Developing Advanced Representation Learning Techniques for mRNA Sequence and Structure Modeling

Sepideh Nahali

A Thesis Submitted to the Faculty of Graduate Studies
In Partial Fulfillment of the Requirements
for the Degree of Master of Arts

Graduate Program in Information Systems Technology

York University
Toronto, Ontario

June 2025

Abstract

Recent studies in bioinformatics and genomics focus on analyzing RNA sequences, which are complex due to diverse nucleotide compositions, varying lengths, and multiple isoforms.

Accurately modeling these sequences is essential for predicting mRNA degradation, a key factor in designing effective RNA-based therapies. However, many existing models struggle to capture the intricate relationships between sequence and structure, limiting their predictive power.

We introduce StructmRNA, a BERT-based model using dual-level and conditional masking to embed RNA sequences and structures. This enables accurate prediction of mRNA sequences and structures without explicit structural data, effectively capturing sequence-structure dependencies. Evaluations show StructmRNA outperforms existing models in predicting mRNA degradation and secondary structure.

Experiments with GAN-generated RNA sequences showed no performance improvement. Nonetheless, StructmRNA's consistent convergence over 30 epochs highlights its robustness and accuracy. This work advances RNA representation learning and demonstrates deep learning's potential in RNA-based therapeutic design and bioinformatics.

Keywords: Bioinformatics, mRNA degradation prediction, mRNA sequences, Secondary structures, StructmRNA model, Machine learning, Two-level masking, Conditional masking, Synthetic RNA data, BERT model, Sequence-structure relationship

Acknowledgments

I am deeply grateful to my supervisor, Professor Jimmy Huang, whose exceptional guidance, profound expertise, and unwavering support have been instrumental throughout my thesis journey. Beyond his role as a professor, his kindness and mentorship as a life guide have inspired me and kept me motivated, and I could always count on his proficiency to steer me forward.

I also extend my sincere thanks to Dr. Leila Safari and Alireza Khanteymoori for their invaluable advice and insights, which greatly enriched my research. Their expertise and encouragement were critical to my success.

Finally, I owe heartfelt gratitude to my family, whose love, patience, and constant support sustained me through the challenges of this endeavor. Their belief in me made this thesis possible.

Contents

List of Tables

List of Figures

1 Introduction

mRNA is a type of RNA molecule that is generated through a transcription process from a DNA strand and exits the cell nucleus to initiate protein synthesis.

Recent research in the field of vaccination has introduced mRNA vaccines as the fastest type of vaccines due to their ease of production. Given the availability of sufficient laboratory datasets containing sequences and information related to their degradation, predicting the degradation rate of these molecules using deep learning methods is a logical approach.

Since chemical reactivity and mRNA degradation data are continuous values, they can be predicted as a regression problem. To achieve this, an efficient method for vector embedding of nucleotides within a sequence is essential. Furthermore, there is currently limited information available regarding which specific parts of the mRNA structure are prone to degradation. To date, despite being time-consuming, laboratory studies remain the only reliable method for predicting mRNA degradation.

With sufficient laboratory datasets available, predicting the degradation rate of these molecules using deep learning methods becomes a viable approach. Given that chemical reactivity and mRNA degradation data are continuous values, they can be formulated as a regression problem. For this purpose, utilizing an efficient vector embedding method for the nucleotides in a sequence is inevitable. By embedding mRNA sequences, in addition to mapping sequence data into a lower-dimensional vector space, new features based on the relationships between nucleotides in a sequence can be extracted.

The goal of this research is to propose an effective approach for estimating the probability of mRNA structural degradation. The proposed solution introduces a deep learning-based model for generating distributed representations of mRNAs, offering high-quality representation. This is achieved by incorporating structural information of mRNAs and modifying the embedding process to align with mRNA sequences.

In this chapter, we first explain the concepts related to different types of RNA and their structures. Then, we examine the problem of estimating mRNA structural degradation, along with its complexities and challenges. At the end of this chapter, commonly used embedding methods in biological applications are discussed in detail.

In summary, this dissertation seeks to answer the following questions:

- Can the embedding of RNA sequences be improved by customizing a language model and extracting patterns within sequences and their structures?
- Can a pretrained model, trained on a vast set of mRNA sequences, accurately predict the degradation of various mRNA structures?
- By relying on the encoded information in vectors obtained from a pretrained model, can mRNA degradation parameters be predicted effectively without using their secondary structures?
- Can the RNA training dataset be expanded in a high-quality manner by utilizing DNA sequences?

- How effective are the generated RNA representations for tasks beyond degradation prediction?

Is the use of a graph embedding approach suitable for the secondary structure of mRNA?

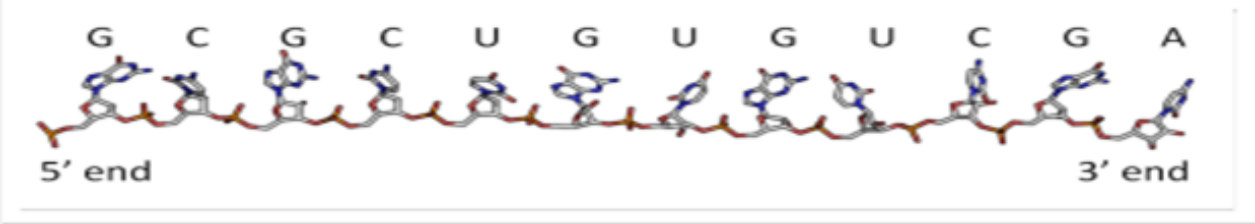


Figure 1: Primary Structure or Nucleotide Sequence of RNA

1.1 Different Approaches to Determining RNA Secondary Structure

The determination of RNA secondary structure is a complex and multifaceted problem that depends on multiple factors, primarily the nucleotide sequence of the RNA molecule as well as the surrounding biochemical environment in which it resides. The sequence dictates the intrinsic propensity of nucleotides to form hydrogen bonds and interact with one another, while the biochemical context, including ionic strength, temperature, and presence of molecular chaperones, modulates the folding process and final structural conformation. Among these environmental influences, one of the most extensively studied and influential factors in RNA secondary structure prediction is the concept of minimum free energy (MFE) [35]. The MFE represents the most thermodynamically favorable configuration that the RNA molecule can adopt, and it is based on the assumption that RNA molecules naturally fold into the structure with the lowest possible free energy.

In addition to minimum free energy considerations, chemical reactivity data has become increasingly recognized as a valuable resource for informing RNA secondary structure models [103]. This data provides experimental insight into the structural flexibility and pairing status of individual nucleotides within the RNA strand by probing their chemical accessibility in various environments. Alongside this, thermodynamic parameters derived from experimental and theoretical studies [53] provide complementary information about the stability of base-pair interactions and loops within the secondary structure. Historically, these approaches, minimum free energy calculations, chemical reactivity probing, and thermodynamic modeling, have often been applied independently in prior research efforts to elucidate RNA secondary structures.

More recently, advancements in machine learning, particularly the development of deep neural networks, have introduced novel methodologies for RNA secondary structure prediction [73] [102]. These data-driven models leverage large datasets of known RNA structures and associated experimental data to learn complex folding patterns, surpassing traditional methods by integrating multiple sources of information and identifying subtle sequence-structure relationships. The integration of such neural network-based approaches marks a significant evolution in the field, offering improved accuracy and robustness in predicting RNA secondary structures under diverse conditions.

1.1.1 Using Minimum Free Energy

Historically, the prediction of RNA secondary structures has been grounded in the use of dynamic programming algorithms that identify the minimum free energy configuration based on established thermodynamic parameters. Tools such as RNAfold [46] have become widely utilized due to their efficiency and relatively high accuracy in predicting local secondary structures by minimizing free energy. RNAfold implements well-established algorithms to scan through all possible foldings of an RNA sequence and select the structure with the lowest overall free energy, which is assumed to represent the most stable and biologically relevant form.

Nevertheless, relying exclusively on the minimum free energy state to predict RNA secondary structure has its limitations. RNA molecules are known to adopt multiple conformations in vivo, often fluctuating among various low-energy states rather than existing in a single static structure. This phenomenon necessitates the consideration of alternative suboptimal structures that are energetically close to the minimum free energy state [98]. To address this, researchers have introduced methods that incorporate the concept of RNA folding pathways, which describe the dynamic process of structure formation influenced by both the RNA sequence and the biochemical environment. These pathways effectively constrain the set of plausible secondary structures, thereby reducing the computational burden associated with enumerating all possible conformations and focusing on biologically meaningful candidates.

1.1.2 Using Chemical Reactivity Information

Chemical reactivity profiling techniques have emerged as a critical component in the study of RNA secondary and tertiary structures. These methods experimentally measure the accessibility and flexibility of nucleotides by reacting RNA with selective chemical probes under various cellular conditions, such as changes in temperature, ion concentration, and molecular crowding [31] [101]. The resulting data provides nucleotide-resolution insights into the structural status of RNA, distinguishing paired bases from unpaired, flexible regions that tend to be chemically more reactive.

To harness the full potential of chemical reactivity data, sophisticated computational frameworks are required to interpret these experimental measurements accurately. Several studies have demonstrated that integrating chemical mapping data can significantly enhance RNA secondary structure predictions, sometimes approaching near-atomic resolution [109]. These approaches have improved the identification of tertiary structure motifs and revealed structural features that are typically undetectable by conventional methods alone.

A notable example of a chemical reactivity technique is the Selective 2'-Hydroxyl Acylation analyzed by Primer Extension (SHAPE) methodology [1], which has become a gold standard for obtaining quantitative nucleotide reactivity profiles [78]. SHAPE data are expressed as vectors of non-negative real numbers, where each value corresponds to the reactivity of a nucleotide at a specific position in the RNA sequence. The fundamental principle behind SHAPE is that unpaired nucleotides, which exist as flexible single strands, exhibit

[1] Selective 2-Hydroxyl Acylation analyzed by Primer Extension

greater chemical reactivity due to their structural accessibility compared to paired nucleotides constrained in stable duplexes. Thus, SHAPE data indirectly reflect the probability that each nucleotide is involved in base-pairing interactions, providing critical constraints for secondary structure modeling.

1.1.3 Using Thermodynamic Information

The integration of thermodynamic principles into RNA secondary structure prediction continues to play a pivotal role in understanding RNA folding mechanisms and functional dynamics [96]. Thermodynamic information encompasses parameters such as enthalpy, entropy, and free energy changes associated with the formation and disruption of base pairs, loops, bulges, and junctions within the RNA structure. These parameters offer insights into the relative stability and feasibility of various conformational states an RNA molecule can adopt under physiological conditions.

Despite the clear advantages of thermodynamic modeling, a significant challenge lies in the sheer complexity of RNA conformational space. RNA molecules can theoretically fold into an astronomical number of possible secondary structures, making exhaustive computational evaluation of all states impractical. This complexity leads to a rugged energy landscape with numerous local minima, complicating the identification of the global minimum free energy structure. Computational methods must therefore balance accuracy with efficiency by narrowing down plausible candidate structures.

Research has shown that thermodynamic data alone is insufficient to fully characterize RNA folding and function. Instead, the most effective predictive frameworks employ a combined approach that integrates thermodynamic metrics with other structural features, such as base-pairing probabilities derived from partition function calculations and the presence of conserved sequence motifs [120]. This holistic strategy enables a more nuanced and accurate representation of RNA behavior, which is essential for developing computational models that reliably predict RNA structure-function relationships across diverse biological contexts.

1.2 mRNA Structure Degradation

Ribosomal RNA (rRNA)[1] and transfer RNA (tRNA)[2] are both examples of highly stable and functionally essential RNA molecules that are distinct from messenger RNA (mRNA) in both structure and role. These stable RNAs are critical for maintaining the efficiency and fidelity of cellular protein synthesis across all domains of life.

In both prokaryotic[3] and eukaryotic[4] cells, rRNA and tRNA genes are encoded within specific regions of the genome. These genes are transcribed into long precursor RNA molecules that undergo post-transcriptional

[1] Ribosomal RNA is a type of non-coding RNA that forms the structural and functional components of ribosomes, the molecular machines responsible for protein synthesis.

[2] Transfer RNA is a small RNA molecule that helps decode a messenger RNA (mRNA) sequence into a protein during the process of translation.

[3] Prokaryotes are unicellular organisms, such as bacteria and archaea, whose cells do not contain a membrane-bound nucleus or other organelles. Their genetic material is typically found in a single circular DNA molecule located in the cytoplasm.

[4] Eukaryotes are organisms ranging from single-celled yeasts to complex multicellular animals and plants. Their cells are characterized by the presence of membrane-bound organelles, including a well-defined nucleus that houses the genetic material.

processing. This includes endonucleolytic cleavage and chemical modification to produce smaller, mature RNA fragments. These mature fragments serve structural and catalytic functions within ribosomes or act as adapters during translation, respectively.

Although mRNA is also a product of transcription, it differs fundamentally from rRNA and tRNA because it contains coding sequences that are translated into proteins. In contrast, rRNA and tRNA do not carry any protein-coding information themselves. However, they are indispensable for interpreting and executing the genetic code stored in mRNA.

In eukaryotic cells, the transcription of rRNA genes, their processing, and the subsequent assembly of ribosomal subunits occur within a specialized sub-nuclear structure known as the nucleolus. This compartmentalization helps coordinate ribosome biogenesis in a spatially organized manner. Conversely, in prokaryotic organisms, which lack a defined nucleus and nucleolus, these processes occur directly in the cytoplasm.

Despite their non-coding nature, both tRNA and rRNA are central to the translation machinery. They facilitate the decoding of mRNA into polypeptides and ensure the correct assembly of amino acids into functional proteins. Their roles in protein biosynthesis are illustrated in Figure 2, highlighting the interconnectedness of RNA species in gene expression.

