakim Rafi et al. “A community effort to optimize sequence-based deep learning models of gene regulation”. In: *Nature biotechnology* (2024), pp. 1–11.

[25] Javier Mendoza-Revilla et al. “A foundational large language model for edible plant genomes”. In: *Communications Biology* 7.1 (2024), p. 835.

[26] Leonardo Ledesma-Dominguez et al. “DeepReg: a deep learning hybrid model for predicting transcription factors in eukaryotic and prokaryotic genomes”. In: *Scientific Reports* 14.1 (2024), p. 9155.

[27] Nguyen Quoc Khanh Le et al. “BERT-Promoter: An improved sequence-based predictor of DNA promoter using BERT pre-trained model and SHAP feature selection”. In: *Computational Biology and Chemistry* 99 (2022), p. 107732.

[28] Tianjiao Zhang et al. “GATv2EPI: Predicting Enhancer–Promoter Interactions with a Dynamic Graph Attention Network”. In: *Genes* 15.12 (2024), p. 1511.

[29] Fatma S Ahmed, Saleh Aly, and Xiangrong Liu. “Epi-trans: an effective transformer-based deep learning model for enhancer promoter interaction prediction”. In: *BMC bioinformatics* 25.1 (2024), p. 216.

[30] Il-Youp Kwak et al. “Proformer: a hybrid macaron transformer model predicts expression values from promoter sequences”. In: *BMC bioinformatics* 25.1 (2024), p. 81.

[31] Bowen Liu et al. “TF-EPI: An interpretable enhancer-promoter interaction detection method based on transformer”. In: *Frontiers in Genetics* 15 (2024), p. 1444459.

[32] Binchao Peng, Guicong Sun, and Yongxian Fan. “iProL: identifying DNA promoters from sequence information based on Longformer pre-trained model”. In: *BMC bioinformatics* 25.1 (2024), p. 224.

[33] Xiaokang Huang et al. “scGRN: a comprehensive single-cell gene regulatory network platform of human and mouse”. In: *Nucleic Acids Research* (2024). Accessed: 2025-09-10.

[34] Wentao Cui et al. “Refining computational inference of gene regulatory networks: integrating knockout data within a multi-task framework”. In: *Briefings in Bioinformatics* 25.5 (2024), bbae361.

[35] N E Kazwini and Guido Sanguinetti. “SHARE-Topic: Bayesian interpretable modeling of single-cell multi-omic data”. In: *Genome Biology* (2024).

[36] Melissa Sanabria et al. “DNA language model GROVER learns sequence context in the human genome”. In: *Nature Machine Intelligence* 6.8 (2024), pp. 911–923.

[37] Jens Füllgrabe et al. “Simultaneous sequencing of genetic and epigenetic bases in DNA”. In: *Nature Biotechnology* 41.10 (2023), pp. 1457–1464.

[38] Asmaa H Hassan, Morad M Mokhtar, and Achraf El Allali. “Transposable elements: multifunctional players in the plant genome”. In: *Frontiers in Plant Science* 14 (2024), p. 1330127.

[39] Andrei P Kozlov. “Diagrams Describing the Evolution of Gene Expression, the Emergence of Novel Cell Types During Evolution, and Evo-devo.” In: *Gene Expression (1052-2166)* 22.3 (2023).

[40] Alexey A Malygin. “Many Faces of Next-Generation Sequencing in Gene Expression Studies”. In: *International Journal of Molecular Sciences* 24.4 (2023), p. 4075.

[41] Eric T Wang et al. “What repeat expansion disorders can teach us about the central Dogma”. In: *Molecular Cell* 83.3 (2023), pp. 324–329.

[42] John S Mattick. “A Kuhnian revolution in molecular biology: Most genes in complex organisms express regulatory RNAs”. In: *BioEssays* 45.9 (2023), p. 2300080.

[43] Jonathan E Henninger and Richard A Young. “An RNA-centric view of transcription and genome organization”. In: *Molecular Cell* 84.19 (2024), pp. 3627–3643.

[44] Stephen R Archuleta, James A Goodrich, and Jennifer F Kugel. “Mechanisms and functions of the RNA polymerase II general transcription machinery during the transcription cycle”. In: *Biomolecules* 14.2 (2024), p. 176.

[45] James C Kuldell and Craig D Kaplan. “RNA Polymerase II Activity Control of Gene Expression and Involvement in Disease”. In: *Journal of molecular biology* 437.1 (2025), p. 168770.

[46] Xizi Chen et al. *Structural visualization of transcription initiation in action*. 2023.

[47] Jun Wang and Vikram Agarwal. “How DNA encodes the start of transcription”. In: *Science* 384.6694 (2024), pp. 382–383.

[48] Zhaojia Chen et al. “From tradition to innovation: conventional and deep learning frameworks in genome annotation”. In: *Briefings in Bioinformatics* 25.3 (2024), bbae138.

[49] Q Sun et al. *Large-scale detection of telomeric motif sequences in genomic data using TelFinder. Microbiol Spectr 11: e03928-22.* 2023.

[50] Hannah K Neikes et al. "Quantification of absolute transcription factor binding affinities in the native chromatin context using BANC-seq". In: *Nature Biotechnology* 41.12 (2023), pp. 1801–1809.

[51] Atreyi Chakraborty, Sumant Chopde, and Mallur Srivatsan Madhusudhan. "Motif distribution in genomes gives insights into gene clustering and co-regulation". In: *Nucleic Acids Research* 53.1 (2025), gkae1178.

[52] Guilherme Miura Lavezzo et al. "Position Weight Matrix or Acyclic Probabilistic Finite Automaton: Which model to use? A decision rule inferred for the prediction of transcription factor binding sites". In: *Genetics and Molecular Biology* 46.4 (2023), e20230048.

[53] Eugene V Korotkov et al. "Mathematical Algorithm for Identification of Eukaryotic Promoter Sequences". In: *Symmetry* 13.6 (2021), p. 917.

[54] Claus Vogl et al. "Inference of genomic landscapes using ordered Hidden Markov Models with emission densities (oHMMed)". In: *BMC bioinformatics* 25.1 (2024), p. 151.

[55] Meng Zhang et al. "Critical assessment of computational tools for prokaryotic and eukaryotic promoter prediction". In: *Briefings in bioinformatics* 23.2 (2022), bbab551.

[56] Mauro de Medeiros Oliveira et al. "TSSFinder fast and accurate ab initio prediction of the core promoter in eukaryotic genomes". In: *Briefings in Bioinformatics* 22.6 (2021), bbab198.

[57] Gustavo Sganzerla Martinez et al. "Machine learning and statistics shape a novel path in archaeal promoter annotation". In: *BMC bioinformatics* 23.1 (2022), p. 171.

[58] Jakub Horvath et al. "Detection and classification of long terminal repeat sequences in plant LTR-retrotransposons and their analysis using explainable machine learning". In: *BioData mining* 17.1 (2024), p. 57.

[59] Mullapudi Venkata Sai Samartha et al. "AI-driven estimation of O6 methylguanine-DNA-methyltransferase (MGMT) promoter methylation in glioblastoma patients: A systematic review with bias analysis". In: *Journal of Cancer Research and Clinical Oncology* 150.2 (2024), p. 57.

[60] Dajo Smet, Helder Opdebeeck, and Klaas Vandepoele. "Predicting transcriptional responses to heat and drought stress from genomic features using a machine learning approach in rice". In: *Frontiers in Plant Science* 14 (2023), p. 1212073.

[61] Niveda Gaspar, Vimal Shanmuganathan, Ahmed Alkhayyat, et al. "Classifying promoters by interpreting the hidden information of DNA sequences for disease prediction in clinical laboratories using Gaussian decision boundary estimation". In: *Intelligent Decision Technologies* 18.1 (2024), pp. 613–631.