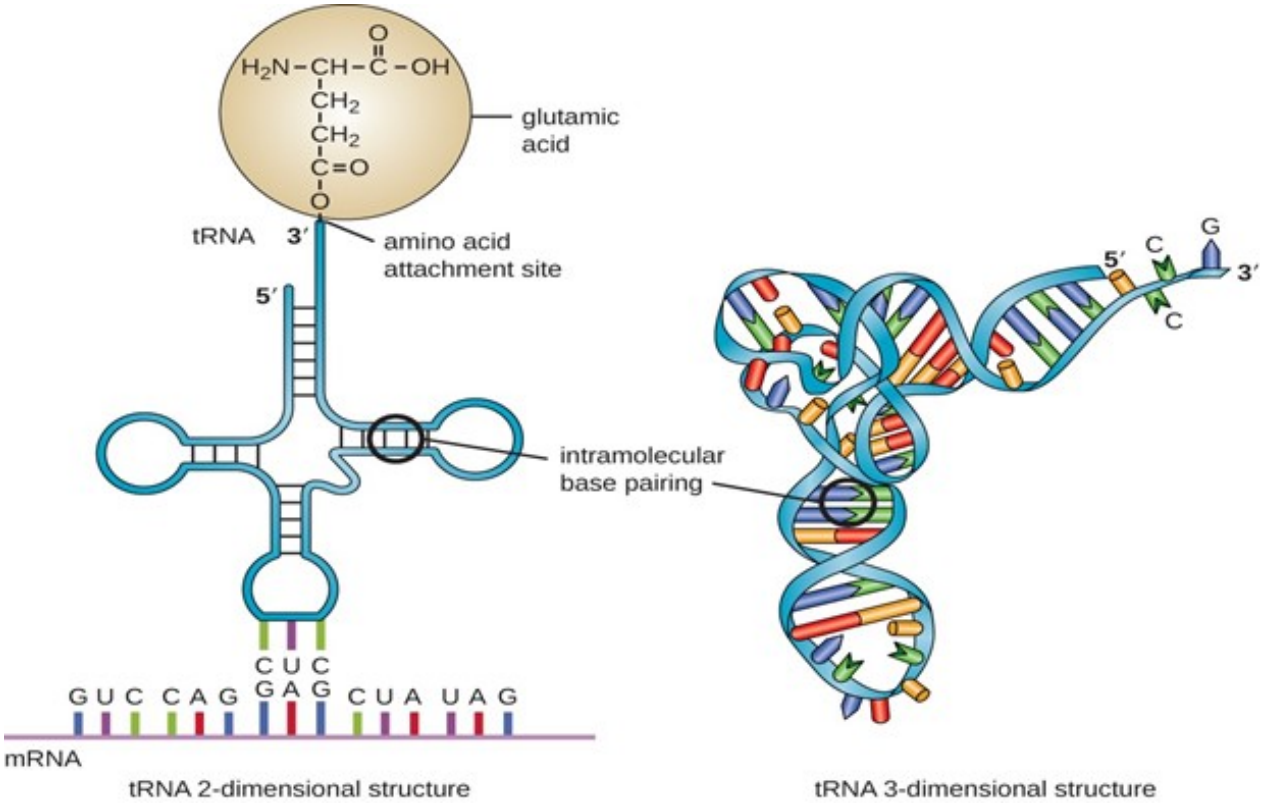


Figure 2: mRNA and tRNA Structure and Protein Formation. (Right) A single-stranded tRNA molecule, which contains nucleotide base pairings within the cell, giving it a distinct three-dimensional shape. (Left) A transcribed copy of a portion of tRNA leading to mRNA formation.

The lifespan (stability) of mRNAs within a cell varies depending on their structure. In bacterial cells, mRNAs

can persist from a few seconds to over an hour [71], whereas in mammalian cells, their lifespan ranges from a few minutes to several days [123]. The primary reason for mRNA degradation and its limited lifespan is to enable cells to rapidly respond to changing protein synthesis demands. On the other hand, greater mRNA stability results in increased protein production. Currently, little information is available about the specific regions of mRNA structures prone to degradation (Figure 3). Several mechanisms contribute to mRNA degradation, some of which are discussed below.

1.2.1 Prokaryotic mRNA Degradation

Generally, the lifespan of mRNA in prokaryotes is shorter than in eukaryotes. Prokaryotes degrade messages using a combination of ribonuclease enzymes, including endonucleases, 3' exonucleases, and 5' exonucleases [30]. In some cases, small mRNA molecules, known as sRNAs, ranging from tens to hundreds of nucleotides, can induce mRNA degradation by base-pairing with their complementary strands, leading to ribonuclease-mediated cleavage. Figure 3 illustrates degradation-prone and stable regions in the mRNA structure.

1.2.2 Eukaryotic mRNA Turnover

In the intricate environment of eukaryotic cells, a delicate equilibrium is maintained between the processes of messenger RNA (mRNA) translation and its subsequent degradation. Actively translating mRNA molecules are safeguarded by ribosomes, which serve as a protective barrier, preventing access by the decapping enzyme, known as DCP2, and the poly(A)-binding protein, which stabilizes the poly(A) tail of the mRNA. This protective mechanism ensures that mRNAs actively engaged in translation remain structurally intact and functional. In contrast, mRNAs that are not actively translated are more susceptible to rapid degradation, as they lack the ribosomal protection that shields them from enzymatic decay processes [89].

1.2.3 AU-rich Element (ARE)-mediated Decay

Certain mRNA molecules contain AU-rich elements, which are specific sequences located within the 3' untranslated region (3' UTR) of the transcript. These elements act as critical regulatory components, contributing to the instability of the mRNA. Cellular proteins bind to these AU-rich sequences, initiating a cascade of events that leads to the removal of the poly(A) tail, a process known as deadenylation. The absence of the poly(A) tail significantly accelerates the degradation of the mRNA, as it becomes more vulnerable to exonucleases and other degradative enzymes [20].

1.2.4 Nonsense-Mediated Decay (NMD)

Eukaryotic cells employ a sophisticated surveillance mechanism known as nonsense-mediated RNA decay (NMD), which functions to detect and eliminate mRNA transcripts containing premature stop codons. These premature termination signals, often arising from mutations or errors in transcription, trigger the rapid degradation of the affected mRNA molecules. By targeting these aberrant transcripts, the NMD system plays a

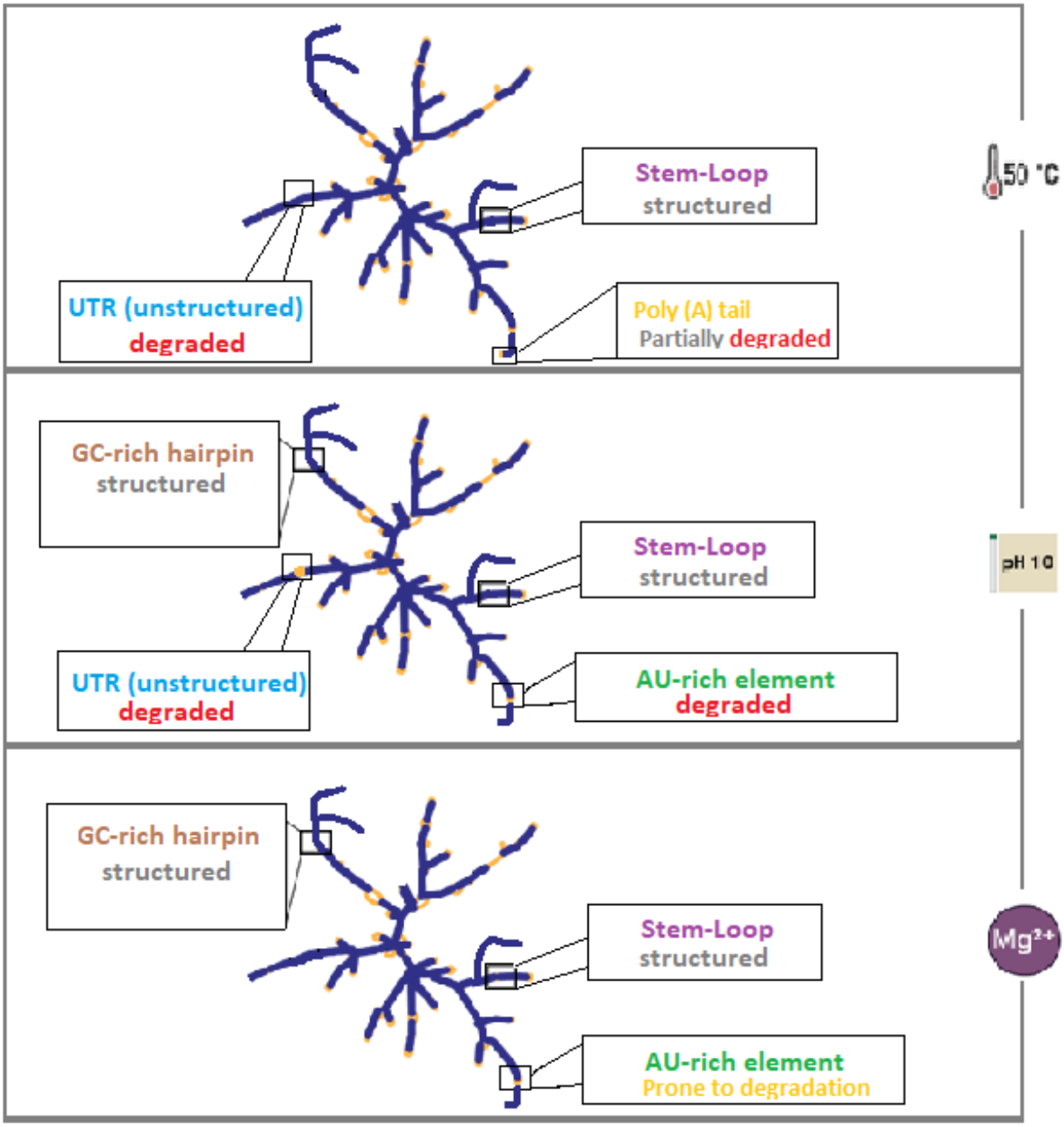


Figure 3: Degradation-prone and stable regions in mRNA structure.

crucial role in maintaining the integrity of gene expression and preventing the production of truncated or dysfunctional proteins [54].

1.2.5 Small Interfering RNA (siRNA)

In animal cells, small interfering RNAs (siRNAs) are critical players in the regulation of gene expression. These small RNA molecules are generated through the processing of double-stranded RNA by the enzyme Dicer. Once produced, siRNAs are incorporated into the RNA-induced silencing complex (RISC), which contains an endonuclease responsible for cleaving mRNA molecules that are complementary to the siRNA sequence. The cleaved mRNA fragments are then targeted for degradation by exonucleases, effectively silencing the expression of the corresponding gene. siRNAs are extensively utilized in laboratory settings to selectively silence gene expression in cell cultures and also serve as a vital component of the innate immune defense system, particularly in combating infections caused by double-stranded RNA viruses [86].

1.2.6 Micro RNA (miRNA)

MicroRNAs (miRNAs) are small, non-coding RNA molecules that play a pivotal role in post-transcriptional gene regulation in metazoan organisms. These miRNAs are complementary to specific regions of messenger RNAs, typically within the 3' UTR. Upon binding to their target mRNA, miRNAs can inhibit translation by interfering with the ribosomal machinery or promote the removal of the poly(A) tail, a process that destabilizes the mRNA and accelerates its degradation. Through these mechanisms, miRNAs exert precise control over gene expression, influencing a wide range of biological processes, including development, differentiation, and cellular responses to environmental stimuli [18].

Other factors influencing mRNA decay include nonstop decay, gene silencing, and certain noncoding RNAs, among others.

1.3 Language Models and Embedding

Word embedding serves as a cornerstone for numerous deep learning applications within the domain of Natural Language Processing (NLP), providing a fundamental framework for processing and understanding textual data [134]. These word embeddings, typically represented as dense vectors, are meticulously trained on vast textual corpora, leveraging the statistical patterns of word co-occurrence to capture meaningful linguistic relationships. The training process is generally unsupervised, meaning it does not rely on manually labeled data, allowing the model to independently learn patterns from the input text [1]. The primary objective of word embedding techniques, which have gained widespread adoption and prominence in the NLP community, is to transform words into a numerical vector format that encapsulates the semantic and syntactic properties of the text. By doing so, this approach ensures that words with similar meanings or contextual usage are positioned close to one another within the multidimensional vector space, thereby reflecting their linguistic proximity. Consequently, these embedding methods have become a highly effective and widely regarded alternative to traditional feature engineering techniques, which often require labor-intensive manual design [2]. In modern NLP tasks, word embeddings provide a robust, automated means of representing

[1] Unsupervised
[2] Feature Engineering

textual information, significantly enhancing the performance of models in capturing the intricate nuances of language.

In general, word embedding methods can be broadly classified into two primary categories: classical and contextual, each with distinct characteristics and approaches. Classical embedding techniques typically rely on traditional statistical methodologies and are inherently static in nature. This static quality implies that the word vectors generated by these methods remain fixed, meaning that a given word is assigned a single, unchanging embedding vector regardless of the varying meanings or contexts in which it may appear across different texts [81]. As a result, these methods do not account for polysemy, where a word can have multiple meanings depending on its usage. In contrast, more advanced and recently developed approaches, referred to as contextualized word embedding techniques, effectively address this limitation by incorporating sophisticated language models that dynamically generate word representations. These contextual methods produce embeddings that adapt to the specific context in which a word is used, thereby capturing nuanced meanings more accurately. The following sections provide a detailed explanation of the most significant and widely used methods within each of these two categories, highlighting their mechanisms, strengths, and applications.

1.4 Classical Word Embedding Methods

1.4.1 Efficient Estimation of Word Representation in Vector Space (Word2Vec)

When the Word2Vec software package became publicly available, a new era in Natural Language Processing began. This embedding method is similar to an autoencoder [1] that encodes each word into a vector and trains it along with neighboring words [82]. This method is carried out in two different ways. In the first method, known as skip-gram, the target word is predicted using the neighboring words. In the second method, the neighboring words are predicted using one target word. This method is called Continuous Bag of Words (CBOW) [2] (see Figure 4). Since the second method produces more accurate results on large datasets, it has gained more popularity.

1.4.2 Global Vectors for Word Representation (GloVe)

GloVe is a count-based learning algorithm for word embedding [92] [15]. In this method, vectors are calculated using a dimensionality reduction technique applied to a matrix called the co-occurrence matrix [3]. The algorithm starts by creating a matrix of co-occurrence information for words in a text and then performs matrix factorization [15]. In this process, an initially high-dimensional matrix representing the frequency of word co-occurrence is created, and then matrix factorization is applied to reduce its dimensionality. As a result, a lower-dimensional matrix is produced, where each row represents the vectorized representation of a word [9].

[1] Autoencoder
[2] Continuous Bag of Words
[3] Co-Occurrence Matrix

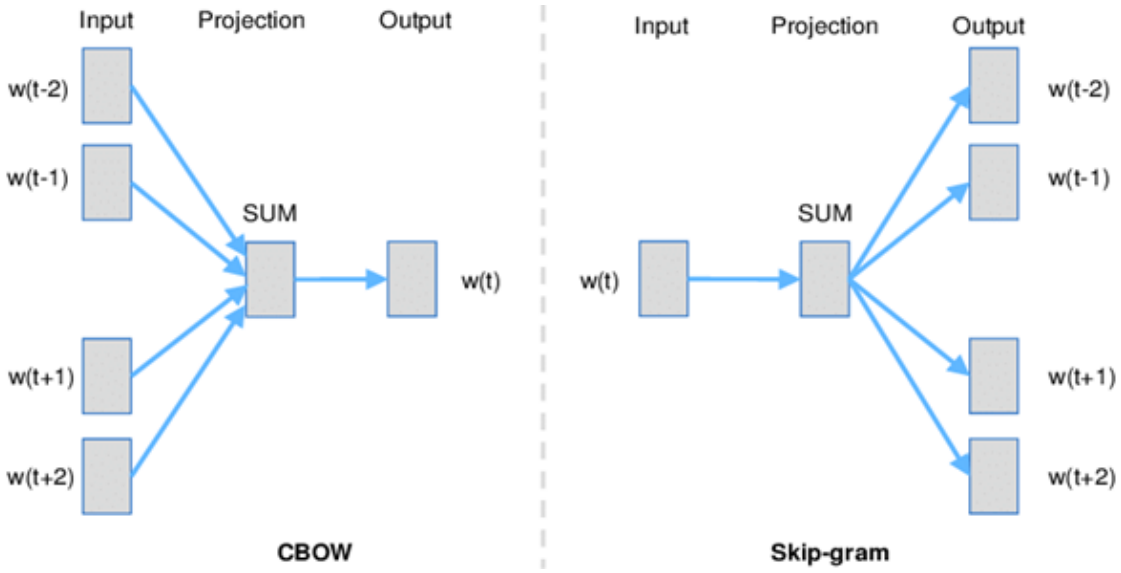


Figure 4: The architecture of the Word2Vec model.

1.4.3 Word Vector Enrichment with Subword Information (fastText)

A fundamental issue with embedding methods such as GloVe and Word2Vec is that they only define words found in the training texts, and they lack the capability to efficiently embed words outside of those texts. The fastText method embeds words at the subword level, using a ”sub-word” approach based on the skip-gram model, where each word is a combination of several characters (n-grams) [17] [16]. Thus, for each character combination or n-gram, a vector is embedded, and the word vector is formed by summing these individual vectors. This method allows the model to easily compute the vectors for words that were not present in the training set [97].

1.5 Text Embedding

The classical methods previously mentioned create the same embedding vector for a word across different texts [59]. Therefore, the vectors produced by pretrained models using these methods will represent homophones or words with the same spelling but different meanings as a combined vector that incorporates all their meanings. This causes the resulting vector to be relatively meaningless and not properly represent any of the word’s meanings. This fundamental weakness is typical of these methods. For example, classic models will generate the same vector for both occurrences of the word ”lion” in the sentences ”The lion’s habitat is in the forest.” and ”The water faucet needs repair.”

In contrast, text embedding methods address this weakness by considering the context and the order of the sentences throughout the entire text using a language model. In other words, these methods leverage the deep understanding of the language model to distinguish between different meanings of a word, and for each meaning, they create different vectors [33]. This concept also applies to biological sequences. For example, distinguishing the vector for nucleotide A in codon AUU from nucleotide A in codon AUC creates a more nuanced, practical understanding of nucleotide sequences for solving RNA sequence-related problems.

1.5.1 Language Models

Language models calculate a probabilistic distribution over a sequence of words, considering consecutive words. LSTMs [1] have become a popular neural network architecture for learning these probabilities.

The language model works by feeding a word sequence one word at a time as input into an LSTM layer. The previous word, along with the internal state of the LSTM, is used to predict the most likely next word. This language model calculates a probabilistic distribution over the likelihood of the next words in the text (it is also possible to go below the word level and create a language model at the character level).

It is important to note that embedding methods like Word2Vec should not be compared with language models because, in a language model, the word order is crucial, while models like Word2Vec do not care about the word order, as all they do during training is predict neighboring words within a window without considering their positions. In fact, Word2Vec models and language models are almost complementary. A language model can benefit from the output of a Word2Vec model. In this case, the language model may perform better than one that randomly converts words into vectors before training.

1.5.2 Deep Text Embeddings for Words (ELMo) [2]

After Word2Vec, the ELMo method marks a significant advancement in word embedding. The core idea of ELMo is to embed information from the entire text into the word vectors [93]. ELMo's solution for embedding homonymous words, which are written the same but have different meanings, is to train a language model by reading sentences both from left to right and vice versa. Essentially, there are two parallel language models, one that learns to predict the next word based on past words, and another that learns to predict past words based on future ones.

As mentioned, language models generally use an LSTM. However, in this model, instead of using a single LSTM layer, LSTMs are stacked. This means that a single-layer LSTM takes the word sequence as input, and a multi-layer LSTM takes the output from the previous LSTM layer as input. Thus, each layer of the LSTM in this language model learns different language features. Therefore, this language model is also referred to as a bidirectional language model (see Figure 5).

Therefore, in this embedding approach, with a pre-trained language model at the top of the network and a supervised neural network architecture at the bottom, the final model, which is a linear combination of vectors from different layers, can be adjusted to solve a natural language processing problem. ELMo can practically replace existing embedding methods; however, the authors of this paper recommend that before incorporating ELMo vectors into a model for a specific problem, these vectors should be combined with context-independent word vectors such as GloVe or fastText [94].

[1] Long-Short Term Memory
[2] Embeddings from Language Models

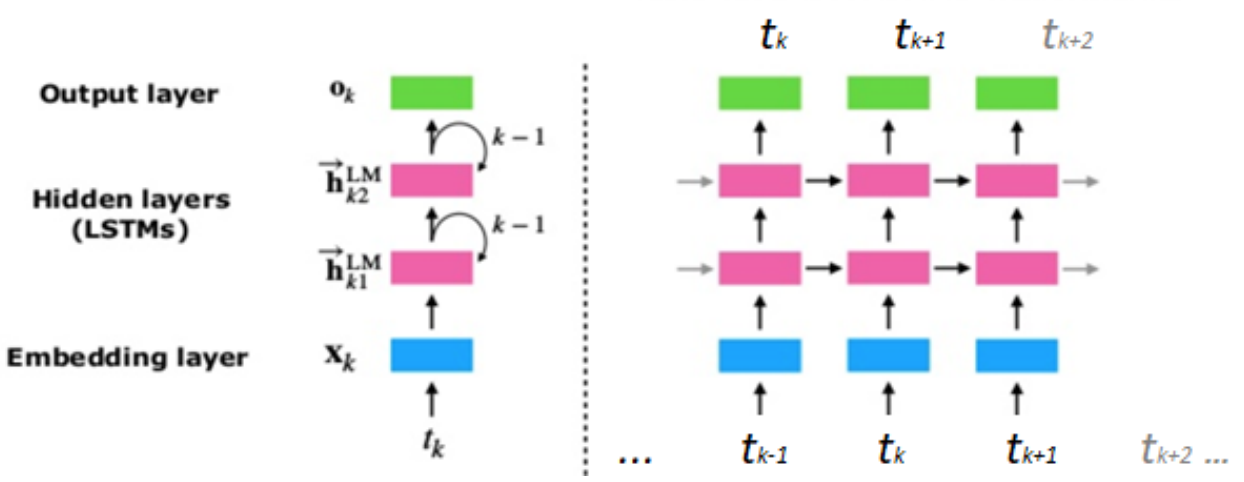


Figure 5: Mechanism of predicting the next word from both directions using bidirectional LSTM neural networks.

1.5.3 Pre-training Bidirectional Deep Transformers for Language Understanding (BERT) [1]

The bidirectional encoder embedding method by transformers is essentially a new approach for embedding using language models, classified as a text-based method. In contrast to the ELMo method, which trains two embeddings for each word (or character)—one from left to right and the other from right to left—and then concatenates them, BERT aims to create a bidirectional language model. This process is based on a simple approach of masking 15% of input words and then passing the entire sequence through a multilayer bidirectional transformer, where the model only predicts the masked words.

In 2017, the concept of a multilayer bidirectional transformer was introduced, inspired by simpler transformers. This concept follows the encoder-decoder architecture used in machine translation models, but with a different network structure and serves as an appropriate alternative to RNN networks [2] [112].

A transformer attempts to learn dependencies, typically encoded by the hidden states of an RNN, using only the attention mechanism. RNN controls dependencies by remembering each state. For example, the current state encodes the necessary information for deciding how to process the subsequent tokens. This means that RNNs must retain states when processing words. However, this method of retaining states is insufficient for applying long-range dependencies between words.

The attention mechanism somewhat mitigates this issue by allowing transformers to learn dependencies using only the attention mechanism, as well as dependencies between input and output tokens. This is achieved through a core component called the **Multi-Head Attention** mechanism, which uses an attention mechanism known as **Scaled Dot-Product Attention** [113].

For model enhancement, instead of calculating a single attention mechanism block, a **Multi-Head Attention** block is used, which is composed of several weighted layers of individual attention mechanisms (Figure 6 and 8). This block is designed such that each of the attention mechanism layers processes the same input linearly, but differently from each other (Figure 7). Therefore, the key feature of transformers is that, unlike

[1] Bidirectional Encoder Representations from Transformers
[2] Recurrent Neural Network

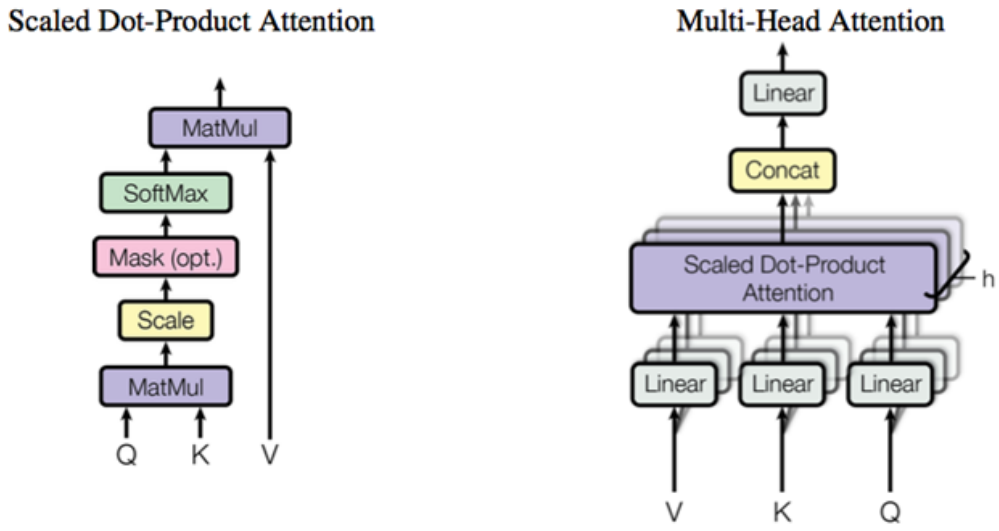


Figure 6: Attention Mechanism. Right: Multi-head attention mechanism combining multiple individual attention mechanisms. Left: Single attention mechanism [112].

RNNs, they store dependencies directly in various parts of the input instead of encoding them in hidden states of model blocks.

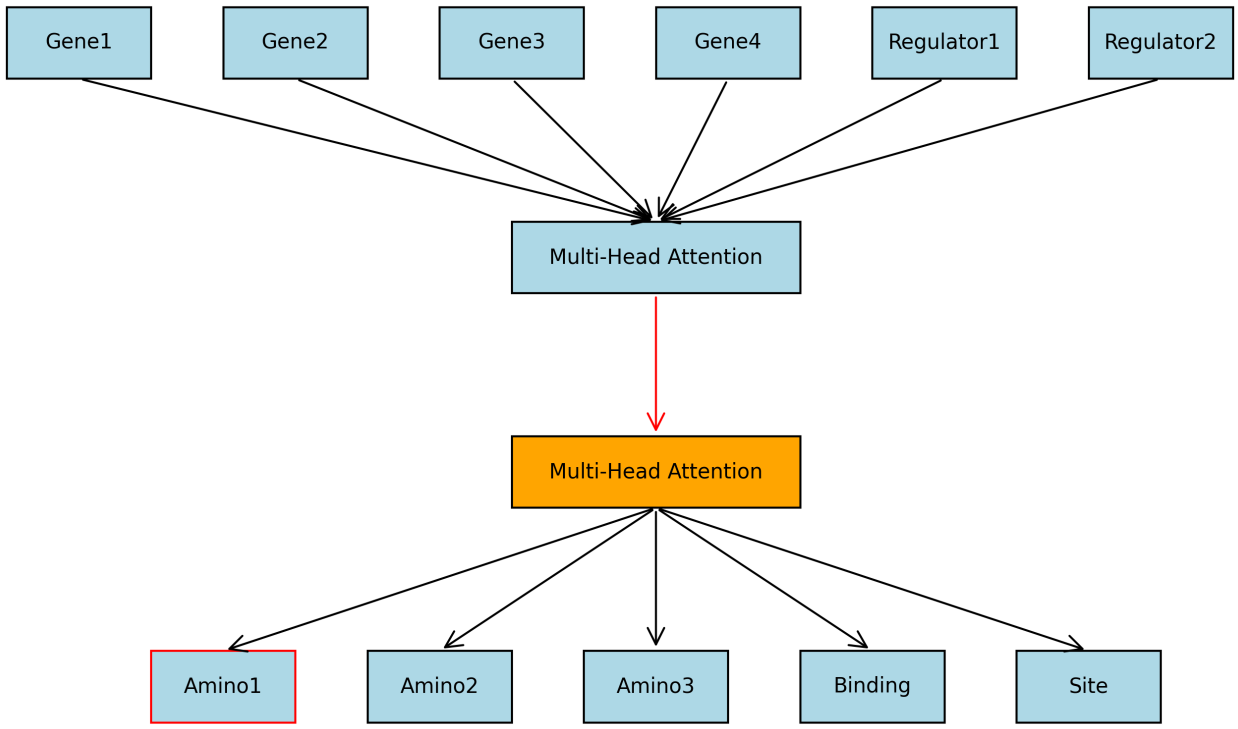


Figure 7: An example of the impact of the attention mechanism in a biomedical context. In sequence-to-sequence tasks, such as protein sequence analysis, the output at step t (e.g., a specific amino acid) is more closely related to certain inputs at that step (e.g., neighboring amino acids or regulatory genes) than to inputs from other steps. These relationships are dynamically calculated, known as the attention mechanism.