[62] Subhojit Paul et al. "MLDSPP: Bacterial promoter prediction tool using DNA structural properties with machine learning and explainable AI". In: *Journal of Chemical Information and Modeling* 64.7 (2024), pp. 2705–2719.

[63] Fabio M Doniselli et al. "Development of a radiomic model for MGMT promoter methylation detection in glioblastoma using conventional MRI". In: *International Journal of Molecular Sciences* 25.1 (2023), p. 138.

[64] Maryam Hosseini, Maryam Lotfi-Shahreza, and Parvaneh Nikpour. "Integrative analysis of DNA methylation and gene expression through machine learning identifies stomach cancer diagnostic and prognostic biomarkers". In: *Journal of Cellular and Molecular Medicine* 27.5 (2023), pp. 714–726.

[65] Subhojit Paul et al. "MLDSPP: Bacterial promoter prediction tool using DNA structural properties with machine learning and explainable AI". In: *Journal of Chemical Information and Modeling* 64.7 (2024), pp. 2705–2719.

[66] Mei Zhao et al. "Precise prediction of promoter strength based on a de novo synthetic promoter library coupled with machine learning". In: *ACS Synthetic Biology* 11.1 (2021), pp. 92–102.

[67] The-Anh Tran, Dinh-Minh Pham, Yu-Yen Ou, et al. "An extensive examination of discovering 5-Methylcytosine Sites in Genome-Wide DNA Promoters using machine learning based approaches". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 19.1 (2021), pp. 87–94.

[68] Nikita Bhandari et al. "Comparison of machine learning and deep learning techniques in promoter prediction across diverse species". In: *PeerJ Computer Science* 7 (2021), e365.

[69] Aman Agarwal and Li Chen. "DeepPHiC: predicting promoter-centered chromatin interactions using a novel deep learning approach". In: *Bioinformatics* 39.1 (2023), btac801.

[70] Xinglong Wang et al. “Accelerating promoter identification and design by deep learning”. In: *Trends in Biotechnology* (2025).

[71] Bindi M Nagda, Ryan T White, et al. “promSEMBLE: Hard Pattern Mining and Ensemble Learning for Detecting DNA Promoter Sequences”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 21.1 (2023), pp. 208–214.

[72] Xuan Xiao et al. “iPSI (2L)-EDL: a two-layer predictor for identifying promoters and their types based on ensemble deep learning”. In: *Current Bioinformatics* 19.4 (2024), pp. 327–340.

[73] Elena Pianfetti et al. “MiREx: mRNA levels prediction from gene sequence and miRNA target knowledge”. In: *Bmc Bioinformatics* 24.1 (2023), p. 443.

[74] Aqsa Amjad et al. “A novel deep learning identifier for promoters and their strength using heterogeneous features”. In: *Methods* 230 (2024), pp. 119–128.

[75] Samin Rahman Khan et al. “DeepBend: an interpretable model of DNA bendability”. In: *Iscience* 26.2 (2023).

[76] Safwan Mahmood Al-Selwi et al. “LSTM inefficiency in long-term dependencies regression problems”. In: *Journal of Advanced Research in Applied Sciences and Engineering Technology* 30.3 (2023), pp. 16–31.

[77] Aqsa Amjad et al. “A novel deep learning identifier for promoters and their strength using heterogeneous features”. In: *Methods* 230 (2024), pp. 119–128.

[78] Juexiao Zhou et al. “DeeReCT-TSS: A novel meta-learning-based method annotates TSS in multiple cell types based on DNA sequences and RNA-seq data”. In: *bioRxiv* (2021), pp. 2021–07.

[79] Aparna R Rajpurkar et al. “Deep learning connects DNA traces to transcription to reveal predictive features beyond enhancer–promoter contact”. In: *Nature communications* 12.1 (2021), p. 3423.

[80] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[81] Yanrong Ji et al. “DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome”. In: *Bioinformatics* 37.15 (2021), pp. 2112–2120.

[82] Y Zhang et al. “msBERT-Promoter: a multi-scale ensemble predictor based on BERT pre-trained model for the two-stage prediction of DNA promoters and their strengths”. In: *BMC Biology* 22.1 (2024), pp. 1–14.