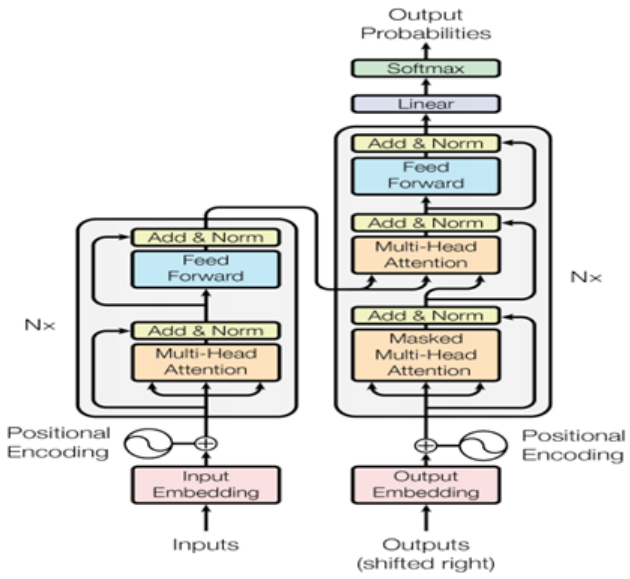


Figure 8: Attention mechanism. (Right) Multi-model encoder attention mechanism. The encoder model on the left and decoder model on the right. Both include a core block of an attention mechanism and a feedforward network, repeated N times. [112]

This concept has been applied in models like BERT. BERT uses **Multi-Head Attention** to understand dependencies and relationships between words in a sentence. Additionally, BERT can also be trained for the next-sentence prediction task. In this case, the model receives two sentences as input and learns whether the second sentence follows the first in the original text. Thus, the use of **Multi-Head Attention** and next-sentence prediction capability makes BERT a powerful tool for natural language processing.

1.6 Problem Statement

Recent research in bioinformatics and genomics has shown that combining machine learning with these fields can lead to significant discoveries and advancements. However, many traditional models, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), fail to extract and understand the semantic dependencies and long-range textual information necessary for understanding the dynamic complexities of nucleotides in mRNA sequences. Furthermore, many existing methods require complete structural data, which limits their applicability to the analysis of new mRNA sequences.

Inspired by the success of BERT-based models in understanding the complex language of non-coding DNA, this research explores and develops a new model called StructmRNA. StructmRNA is a BERT-based model designed to accurately analyze mRNA sequences and structures. Using advanced masking techniques at two levels and conditional masking, this model can generate meaningful embeddings for mRNA sequences, even in the absence of explicit structural data. StructmRNA, leveraging the complex correlations between sequence and structure learned during extensive pre-training on large datasets, has achieved more accurate results in mRNA degradation prediction.

Unlike traditional models, StructmRNA has the ability to predict secondary structures and biological func-

tions of unknown mRNA sequences, which could facilitate RNA-based therapeutic methods. Its superior performance, when compared to the well-known models suggested for Stanford's *OpenVaccine* project, demonstrates that it can also have significant applications in the design of new drugs and vaccines.

The primary goal of this research is to introduce the StructmRNA model as an innovative solution for analyzing mRNA sequences and structures and to evaluate its effectiveness in mRNA degradation prediction and RNA-based therapeutic applications. Given its remarkable capabilities in advancing genomics and RNA-based therapies, this research can set a new standard for mRNA analysis in bioinformatics and medical sciences.

Some key challenges in this area include:

- **Complexity of mRNA Sequences and Structures:** mRNA sequences have a high degree of complexity and variability that traditional models cannot fully comprehend.
- **Need for Complete Structural Data:** Many existing methods require precise structural data, which limits their application to new sequence analyses.
- **Integration of Machine Learning with Genomics:** The effective integration of machine learning techniques with genomics is still in development and requires innovative solutions.
- **Advanced Computational Tools:** With the rapid advancements in mRNA, there is a need for advanced computational tools capable of precise and rapid sequence analysis.

1.7 StructmRNA Model Achievements

The StructmRNA model, leveraging the BERT architecture and two-level and conditional masking techniques, has made significant advancements in the analysis of mRNA as detailed below:

1. **Accurate Prediction of mRNA Structures:** This model is capable of predicting secondary structures and biological functions of unknown mRNA sequences.
2. **Improved mRNA degradation Prediction:** With its superior performance compared to reference models, StructmRNA has demonstrated its potential in accurately predicting mRNA degradation.
3. **Facilitating the Development of RNA-Based Therapies:** This model can play a key role in the design of new drugs and vaccines.
4. **Providing Analytical Tools for Bioinformatics Problems:** By introducing novel masking methods, it sets new standards for analyzing mRNA sequences and structures.

1.8 Thesis Structure

This thesis consists of five chapters. In Chapter 1, the introduction, concepts related to mRNA structural degradation and possible methods for sequence vectorization and degradation prediction are discussed. Chapter 2 reviews the studies conducted and the concepts related to the topic of the thesis. In Chapter

3, the proposed model is described, which includes the use of a pretrained BERT-based model and related techniques. Additionally, the proposed framework based on the entire sequence of the mRNA is explained. Chapter 4, titled Results and Discussion, presents the experimental results from the tests and performance evaluations of the models. Finally, the thesis concludes in Chapter 5 with a summary and introduction to future work.

1.9 Conclusion

In this chapter, after introducing and reviewing the performance of different RNA types, the details of the first and second structures, as well as the importance of determining the secondary structure state in identifying RNA functions, were explained. The conventional methods and important parameters for determining RNA structures, including the free energy minimum state, data on chemical reactivity, thermodynamic information, and the concept of mRNA degradation and its influencing factors, were presented. Lastly, various popular embedding methods in the field of Natural Language Processing (NLP) were discussed in detail. In the next chapter, the expansion of these methods in the biological field will be addressed, which is the focus of the current research.

Given the advantages of embedding vectors over traditional text-based methods, these approaches will serve as suitable replacements for classical methods used in current techniques. Inspired by text embedding methods that create semantically rich word representations through deep understanding of the words and texts they contain, similar modifications can be made to adapt these methods for embedding biological sequences such as nucleotide sequences.

As explained in this chapter, RNA sequences can consist of any of the 64 possible codons. The nucleotides present in each of these codons must be distinguished and uniquely embedded. Therefore, text-dependent embedding methods, using a language model specifically designed for RNA sequence data, will be effective. This language model will be trained on a large collection of RNA sequences, and by capturing the complex patterns both within and between sequences, it will create rich vector representations of these sequences. Ultimately, the generated model will be used to predict RNA structure degradation.

Furthermore, since predicting mRNA structure degradation requires both sequence and structural information, structural data for sequences whose structure is unavailable will be generated by text-based embedding methods. This structural data will be provided as input to the main model.

2 Literature Review

2.1 Introduction

mRNA (messenger RNA) molecules are fundamental components in the molecular biology domain, playing an indispensable role in a wide range of biological and cellular processes. These include, but are not limited to, gene expression, regulation of gene activity, protein synthesis, and involvement in the pathogenesis of numerous viral infections. Given their central role in the flow of genetic information from DNA to proteins, understanding mRNA is crucial for both basic biological research and applied biomedical studies.

Historically, the majority of traditional methods used for analyzing mRNA sequences have heavily depended on explicit structural information derived from experimental techniques. These conventional approaches often necessitate access to detailed structural data, which is not always available or feasible to obtain, particularly for newly discovered or poorly characterized mRNA sequences. This dependency stems largely from the lack of comprehensive, publicly available datasets that encompass the diverse structural variations of mRNA molecules [90]. As a result, these traditional methods face significant limitations in scalability and adaptability when applied to novel or unannotated sequences.

In response to these limitations, computational approaches have emerged as indispensable tools for advancing mRNA analysis and interpretation [95]. These methods offer a valuable alternative by enabling researchers to infer structural and functional properties of mRNA molecules using algorithmic techniques, without the need for direct structural data. The development and application of such computational frameworks are especially critical in the context of large-scale genomics and transcriptomics studies, where high-throughput analysis is essential.

The progress in computational biology and RNA informatics has been thoroughly documented in recent literature, particularly concerning the modeling of mRNA degradation dynamics and structural variability [129], [130]. Although these approaches have introduced innovative strategies, they still grapple with certain inherent challenges, such as the structural heterogeneity of mRNA and the biological complexity underlying regulatory mechanisms.

To address these persistent issues, modern advances in artificial intelligence, particularly in the realm of machine learning—and more specifically, deep learning—have demonstrated promising potential. These techniques have revolutionized the way complex biological data is modeled and interpreted. For instance, transformer-based architectures and neural networks capable of learning intricate sequence-structure relationships have been increasingly applied to mRNA data [32], [34]. Such methods hold the promise of uncovering latent patterns and features that were previously inaccessible using classical approaches. The implications of these innovations will be discussed in greater detail in the following sections.

The field of embedding learning, particularly in the context of biological data, has witnessed considerable advancement in recent years [58]. These developments have laid a strong foundation for a deeper understanding of complex biological systems through representation learning techniques. In parallel, there has been a surge of progress in the development of methods aimed at the degradation, interpretation, and com-

putational analysis of biological sequences, with RNA molecules receiving particular attention due to their structural and functional complexity [4].

A wide array of machine learning and deep learning models have been introduced to tackle the challenges associated with RNA sequence representation and structural modeling. Among these, sequence-to-sequence autoencoders have proven to be highly effective in capturing latent representations of sequence data [99]. In addition, Convolutional Neural Networks (CNNs) have shown promise in identifying local patterns within sequences, while Long Short-Term Memory (LSTM) networks are adept at modeling temporal dependencies and sequential characteristics [118]. Variational Autoencoders (VAEs), by learning probabilistic latent spaces, have facilitated the generation and interpolation of RNA sequences [37]. Furthermore, Graph Neural Networks (GNNs) have emerged as powerful tools for modeling the non-linear, graph-like structure of RNA molecules and have contributed significantly to structural analysis tasks [85, 122]. Collectively, these models have greatly enriched the toolbox available for computational RNA biology, enabling more sophisticated and accurate analyses.

In addition to the aforementioned models, specialized embedding techniques have been proposed to map nucleotide sequences into dense vector representations. Notably, methods such as `dna2vec` and `rna2vec` were specifically designed for encoding DNA and RNA sequences, respectively, into continuous vector spaces that preserve biological meaning and contextual relationships [114]. More recently, the adaptation of transformer-based architectures, most prominently BERT, to the domain of biological sequence analysis has brought about a paradigm shift in how such sequences are processed and understood. These models, such as DNABERT, have demonstrated superior performance in various biological tasks by capturing long-range dependencies and contextual semantics in sequence data [37, 56].

The advancements discussed above in the field of bioinformatics clearly highlight the effectiveness and growing capability of computational techniques, particularly in accurately predicting the degradation probability of messenger RNA (mRNA). These developments underscore the potential of leveraging advanced machine learning and deep learning frameworks for solving complex biological problems. In this context, the task of predicting the degradation probability of mRNA structures through deep learning-based approaches necessitates the transformation of biological dataspecifically RNA sequences and their associated structural information into a low-dimensional numerical representation, often referred to as an embedding space. This transformation is essential because it enables the deep learning model to process and learn from high-dimensional biological data in a computationally efficient manner.

Within the domain of biological data analysis, particularly in problems involving RNA sequences, traditional computational methods are typically employed in one of two main forms: sequence-based methods or structure-based methods [11, 127]. Sequence-based approaches operate by taking the linear RNA sequence and converting it directly into a vector space using classical embedding techniques. These include widely known natural language processing (NLP) models such as Word2Vec, GloVe, and fastText, which have been adapted to work with biological sequences by treating nucleotides or k-mers as tokens, similar to words in a sentence.

In contrast, structure-based methods aim to capture not only the linear nucleotide information but also the secondary and tertiary structural conformations of RNA molecules. These approaches often involve the construction of a graph representation of the RNA structure, wherein nucleotides are treated as nodes and their interactions—such as base pairing—are represented as edges. Once the graph is generated, it is then embedded into a vector space using one of the aforementioned embedding techniques, thereby preserving both sequence and structural context (Figure 9). This method is especially useful in applications where structural information plays a critical role in understanding RNA behavior, such as stability, localization, or degradation.

It is important to note that the embedding models employed for RNA representation in existing studies have predominantly been classical in nature. While these traditional models have demonstrated effectiveness, they may have limitations in capturing complex dependencies and contextual nuances inherent in biological sequences. In this chapter, we provide a comprehensive overview of the different embedding techniques that have been developed and applied within the biological domain. Following this review, we delve into the specific models that have been utilized in prior research to predict RNA sequence degradation, outlining their methodologies, strengths, and limitations in order to establish a foundation for the proposed deep learning framework.

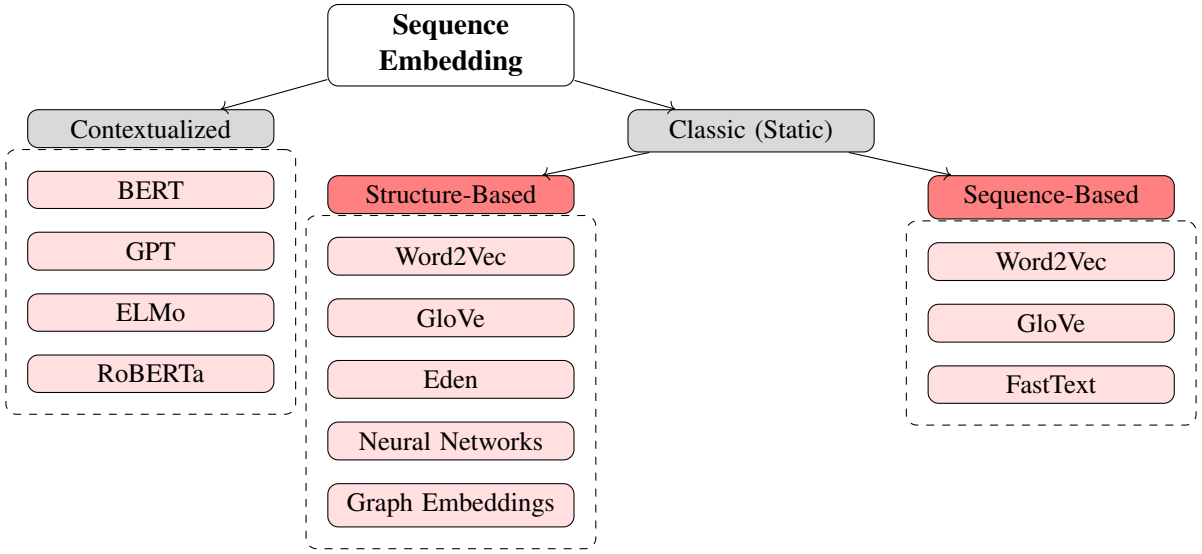


Figure 9: Methods Applicable for Embedding Biological Sequences

2.2 Classic Embedding Methods

Biological sequence embedding, such as protein, DNA, and RNA sequences, has been performed in previous studies in two ways: sequence-based and structure-based methods [12] [119]. In sequence-based methods, the vector representation of each protein, DNA, and RNA is derived solely from their sequence. In other words, these sequences are directly represented as vectors using embedding methods such as Word2Vec, GloVe, or fastText [13] [43] [136] [116]. On the other hand, structure-based methods create a graph corresponding to the sequence and aim to automatically learn low-dimensional feature vectors for each node

in the graph [107]. Simply put, low-dimensional vectors are learned with detailed structural information of the sequences. For RNA sequence embedding using this method, a labeled path graph of RNA structural information needs to be created, and the graph is then mapped to a vector space using one of the embedding methods. Each node in this graph corresponds to a nucleotide in the RNA sequence, and each edge is labeled with either a ribose-phosphate backbone or hydrogen bond between two nucleotides. Finally, each node in the generated graph can be converted into a set of vectors using any of the embedding methods [107]. The vectors generated from each of these two methods can serve as features in building machine learning models for various tasks such as prediction, node classification, and clustering [41] [19] [57]. Graph-based embedding methods are more commonly applied at a higher level, such as molecules, proteins, diseases, drugs, and their interrelationships, rather than directly on molecular sequences, nucleotides, and amino acids. Traditional methods like Laplacian Eigenmaps (LE) [80] and Matrix Factorization (MF) [7] have shown promising results in various problems like degradation and biological graph analysis. Moreover, it has been shown that newer graph-based embedding methods based on neural networks outperform traditional methods in most non-medical problems [135]. Due to the popularity of these new methods, many graph embedding methods have been introduced for biological networks [126]. However, these methods have still been overlooked in the context of converting RNAs into vectors.