[83] Bowen Liu et al. "TF-EPI: an interpretable enhancer-promoter interaction detection method based on Transformer". In: *Frontiers in Genetics* 15 (2024), p. 1444459.

[84] Dung Hoang Anh Mai, Linh Thanh Nguyen, and Eun Yeol Lee. "TSSNote-CyaPromBERT: development of an integrated platform for highly accurate promoter prediction and visualization of Synechococcus sp. and Synechocystis sp. through a state-of-the-art natural language processing model BERT". In: *Frontiers in Genetics* 13 (2022), p. 1067562.

[85] Zhongshen Li et al. "PLPMpro: Enhancing promoter sequence prediction with prompt-learning based pre-trained language model". In: *Computers in Biology and Medicine* 164 (2023), p. 107260.

[86] Dohoon Lee, Jeewon Yang, and Sun Kim. "Learning the histone codes with large genomic windows and three-dimensional chromatin interactions using transformer". In: *Nature Communications* 13.1 (2022), p. 6678.

[87] Giulia Muzio, Leslie O'Bray, and Karsten Borgwardt. "Biological network analysis with deep learning". In: *Briefings in bioinformatics* 22.2 (2021), pp. 1515–1530.

[88] Petar Veličković et al. "Graph Attention Networks". In: *CoRR* abs/1710.10903 (2017).

[89] Hakan T Otal et al. "Analysis of gene regulatory networks from gene expression using graph neural networks". In: *Digital Healthcare, Digital Transformation and Citizen Empowerment in Asia-Pacific and Europe for a Healthier Society*. Elsevier, 2025, pp. 249–270.

[90] Sudipto Baul et al. "omicsGAT: Graph attention network for cancer subtype analyses". In: *International Journal of Molecular Sciences* 23.18 (2022), p. 10220.

[91] Alireza Karbalayghareh, Merve Sahin, and Christina S Leslie. "Chromatin interaction–aware gene regulatory modeling with graph attention networks". In: *Genome Research* 32.5 (2022), pp. 930–944.

[92] Alireza Karbalayghareh, Merve Sahin, and Christina S Leslie. "Chromatin interaction–aware gene regulatory modeling with graph attention networks". In: *Genome Research* 32.5 (2022), pp. 930–944.

[93] Zhi-Hua Du et al. "GraphTGI: an attention-based graph embedding model for predicting TF-target gene interactions". In: *Briefings in Bioinformatics* 23.3 (2022), bbac148.

[94] Ke Ma et al. “PPRTGI: A Personalized PageRank Graph Neural Network for TF-Target Gene Interaction Detection”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2024).

[95] Guo Mao et al. “Predicting gene regulatory links from single-cell RNA-seq data using graph neural networks”. In: *Briefings in Bioinformatics* 24.6 (2023), bbad414.

[96] Shuo Li et al. “GMFGRN: a matrix factorization and graph neural network approach for gene regulatory network inference”. In: *Briefings in Bioinformatics* 25.2 (2024), bbad529.

[97] Samet Tenekeci and Selma Tekir. “Identifying promoter and enhancer sequences by graph convolutional networks”. In: *Computational Biology and Chemistry* 110 (2024), p. 108040.

[98] Lin Yuan et al. “scMGATGRN: a multiview graph attention network–based method for inferring gene regulatory networks from single-cell transcriptomic data”. In: *Briefings in bioinformatics* 25.6 (2024), bbae526.

[99] Sören Richard Stahlschmidt, Benjamin Ulfenborg, and Jane Synnergren. “Multimodal deep learning for biomedical data fusion: a review”. In: *Briefings in bioinformatics* 23.2 (2022), bbab569.

[100] Ying Wang et al. “A successful hybrid deep learning model aiming at promoter identification”. In: *BMC bioinformatics* 23.Suppl 1 (2022), p. 206.