In conclusion, graph-based embedding techniques, when applied to domains beyond RNA sequences, can be systematically classified into three distinct categories, each characterized by unique methodologies and approaches to generating embeddings:

- The first category encompasses matrix factorization techniques, which rely on a data matrix, such as an adjacency matrix representing the graph structure, as the primary input. These methods aim to learn low-dimensional embeddings by performing matrix factorization or decomposition, thereby transforming the input matrix into a set of vector representations that capture the essential structural properties of the graph in a computationally efficient manner.
- The second category comprises methods that leverage random walk-based strategies on adjacency graphs to generate embeddings. In this approach, random walks are employed to produce sequences of nodes, effectively simulating paths through the graph. These node sequences are subsequently processed using the Word2Vec model, as described in the referenced work [81], to train and generate node embeddings that encode the structural and relational information inherent in the graph's topology.
- The third category includes neural network-based approaches, which generate enriched embeddings through sophisticated optimization algorithms. These methods typically start with random initialization or one-hot encoding of nodes and utilize neural network architectures to learn embeddings. The models within this category are designed with diverse architectures and accept a variety of input types, as illustrated in Figure 10, enabling them to capture complex patterns and relationships within the graph data through iterative optimization processes.

Results from graph embedding methods applied to biological network graphs indicate that high-order proximity modeling methods [70] perform better in edge prediction problems compared to node classification

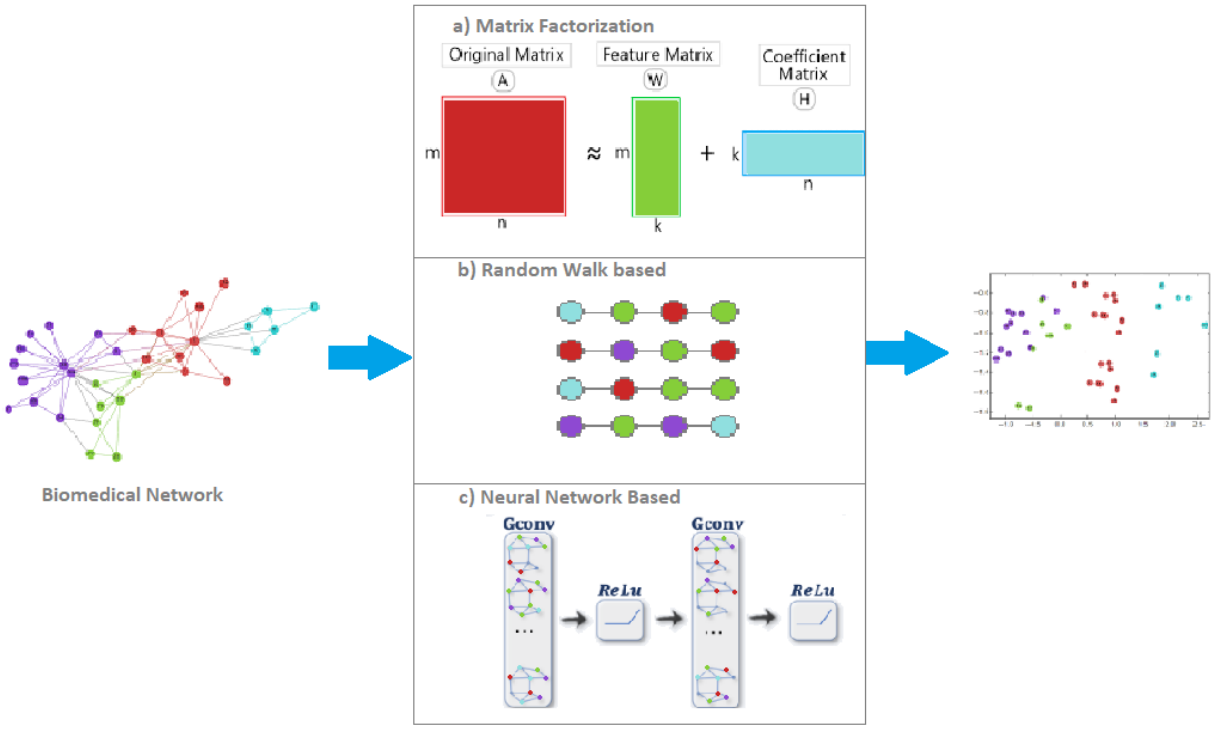


Figure 10: General framework of graph embedding methods in biomedical works.

tasks. On the other hand, random walk-based methods show better performance in node classification problems [41] [19]. Notably, one of these models, called Struc2vec, has shown better results in edge prediction problems.

Figure 11 illustrates the comprehensive distribution of academic research papers that concentrate on the domains of biological sequence data and embedding learning methodologies, with the vast majority of these scholarly works having been published over the span of years from 2016 to 2023. Within this section of the document, we provide an in-depth exploration and detailed description of the advancements and notable progress achieved in this particular area of scientific inquiry. Furthermore, we introduce a systematic classification framework that organizes embedding learning techniques by considering multiple key attributes, including the architectural design of the models employed, the performance metrics utilized for evaluation, and the techniques' suitability and applicability to various distinct types of RNA sequences encountered in biological research.

Recent publications in bioinformatics focus on embedding learning, prediction algorithms, and sequence analysis of biological data such as DNA [124], RNA [4], and proteins [65] [38]. Most studies from 2019 to 2022 use various machine learning algorithms and models. For instance, sequence-to-sequence autoencoders have been used for embedding DNA [3] and RNA [99] sequences, while CNN and LSTM are widely used in predicting sequence repetitive patterns and structural features [36] [64] [105] [118]. Additionally, several review papers summarize the methods, techniques, or applications in these areas, including the use of Graph Neural Networks (GNN) [122], Variational Autoencoders (VAE), and Transformers [37].

Some of the most significant studies in this area include the Unirep vector space for encoding structural [8] and evolutionary and functional information for proteins; the Deepred-Mt application of deep convolutional neural networks for predicting C-to-U RNA editing in plant mitochondria [36], and RNABERT for RNA informatics [5]. Techniques like Word2Vec, one-hot encoding, and more specialized algorithms like dna2vec and rna2vec are used for gene embedding [66]. Various papers also investigate or apply transfer learning [28], graph embedding learning, and machine learning algorithms for tasks such as predicting interactions between proteins, RNA, and DNA [100] [72]. Some studies review previous research, while others take practical approaches for sequence analysis and degradation prediction [27].

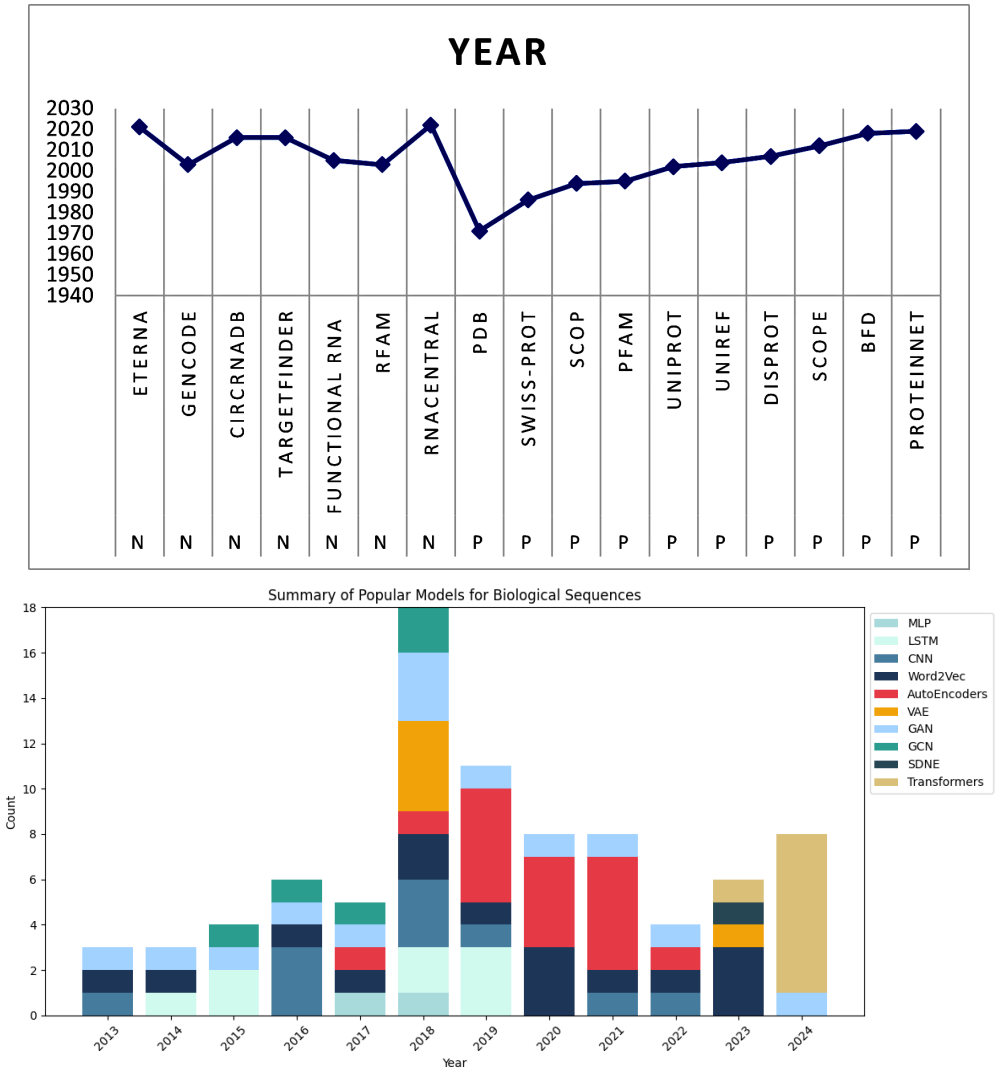


Figure 11: Top: The number of different models used between 1994 and 2023, Bottom: The distribution of published database sets between 1994 and 2023, categorized by protein (p) and nucleotide (N) data.

As shown in Figure 12, neural network models can be classified based on several aspects such as model type, application domain, learning type, model architecture, and input data type. For example, Word2Vec and ELMo focus on embedding, CNN and LSTM are specialized neural network models, VAE is a generative

model, and AGC [1] and BERT act as deep neural network models based on attention mechanisms and graphs. The application domains of these models range from natural language processing and computer vision to graph-based machine learning tasks. In this research, we have chosen mRNA degradation as the focus of our

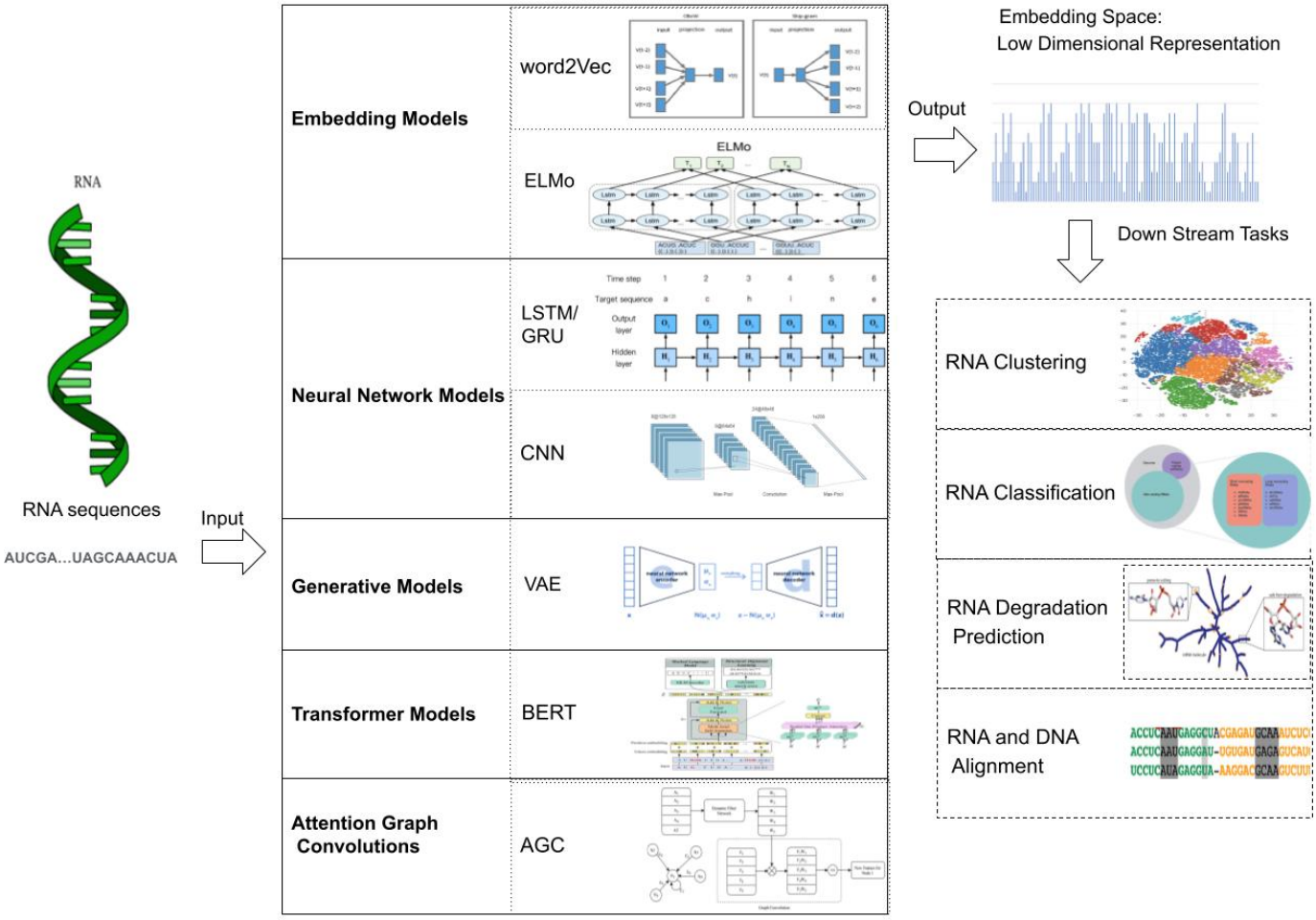


Figure 12: Schematic representation of RNA sequence embedding methods and their applications.

study to gain deeper insights into RNA sequencing. Our goal is to explore the degradation problem and analyze the performance of different embedding models, each with its own strengths and unique characteristics. We will explore models like Word2Vec and ELMo, which are known for their ability to capture semantic nuances, as well as specialized models such as CNN and LSTM, generative models like VAE, and attention- and graph-based models like AGC and BERT.