[101] Amandeep Kaur, Ajay Pal Singh Chauhan, and Ashwani Kumar Aggarwal. “Prediction of enhancers in DNA sequence data using a hybrid CNN-DLSTM model”. In: *IEEE/ACM transactions on computational biology and bioinformatics* 20.2 (2022), pp. 1327–1336.

[102] Luís A Vale-Silva and Karl Rohr. “Long-term cancer survival prediction using multimodal deep learning”. In: *Scientific Reports* 11.1 (2021), p. 13505.

[103] Ashley Mae Conard, Alan DenAdel, and Lorin Crawford. “A spectrum of explainable and interpretable machine learning approaches for genomic studies”. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 15.5 (2023), e1617.

[104] Aman Agarwal and Li Chen. “DeepPHiC: predicting promoter-centered chromatin interactions using a novel deep learning approach”. In: *Bioinformatics* 39.1 (2023), btac801.

[105] Erik Hartman et al. “Interpreting biologically informed neural networks for enhanced proteomic biomarker discovery and pathway analysis”. In: *Nature Communications* 14.1 (2023), p. 5359.

[106] Glen McGee et al. “Incorporating biological knowledge in analyses of environmental mixtures and health”. In: *Statistics in Medicine* 42.17 (2023), pp. 3016–3031.

[107] Qiao Kang et al. “Causal prior-embedded physics-informed neural networks and a case study on metformin transport in porous media”. In: *Water Research* 261 (2024), p. 121985.

[108] Liang Dong et al. “Leveraging saliency priors and explanations for enhanced consistent interpretability”. In: *Expert Systems with Applications* 249 (2024), p. 123518.

[109] Zhaojia Chen et al. “From tradition to innovation: conventional and deep learning frameworks in genome annotation”. In: *Briefings in Bioinformatics* 25.3 (2024), bbae138.

[110] Andrés R Mansisidor and Viviana I Risca. “Chromatin accessibility: methods, mechanisms, and biological insights”. In: *Nucleus* 13.1 (2022), pp. 238–278.

[111] Sandra Steyaert et al. “Multimodal data fusion for cancer biomarker discovery with deep learning”. In: *Nature machine intelligence* 5.4 (2023), pp. 351–362.

[112] Mengdi Liu et al. “G2PDiffusion: Genotype-to-Phenotype Prediction with Diffusion Models”. In: *arXiv preprint arXiv:2502.04684* (2025).

[113] NQK Le et al. “BERT-Promoter: An improved sequence-based predictor of DNA promoter using BERT pre-trained model and SHAP feature selection”. In: *Computational Biology and Chemistry* 99 (2022), p. 107732.

[114] Raul I Perez Martell et al. “Supervised promoter recognition: a benchmark framework”. In: *BMC bioinformatics* 23.1 (2022), p. 118.

[115] Ramzan Umarov et al. “ReFeaFi: genome-wide prediction of regulatory elements driving transcription initiation”. In: *PLoS Computational Biology* 17.9 (2021), e1009376.

[116] Travis L LaFleur, Ayaan Hossain, and Howard M Salis. “Automated model-predictive design of synthetic promoters to control transcriptional profiles in bacteria”. In: *Nature communications* 13.1 (2022), p. 5159.

[117] Zhi-min Zhang et al. “iPromoter-CLA: identifying promoters and their strength by deep capsule networks with bidirectional long short-term memory”. In: *Computer Methods and Programs in Biomedicine* 226 (2022), p. 107087.

[118] Marinka Zitnik et al. “Current and future directions in network biology”. In: *Bioinformatics advances* 4.1 (2024), vbae099.

[119] Michael Tsabar et al. “Connecting timescales in biology: Can early dynamical measurements predict long-term outcomes?” In: *Trends in cancer* 7.4 (2021), pp. 301–308.

[120] David A Garcia et al. “Power-law behavior of transcription factor dynamics at the single-molecule level implies a continuum affinity model”. In: *Nucleic acids research* 49.12 (2021), pp. 6605–6620.

[121] Zihan Zhang, Lei Shi, and Ding-Xuan Zhou. “Classification with deep neural networks and logistic loss”. In: *Journal of Machine Learning Research* 25.125 (2024), pp. 1–117.

[122] Qiufu Li, Huibin Xiao, and Linlin Shen. “BCE vs. CE in Deep Feature Learning”. In: *arXiv preprint arXiv:2505.05813* (2025).

[123] Kevin L Howe et al. “Ensembl 2021”. In: *Nucleic acids research* 49.D1 (2021), pp. D884–D891.

[124] MA Zabidi and CD Arnold. “The core promoter is a regulatory hub for developmental gene expression”. In: *Frontiers in Cell and Developmental Biology* 9 (2021), p. 666508.

[125] M Pham and S Roy. “Enhlink infers distal and context-specific enhancer–promoter linkages”. In: *Genome Biology* 25.1 (2024), pp. 1–20.

[126] A Sandelin and P Carninci. “Highly diversified core promoters in the human genome and their effects on gene expression and disease predisposition”. In: *BMC Genomics* 21 (2020), p. 456.

[127] Peter JA Cock et al. “Biopython: freely available Python tools for computational molecular biology and bioinformatics”. In: *Bioinformatics* 25.11 (2009), p. 1422.

[128] Marouen Ben Guebila et al. “GRAND: a database of gene regulatory network models across human conditions”. In: *Nucleic Acids Research* 50.D1 (2022), pp. D610–D621.

[129] Kimberly Glass et al. “Passing messages between biological networks to refine predicted interactions”. In: *PloS one* 8.5 (2013), e64832.

[130] Marieke Lydia Kuijjer et al. “Estimating sample-specific regulatory networks”. In: *Iscience* 14 (2019), pp. 226–240.

[131] Akkinepally Vanaja and Venkata Rajesh Yella. “Delineation of the DNA structural features of eukaryotic core promoter classes”. In: *ACS omega* 7.7 (2022), pp. 5657–5669.

[132] Mengyuan Wang, Qian Li, and Lingbo Liu. "Factors and methods for the detection of gene expression regulation". In: *Biomolecules* 13.2 (2023), p. 304.

[133] Mohsen Hajheidari and Shao-shan Carol Huang. "Elucidating the biology of transcription factor–DNA interaction for accurate identification of cis-regulatory elements". In: *Current opinion in plant biology* 68 (2022), p. 102232.

[134] Rabia Mishal and Juan Pedro Luna-Arias. "Role of the TATA-box binding protein (TBP) and associated family members in transcription regulation". In: *Gene* 833 (2022), p. 146581.

[135] Manuel Tognon, Rosalba Giugno, and Luca Pinello. "A survey on algorithms to characterize transcription factor binding sites". In: *Briefings in Bioinformatics* 24.3 (2023), bbad156.

[136] Jan Deneweth, Yves Van de Peer, and Vanessa Vermeirssen. "Nearby transposable elements impact plant stress gene regulatory networks: a meta-analysis in A. thaliana and S. lycopersicum". In: *BMC genomics* 23.1 (2022), p. 18.

[137] Yuanzhong Pan, Pauline J van der Watt, and Steve A Kay. "E-box binding transcription factors in cancer". In: *Frontiers in oncology* 13 (2023), p. 1223208.

[138] Connor A Horton et al. "Short tandem repeats bind transcription factors to tune eukaryotic gene expression". In: *Science* 381.6664 (2023), eadd1250.

[139] Sebastian Canzler et al. "Prospects and challenges of multi-omics data integration in toxicology". In: *Archives of Toxicology* 94.2 (2020), pp. 371–388.

[140] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (2017).

[141] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[142] Matthias Fey and Jan Eric Lenssen. "Fast graph representation learning with PyTorch Geometric". In: *arXiv preprint arXiv:1903.02428* (2019).

[143] J.O. De Sordi. *Design Science Research Methodology: Theory Development from Artifacts*. Springer International Publishing, 2021. ISBN: 9783030821562.