Prioritizing the mRNA degradation problem allows us to uncover complex relationships between embedding models and their ability to predict mRNA degradation accurately. This choice is based on the belief that advancements in this specific area could lead to significant progress in genomics, personalized medicine, and our understanding of fundamental biological processes. This research not only contributes to enriching the scientific community's knowledge but also enables the development of more accurate and effective tools

[1] Adaptive Graph Convolution

for RNA sequence analysis.

2.3 Text-Based Embedding

Despite the existence of numerous studies focused on generating embeddings for proteins based on their textual or sequential representations [79], the application of such text-based embedding techniques to nucleotide sequences remains relatively limited. To date, the only prominent approach that has successfully mapped nucleotide sequences into a continuous vector space using these methodologies is the DNABERT model [56]. This model specifically targets DNA sequences and employs a deep learning architecture inspired by natural language processing frameworks.

DNABERT operates by first encoding DNA sequences through an embedding layer, which transforms discrete nucleotide tokens into dense vector representations. These embeddings are subsequently processed by a series of 12 transformer blocks, which leverage the power of bidirectional self-attention mechanisms to capture complex patterns and relationships within the sequence. The bidirectional transformers in DNABERT generate comprehensive text embeddings by simultaneously attending to both upstream and downstream contexts of each nucleotide, thereby enabling the model to maintain and recall long-range dependencies and interactions across the entire sequence.

This ability to model long-range dependencies is particularly important for biological sequences, where distant nucleotides may influence functional or structural properties. Consequently, DNABERT's transformer-based embedding approach provides a powerful framework for representing DNA sequences in a manner that preserves their intricate contextual information, facilitating downstream tasks such as sequence classification, motif detection, and functional annotation. The success of DNABERT highlights the potential of adapting state-of-the-art natural language processing models to genomic data, paving the way for further innovations in sequence embedding methodologies.

The previously mentioned pretrained model, which draws inspiration from the BERT methodology, generates vector representations at both the DNA sequence level and the token level, accommodating sequences of arbitrary length. Nevertheless, there remains a substantial and noteworthy deficiency in the availability of pretrained text models specifically designed for RNA sequences, highlighting a critical area for further development and research in the field.

The DNABERT model, specifically engineered for the analysis and processing of DNA sequences, primarily concentrates on the sequential data inherent in DNA without incorporating the structural characteristics of the molecule. This limitation suggests that the model overlooks critical three-dimensional conformational details and other structural nuances of DNA that could be significant for certain applications. Consequently, this focus on sequence data alone may lead to substantial challenges when attempting to apply the DNABERT model to tasks involving RNA sequences. The inherent differences between DNA and RNA, including their distinct molecular structures, chemical compositions, and functional roles, could render the DNABERT model unsuitable or, at the very least, highly inefficient for generating accurate and effective RNA vectors. These shortcomings highlight the need for models that can account for both sequential and

structural information to achieve more robust and versatile genomic analyses.

2.4 Prediction of mRNA Degradation Probability

In recent years, a substantial number of researchers and scientists have been dedicating their efforts to investigating the potential of mRNA vaccines, primarily due to the remarkable advantages these vaccines offer in terms of production speed and simplicity when compared to traditional vaccine manufacturing methods [110] [23]. The significance of this innovative approach became particularly evident during the global pandemic, when the urgent need for rapid vaccine development was paramount to mitigate the devastating health, economic, and societal impacts, thereby helping to avert substantial material losses and profound moral costs associated with prolonged delays in vaccine availability. However, one of the most formidable challenges associated with mRNA vaccines lies in ensuring their stability over time. Specifically, these vaccines are highly susceptible to self-degradation if not meticulously stored under stringent conditions, typically requiring advanced, high-performance refrigeration units to maintain their efficacy and integrity [26] [62].

In order to find stable mRNA sequences, researchers at Stanford University have organized competitions to predict the degradation probability of the COVID-19 mRNA vaccine in collaboration with the Kaggle community. To achieve this, investigating RNA structural reactivity under various experimental conditions such as temperature, pH, and the presence of ions like magnesium at the transcriptome level helps determine the degradation probability of these molecules. Furthermore, examining this information contributes to the structural interpretation of these molecules, which reveals their identity and function within the cell. Specifically, the goal of the Kaggle competition was to predict the reactivity and degradation of nucleotide bases in RNA sequences under specific laboratory conditions such as in the presence or absence of magnesium, at high temperatures, and different pH levels.

The mentioned problem is a regression problem for continuous reactivity values under different laboratory conditions. The proposed methods generally utilize RNA sequence data in the form of a vector matrix and one-by-one vectors during the preprocessing stage. In terms of neural network architecture, these methods can be divided into three main categories based on LSTM, GRU [1], and GCN [2]. GRU networks [125], with their faster speed compared to LSTM networks (due to simpler architecture), have shown better performance in this problem. On the other hand, GCN networks, which offer a new method for degradation and matrix analysis in deep neural networks using CNNs, have provided the best predictions.

In one of the hybrid solutions informally proposed for this competition, a combination of GRU and LSTM neural networks has been introduced as highly suitable for these types of conditions, as their internal memory performs better in predicting when temporal patterns or long-term dependencies exist. In this method, multiple features related to nucleotides and their relationships are first extracted using existing tools.

These features, along with the RNA sequences, are then fed into transformers to learn the vectors corresponding to each of these sequences, and finally, these vectors pass through two layers of bidirectional RNNs. In

[1] Gated Recurrent Unit

[2] Graph Convolutional Network

each experiment, these two layers are replaced with one of the combinations GRU + LSTM, GRU + GRU, LSTM + LSTM [111].

2.5 Summary

This chapter addressed two main topics. In the first part of this chapter, embedding models in the biological field, especially for RNA sequences, were discussed. There are various models for embedding biological data such as proteins and DNA, but for RNA sequences, most methods are limited to classical and non-text-based techniques. These limitations mean that more labeled data are needed for generalization and improvement of model performance in RNA-related tasks.

Developing a pretrained language model on RNA sequence and structure data and utilizing the advantages of text embedding methods can help solve this issue. However, since existing models have mostly been tested on DNA sequences and there are differences between DNA and RNA, using these models for RNA sequences has not been optimized.

Next, the estimation of RNA sequence degradation was reviewed. Despite the existence of extensive datasets, predicting stable mRNA molecules still faces challenges. Using deep learning architectures to predict the degradation of these molecules seems logical. One of the key issues is the use of non-text methods for embedding RNA sequences, which makes it difficult to preserve long-range dependencies between nucleotides and increases the likelihood of incorrect degradation predictions.

Additionally, the proposed models in previous studies generally require more labeled data for optimal performance and suffer from reduced efficiency when predicting sequences longer than 107-130 nucleotides, while the actual length of mRNA in vaccines ranges between 3000 to 4000 nucleotides. These limitations can be significantly mitigated with the help of a pretrained model on actual mRNA sequence lengths.

3 Method

3.1 Introduction

In this chapter, we introduce the proposed model for predicting the degradation of mRNA sequences using deep learning-based approaches with the suggested model (StructmRNA). To improve the performance and convergence speed of the proposed model, we will use Generative Adversarial Networks (GAN) based techniques for data augmentation. Additionally, in the proposed model, we will incorporate not only sequence data but also structural mRNA data as inputs. This is achieved by embedding mRNA sequence structural details and making modifications to the BERT architecture to generate enriched vectors for mRNA sequences. These vectors will contain structural details of mRNA, including local and global nucleotide dependencies within the sequence and the interactions between mRNAs.

As discussed in the previous chapter, the embedding of biological sequences, particularly RNA sequences, into vector representations is of great importance. This is because the embedding of these sequences forms the foundation for solving the problem of predicting mRNA structural degradation using machine learning methods. While many methods have been applied to embedding these sequences inspired by natural language processing techniques, there is still a need for embedding methods specifically designed for these sequences. Furthermore, due to the significant need for large labeled datasets for neural networks in the problem of predicting mRNA structural degradation, creating a model based on vast RNA sequence texts and training an embedding model as a pretrained model can increase the generalizability of the neural network.

Although a pretrained embedding model for DNA sequences has been designed, the significant differences between RNA and DNA molecules, along with the lack of involvement of sequence structure in the generated vectors, make the existing model unsuitable for RNA sequences. Therefore, it is necessary to create a pretrained embedding model based on RNA structural details and generate enriched vectors for these sequences.

While the primary focus of this study is on RNA sequence and structures representation, we also explore graph-based representation learning in biological networks like PPI. To this end, we introduce IsoGloVe, a novel embedding method that leverages geodesic distances for protein-protein interaction (PPI) networks. This approach complements sequence-based models like StructmRNA by capturing topological relationships in biological data, demonstrating the broader applicability of representation learning in bioinformatics.

3.2 Introduction to IsoGloVe

During the development process of StructmRNA, we introduced a method called IsoGloVe as a novel graph-based embedding technique for biological networks. IsoGloVe offers a new perspective by introducing vector embeddings based on graphs.

IsoGloVe is a method for generating dense graph embeddings. This is done by calculating the geodesic distances between nodes. The resulting embeddings can be used for tasks such as node classification, link prediction, and community detection. IsoGloVe generates vector embeddings in three steps:

Step 1: IsoGloVe generates a short random walk for each node by randomly choosing a node and performing a random walk of length L using the list of edges. The random walk algorithm preserves the local neighborhood structure of nodes by randomly selecting a node and performing a random walk of length L.

Step 2: IsoGloVe creates a large matrix of co-occurrence of nodes from the random walks. In this matrix, the i_{th} row and j_{th} column represent the co-occurrence of nodes i and j in the random walks. The matrix of co-occurrence needs to be decomposed into a lower-dimensional matrix (D), where each row represents a node's embedding vector. The size of D depends on the graph's size, and there are high correlations between the rows associated with nodes close to each other in the graph, which is why matrix decomposition is performed.

Step 3: The vector representation of each node is trained by minimizing the difference between the geodesic distance between embeddings and the logarithm of their co-occurrence. To do this, the co-occurrence counts in the matrix are first normalized, and then their logarithms are taken. Subsequently, the geodesic distance between each node pair is calculated. To train the embeddings, the same GloVe error function is used, but with geodesic distance instead of the dot product. The IsoGloVe loss function is expressed as:

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \left\{ (d(n_i, \tilde{n_j}) + b_i + \tilde{b_j} - \log(X_{ij})) \right\}^2 \tag{1}$$

where $d(n_i, \tilde{n_j})$ is the geodesic distance between the embeddings of node n_i and context node $\tilde{n_j}$ for nodes i_{th} and j_{th} in the vocabulary V. b_i and $\tilde{b_j}$ are their respective biases, and X_{ij} is the co-occurrence probability between nodes i_{th} and j_{th}. The function $f(X_{ij})$ reduces the impact of highly frequent node pairs. The similarity between n and $\tilde{n_j}$ in the Euclidean space can be computed by calculating the dot product of n and $\tilde{n_j}$ [40]. However, the dot product of two vectors is not always identical to their similarity in geodesic space, and other measures may be more suitable for similarity.

The geodesic distance is the sum of edge weights along the shortest path between two nodes. The primary vectors N from the geodesic distance matrix represent coordinates in the new N-dimensional Euclidean space. In this study, the standard Riemannian distance metric is used to calculate the distance between two points in the embedding space. Using this metric allows the intrinsic curvature of the embedding space to be used in modeling the nonlinear relationships between nodes [39]. Initially, the geodesic distances between every pair of nodes in the high-dimensional space are computed, and these distances are used to define a Riemannian metric. Then, the Riemannian metric is used to embed the nodes into a low-dimensional space, where the distances between each pair of embedded nodes closely match the original geodesic distances. More specifically, given a Riemannian manifold (M, g), where M is the underlying space and g is a Riemannian metric matrix defined on M, the distance $d(p,q)$ between two points p and q in M is given by the shortest path between them, known as the Riemannian distance:

$$d(p,q) = \inf_{\gamma} L(\gamma) \tag{2}$$

The `inf` denotes that the length $L(\gamma)$ must be computed for all possible paths γ that connect the two points p and q. Then, the shortest length or minimum $L(\gamma)$ is chosen. This minimum value is the Riemannian distance between points p and q.

Thus, the role of `inf` here is to find the shortest path between two points from all possible paths. This is crucial for accurately calculating the geodesic distance in the Riemannian space.

The infimum is computed over all paths γ connecting p and q, and $L(\gamma)$ is the length of path γ, defined as follows:

$$L(\gamma) = \int_a^b \sqrt{g(\gamma'(t), \gamma'(t))}\, dt \tag{3}$$

where a and b are the start and end points of the path γ, and $g(\cdot,\cdot)$ is the Riemannian metric tensor.

Several challenges exist in accurately measuring geodesic distances. The first challenge is that if the k-nearest neighbor graph is not connected, then the shortest path between some pairs of nodes may not exist, and the value $d(w_i, w_j)$ will be undefined. In this case, the IsoGloVe error function becomes infinite. Therefore, it is important to use an algorithm such as Dijkstra's algorithm, which finds the shortest path between two nodes in a graph regardless of its connectivity.
Second, when using Dijkstra's algorithm, the memory required to store the distance matrix and the eigenvector decomposition of a matrix can become computationally expensive for large graphs. The time complexity of computing the eigenvalue decomposition of a matrix scales as $O(n^3)$, where n is the number of nodes in the graph, making it infeasible for large graphs with tens of thousands or even millions of nodes.
IsoGloVe uses an algorithm called Lanczos [68] to perform eigenvalue decomposition of the geodesic distance matrix to find node embeddings in a low-dimensional space. This method is designed to capture the underlying structure of the graph, resulting in compact and interpretable vector embeddings that preserve both global and local relationships between nodes. The resulting embeddings can explain a significant portion of the variance in the original high-dimensional graph.

The overall computational complexity of IsoGloVe for a graph $G(V,E)$, considering K nearest neighbors for each node, is $O(V^2(K+\log(V))) + O(V^2)$.

The term $O(V^2(K+\log(V)))$ represents the complexity of searching for the nearest neighbors and computing the shortest path search on the graph to find the geodesic distance between two nodes in the graph. $O(V^2)$ represents the complexity of learning the d-dimensional embeddings from the co-occurrence matrix.

3.3 Advantages of IsoGloVe for Graph Embedding

Some of the advantages of IsoGloVe for biological network graph embedding are as follows:

- **Improved vector embedding**: IsoGloVe, by leveraging geodesic distances and feature-based vector embeddings, effectively captures complex relationships within biological networks, providing richer vector embeddings compared to traditional methods.
- **Improved prediction performance**: By encoding the PPI interaction networks, IsoGloVe enhances prediction performance in tasks such as node classification and graph reconstruction, which are crucial

for understanding biological pathways and interactions.

- **Diversity in biological networks**: In addition to PPI networks, the IsoGloVe method can be extended to analyze and model diverse biological networks such as gene regulatory networks.

Despite the advantages of IsoGloVe, there are reasons for its development and non-integration into the main StructmRNA model, as outlined below:

1. **Focus on core prediction algorithms**: StructmRNA prioritizes the refinement and optimization of RNA structure prediction algorithms. Integrating IsoGloVe, which focuses on graph embedding techniques, requires extensive software development and evaluation, which could divert attention from the primary goal of RNA structure prediction.
2. **Specific features of graph embedding application**: While IsoGloVe demonstrates significant improvements in vector representation for biological applications using geodesic distances and co-occurrence statistics, its application may not directly align with the core goal of predicting RNA secondary structure in StructmRNA.
3. **Complexity of validation and integration**: Integrating IsoGloVe requires careful validation on reliable datasets for RNA structure prediction. This introduces complexities in the integration and validation process, which could lead to delays in the practical use of StructmRNA.

3.4 Proposed Method

The proposed approach, StructmRNA, integrates advanced computational techniques for analyzing and embedding RNA sequences and structures and employs a comparative framework for various baseline models to gain a comprehensive understanding of the current model's capabilities in predicting mRNA degradation. The core of the proposed model in this research utilizes the BERT framework, enhanced with a two-level masking strategy aimed at providing an accurate analysis of mRNA sequences.
This model includes defining masking thresholds and developing specific strategies for masking sequences and structures, which are crucial for the model's ability to accurately predict mRNA sequences and their corresponding structures. To further understand the mRNA sequences and their impact on the degradation prediction problem, detailed explanations of the model architecture, training protocols, dataset configuration, and data loading settings are provided.

Figure 14 illustrates the overall architecture of the proposed model. This diagram presents a comprehensive process that includes data configuration and model training stages in this research. The stages are as follows:

1. **Obtain the primary dataset:** This process begins with the **RNA primary dataset**.
2. **Masking:**
   (a) A **masking process** is performed on the primary dataset to generate the columns named masked_sequence and masked_structure.

(b) These modified columns simulate scenarios where certain nucleotides or structural elements are unknown, thus providing a more realistic training environment for the model.

3. **BERT-based tokenization:** After masking, BERT-based tokenization is applied to split sequences and structures into a form that can be used for training and prediction in the proposed model.

4. **Managing tokenized data:**

   (a) The tokenized data is then managed in a custom PyTorch Dataset class, specifically designed to handle the complexities of RNA data and facilitate their efficient management during the training phase.

   (b) A DataLoader with a batch size of 16 processes the PyTorch mRNA dataset in batches.

5. **Model training:**

   (a) The final step, "Model training," involves training the embedding learning model based on BERT using the prepared data.

   (b) This model generates the computational vectors needed as input for mRNA degradation prediction.

Figure 13 shows the architecture of the StructmRNA model used in the StructmRNA project, specifically designed for predicting sequence and structure in the masked language model context. The masking process begins by receiving initial input sequences and progresses through steps such as masking both the sequence and structure simultaneously, enabling the prediction of masked data using embedding layers, concatenation, and eight transformer layers.

The following sections will explain each of the components of the proposed model.

3.5 Two-Level Masking Process

The two-level masking process is the core of StructmRNA, involved in integrating sequence and structural information for generating accurate embeddings or embeddings of mRNA sequences. This section discusses the details of masking at both the sequence level and the structure level. Sequence-level masking is inspired by the BERT masking technique, where mRNA sequence nucleotides are randomly replaced with a mask token. This method guides the model to predict masked nucleotides based on the contextual content, learning the underlying sequence patterns and dependencies.

Structure-level masking, alongside sequence-level masking, targets the structural elements of mRNA. This dual approach allows the model to learn how sequences are transformed into specific structural motifs and emphasizes the role of structure in understanding mRNA function.

In this research, a masking probability of 25% was selected for each nucleotide or structural element. This value was carefully determined after several reviews to strike a balance between preserving informational

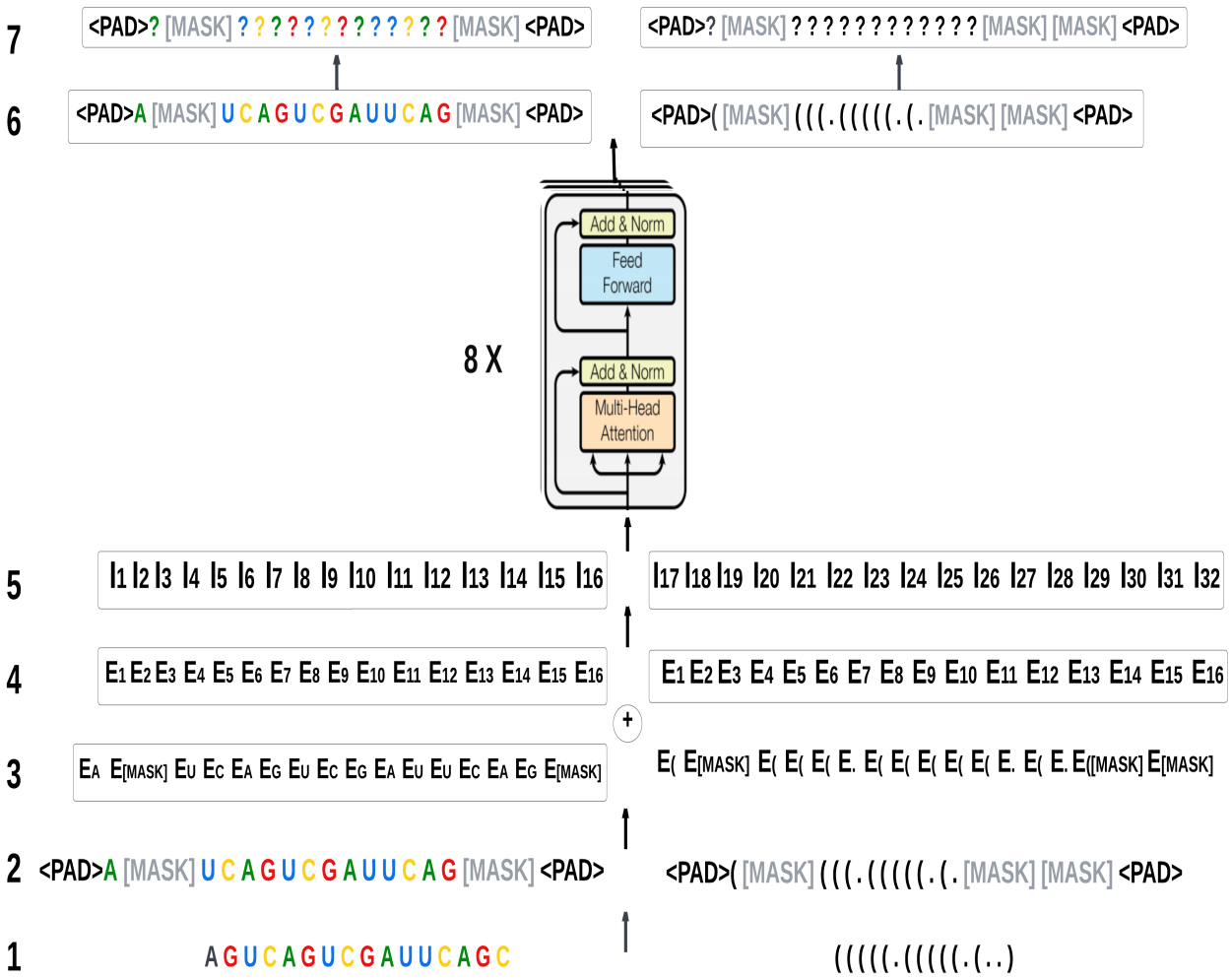


Figure 13: The process of simultaneous masking of sequence and structure in StructmRNA.

data and adding uncertainty to the model. The goal of this choice is to enhance the model's generalization ability and prevent overfitting to the training data.

This higher probability helps the model understand complex relationships between mRNA sequences and secondary structures in a broader sense, avoiding dependence on specific features. This choice is also based on the results and experiences of previous research. For instance, in the BERT paper published by Devlin et al. in 2019, a masking probability of 15% was considered. In our study, due to the complexity and specific features of the data, a masking probability of 25% was chosen to achieve optimal performance.

Therefore, this choice is based not only on theoretical principles but also on the specific needs of this research, aiming to improve the model's final performance. The proposed masking strategy uses a probabilistic decision-making process, where each nucleotide is compared with a randomly generated number, and with a probability of p, a decision is made whether the nucleotide should be masked or not. This process is described as follows:

For a sequence of nucleotides $S = \{s_1, s_2, \ldots, s_n\}$, each nucleotide s_i is compared with a random number r_i,

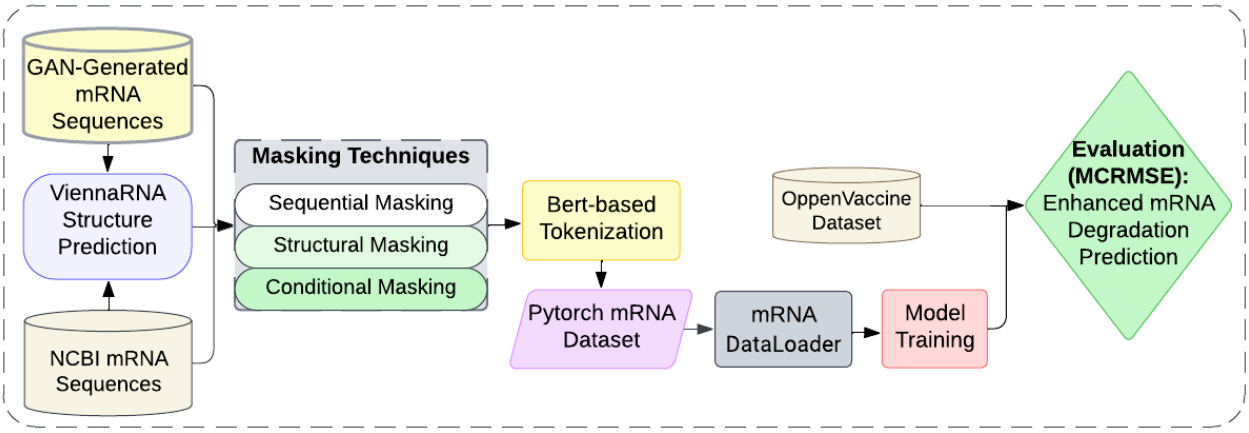


Figure 14: Different phases of StructmRNA from mRNA production to model evaluation. Input sequences (generated by GAN or extracted from NCBI [1]) are processed using the ViennaRNA tool to predict structure. Then, the masking process is applied, and tokenized data is organized into a PyTorch dataset and processed through the DataLoader for model training. Finally, model evaluation is carried out using the OpenVaccine dataset and the MCRMSE metric for mRNA degradation prediction.

which is uniformly distributed between 0 and 1. This uniform distribution is chosen for simplicity and to provide direct interpretability of the chosen masking probability, while ensuring proper random behavior. This method directly determines the masking probability and is very simple to implement. If $r_i < p$ (where p is the masking probability), s_i is replaced with the token `[MASK]`. Equation (4) shows the masking process mathematically:

$$s'_i = \begin{cases} \texttt{[MASK]} & r_i < p, \text{ if} \\ s_i & \text{otherwise} \end{cases} \tag{4}$$

Advanced structure-level masking techniques, including conditional and dynamic masking patterns, focus on the biological importance of specific nucleotides and aim to simulate natural variability in RNA sequences. The correlation between sequence-level and structure-level masking patterns indicates a relatively strong positive linear relationship. This means that as masking increases in the sequence, masking in the structure also increases linearly and directly. This emphasizes the importance of considering both aspects in the proposed model. The Pearson correlation coefficient $\rho_{\text{seq, struct}}$ is calculated as follows:

$$\rho_{\text{seq, struct}} = \frac{\text{cov}(\text{seq}_D, \text{struct}_D)}{\sigma_{\text{seq}_D} \cdot \sigma_{\text{struct}_D}} \tag{5}$$

Where cov represents covariance, and σ denotes the standard deviation. This mutual dependence indicates the model's ability to predict masked parts based on context and increases its generalization capability.

By using two-level masking and considering the complexities that reflect the real characteristics of mRNA, the proposed model has capabilities such as identifying important biological patterns (such as secondary structural motifs like hairpin, loop, and stem), regulatory elements in untranslated regions, splice sites, codon

usage biases, and mRNA degradation signals. This capability enables RNA structure prediction using only sequences, which is a critical ability when structural data is unavailable. This feature reflects the proposed model's potential in improving the examination of mRNA sequences and structures and helps gain deeper insights into their functional significance in various biological scenarios.

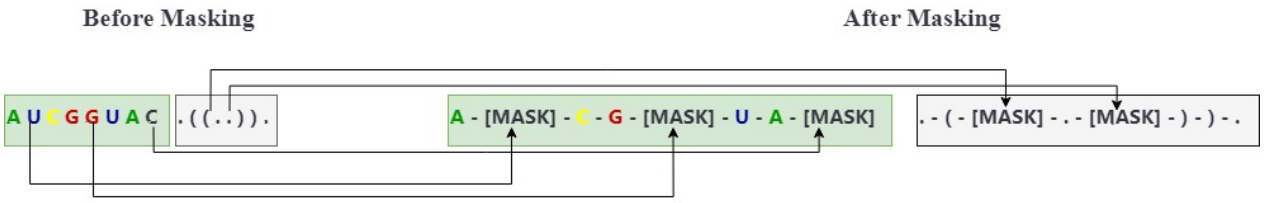


Figure 15: An example of the two-level masking process applied to an mRNA sequence.

Figure 15 shows the two-level masking process applied to a sample mRNA sequence, demonstrating the model's approach to mimicking natural variability in RNA sequences.

Conditional Masking

This technique introduces dynamic masking based on nucleotide types and enables conditional masking to match different molecular structures and functions. Specifically, conditional masking focuses on nucleotides such as guanine (G), which plays a key role in the structural stability and functionality of RNA. Given the biological significance of guanine and its diversity in RNA sequences, this method offers a more realistic simulation of the natural diversity found in RNA sequences.

To formalize this process, let $P(s_i)$ represent the masking probability for a nucleotide s_i, where $s_i \in \{G, A, C, U\}$. Given the biological significance of guanine, its masking probability is set higher than that of the other nucleotides, which are assigned equal probabilities. Mathematically, this is expressed as:

$$P(s_i) = \begin{cases} p_G & \text{if } s_i = G, \\ p & \text{if } s_i \in \{A, C, U\}, \end{cases} \quad \text{where} \quad p_G > p$$

For each nucleotide s_i at position i in the sequence, a random number $r_i \sim \text{Uniform}(0,1)$ is generated. The nucleotide is masked if $r_i < P(s_i)$. This selective masking, particularly for guanine, enhances the StructmRNA model's ability to simulate guanine mutations and their effects on RNA structure and function. By prioritizing guanine, the model better captures its impact on gene regulatory regions and telomeres, improving the accuracy and realism of RNA structure simulations.

3.6 Data Preparation and Preprocessing for StructmRNA

In the process of applying two-level masking functions to the RNA dataset, two important data columns are produced: 'masked_sequence' and 'masked_structure'.
These columns are essential as they contain the original RNA sequences and structures that have been modified by the proposed masking method. This modification is crucial for simulating scenarios where specific

nucleotides or structural elements are unknown or missing, thereby providing a more realistic training environment for the proposed deep learning model. These two masking methods operate simultaneously, and the masking token used for both is identical.

As previously mentioned, this research uses the BERT tokenizer mechanism to efficiently process the modified data. This tokenizer is fundamental to the study as it maps RNA sequences to a format compatible with the BERT-based deep learning model. More precisely, the tokenizer plays a key role in converting sequences and structures into tokenized forms, which are essential for accurate learning and prediction by the model.

To complement the tokenizer, a custom PyTorch Dataset class named 'RNADataset' has been developed. This class generates the **mRNA dataset** in PyTorch and is designed to be compatible with the unique aspects of mRNA data, particularly in managing and efficiently accessing data during the training phase. The 'RNADataset' class not only simplifies the management of masked sequences and structures but also ensures that the data aligns seamlessly with the needs of the BERT model.

To optimize the training process, we employ a component called 'DataLoader', which is essential in PyTorch for batch processing. The proposed DataLoader is equipped with a custom function called 'collate'. This function is specifically designed to handle the complexities of the dataset used in this research and is capable of batch processing RNA sequences and structures.

Tokenizer Configuration

The proposed tokenization method is tailored to RNA sequence and structure data for deep learning applications. This approach involves converting each nucleotide (**A**, **C**, **G**, **U**) and structural symbols ((,), **.**) in RNA sequences and their corresponding structural forms into unique numerical identifiers. This transformation is facilitated by a dictionary called token2int, which includes a unique identifier for the token [MASK].
The [MASK] token plays a crucial role in training the proposed model, as it directs the model towards content-based predictions by masking and hiding tokens. By using the custom tokenization method, the proposed approach bridges the gap between complex biological sequences and the computational framework of transformer-based models, ensuring that the precise information in RNA sequences and structures is preserved and effectively used throughout the model training.

A summary of the parameters and hyperparameters for the proposed model is shown in Table 1.

3.7 Data Augmentation with GAN

The limitation of datasets significantly hinders the training of powerful predictive models. To address this issue, StructmRNA introduces a data augmentation strategy that utilizes GANs. GANs are known for their ability to generate synthetic data that closely resemble real samples. In this context, StructmRNA trains a Generative Adversarial Network (GAN) on real mRNA sequences, enabling the generation of new mRNA sequences. This, in turn, allows the model to be trained with a more diverse set of data.

The integration of real data with synthetic data generated by GANs in StructmRNA is important from two perspectives. First, it directly addresses the issue of data scarcity in bioinformatics and provides a mechanism

Table 1: Summary of Parameters and Hyperparameters for the Training Phase of StructmRNA Model

Description	Value
Vocabulary Size	800
Hidden Layer Size	128
Number of Hidden Layers	8
Number of Attention Heads	8
Intermediate Layer Size	500
Training Batch Size	16
Validation Batch Size	16
Initial Learning Rate (AdamW)	1×10^{-5}
Max Learning Rate (OneCycleLR)	1×10^{-4}
Number of Training Epochs	50
Masking Probability	25%
Optimizer	AdamW
Loss Function	CrossEntropyLoss
Early Stopping Patience	5
Mask Token	`[MASK]`

for data augmentation. The second, and perhaps more significant point, is that it evaluates the potential of synthetic sequences in advancing biological research.

Table 2: Parameters and hyperparameters of the GAN model

Parameter	**Description/Value**
Batch Size	32
Epochs	100
LR	0.0002
Betas	(0.5, 0.999)/(0.9, 0.999)
Dropout	0.7 (Gen), 0.2 (PosEnc)
d_model	1024 (Disc), Var (Gen)
Transformer Layers	10 (Gen), 4 (Disc)
Heads	64 (Gen), 4 (Disc)
CNN Out Channels	Var (Gen)
CNN Kernel Size	3
PosEnc Max Len	107
Optimizer	Adam
Loss Function	BCEWithLogitsLoss and Custom
Scheduler	ReduceLROnPlateau, factor=0.1, patience=10
Weight Init	Xavier, Kaiming, Zero
Noise Std Dev	0.2
Lambda_fm	0.1
Lambda_identity	0.00005
Clip Grad Norm	1.0

Using GAN-generated data in StructmRNA requires considerations such as ensuring the biological validity of the generated sequences. Therefore, this study relies on precise validation methods to ensure that the generated sequences are not only statistically accurate but also biologically valid [67]. By ”statistically accurate,” we mean that the synthetic sequences generated by GAN should correctly reflect statistical features

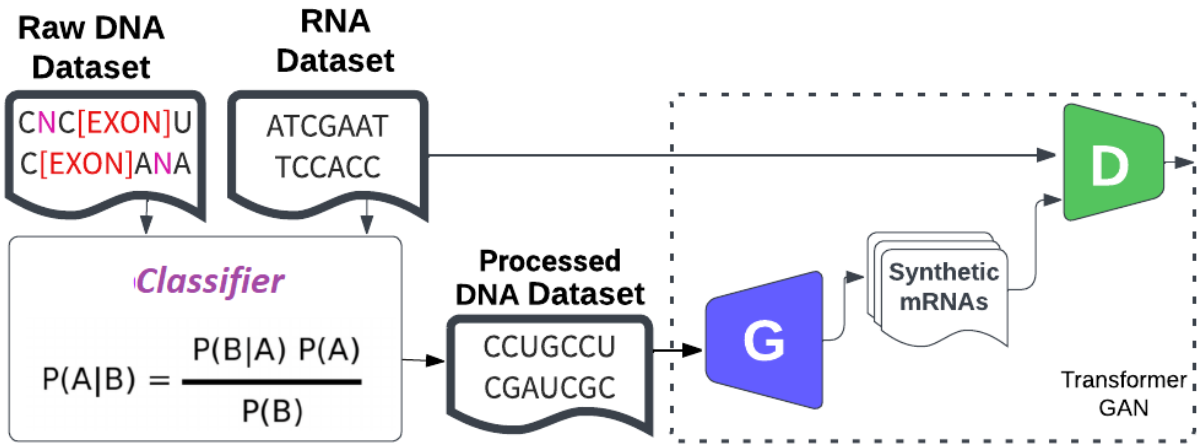


Figure 16: A workflow diagram illustrating the sequence augmentation and mRNA structure process for the Structm-RNA model. This diagram shows the stages of training an mRNA classifier, executing it on the training set, and generating synthetic sequences.

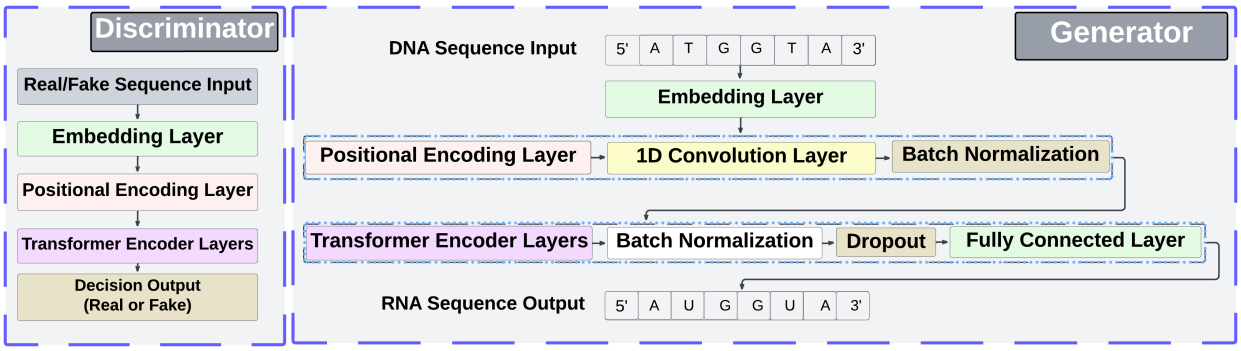


Figure 17: Generator and discriminator architecture of the Transformer-based Generative Adversarial Network for generating synthetic mRNA sequences.

like nucleotide composition, sequence length, and distribution patterns observed in real mRNA sequences. This ensures that the synthetic sequences are computationally and probabilistically representative of natural data and will be useful for further analyses or modeling [2] [84].

Figure 16 illustrates the comprehensive process of integrating GAN-generated data for use in the Structm-RNA model. Additionally, Figure 17 provides an overview of the Generator and Discriminator architecture in the Transformer-based GAN framework specifically designed for generating synthetic mRNA sequences. Table 17 shows the parameters and hyperparameters obtained after model optimization.

The use of GAN-generated data can lead to significant advancements in bioinformatics. By combining machine learning with the needs of biological research, this approach could aid in the development of treatments, vaccine research, and the study of mRNA. In the next chapter, we will explore these results through experiments.

3.8 Conclusion

In this chapter, a new model called StructmRNA was introduced for predicting the degradation of mRNA sequences. This model was designed using deep learning techniques and utilizes mRNA structural and sequence data. To enhance performance and achieve faster convergence, a Generative Adversarial Network (GAN) was employed to augment the training data.

One of the main strengths of the proposed model is the two-level masking process, which combines structural and sequence information to create more accurate embeddings for mRNA sequences. This two-level approach enables the model to not only understand sequence patterns but also to capture important structural motifs and the complex relationships between nucleotides. This is crucial for more accurate predictions of mRNA degradation.

Furthermore, conditional masking based on nucleotide type helps the model realistically simulate the natural diversity of RNA sequences. This method is particularly effective for guanine, which plays an important role in the stability and function of mRNA.

In addition, the use of GANs for data augmentation facilitates the generation of synthetic mRNA sequences, which can enrich the training datasets. This strategy is especially useful in addressing the challenge of a limited number of labeled datasets in bioinformatics. The sequences generated by GANs not only increase the volume of training data but also improve the model's ability to generalize by injecting noise into the model.

Overall, this research presents a comprehensive process for preparing and preprocessing data for the StructmRNA model. This process includes advanced techniques for masking, tokenization, and data management, which contribute to optimizing the model's training and prediction processes.

4 Experiments

4.1 Introduction

In this chapter, we review the results of the proposed models in three separate sections (4.2 to 4.4). This review includes the techniques used for protein networks, embedding models for mRNA embedding, and finally the results of the StructmRNA model in analyzing mRNA data and its application in mRNA degradation prediction.

In each subsection, we will compare the results of implementing the proposed models with other existing models and, based on the evaluation criteria introduced later, analyze the performance of the proposed models. It should be noted that StructmRNA, as the proposed method in this research, is presented for mRNA degradation prediction. However, before that, we will explain the experiments related to IsoGloVe and the mRNA embedding models to describe the process followed in this research to present the new method, StructmRNA. This will give us a deeper understanding of the effectiveness of StructmRNA in degradation prediction and will base the comparison of this model with others on solid grounds derived from the IsoGloVe experiments and mRNA embedding models. The main goal of this chapter is to provide a comprehensive and precise analysis of the performance of the introduced models under various conditions and to investigate their applications, including mRNA degradation prediction, to achieve the best results in biological data analysis.

4.2 Experiments Related to IsoGloVe

4.2.1 IsoGloVe Settings

The IsoGloVe experiments involve different embedding methods, each conducted with specific parameter settings. The HOPE [1] method uses a decay factor of 0.01. The GF [2] method uses a regularization parameter of 1 and a learning rate of 10^{-4}. It should be noted that these parameters in this research are set to optimize the models for better embeddings of the graph structure.

Method Explanation: Each embedding method uses different techniques for learning relationships between nodes in the graph. HOPE is a matrix-based method [3] and is an extended version of the Singular Value Decomposition (SVD) method. The GF method uses dimensionality reduction to represent nodes in a vector space. The node2vec model uses random walk-based techniques to learn vectors, and the walk length in this model is set to 50. IsoGloVe is inspired by a GloVe-based method that learns co-occurrence patterns in random walks. Both node2vec and IsoGloVe have a walk length of 50 [4], with node2vec set to 10 to better model close nodes. The dimensions of the pre-trained vectors are set to 100 for all methods. These dimensions were chosen considering a balance between vector accuracy and computational time. However, methods like LLE [5] and LE do not perform well in large-scale protein-protein interaction (PPI) networks,

[1] Higher-Order Proximity Embedding
[2] Graph Factorization
[3] Similarity matrix
[4] Context size
[5] Locally Linear Embedding

such as the tissue PPI dataset [1] due to high computational complexity (($O(|E|d^2)$)). Some eigenvectors do not converge for the tissue PPI graph, making node classification impossible for these datasets. This issue arises from the computational complexity of eigenvectors in large graphs, which is seen in methods like LLE and LE.

Parameter Explanation: Walk length and context size are two important parameters in learning vectors. Walk length refers to the number of steps taken during each random walk. This parameter directly affects the graph coverage and the model's ability to learn interaction patterns between nodes. Increasing the walk length can help learn relationships over larger distances in the graph but may reduce focus on local structures. Context size refers to the number of nodes considered as near neighbors during each step of the walk and impacts the accuracy of local vectors.

Software and Hardware Version: The IsoGloVe model was implemented using PyTorch 1.11, and the experiments were conducted on two Intel(R) Xeon(R) CPU X5660 @ 2.80GHz processors with 16 GB memory. Although this hardware is sufficient for processing large networks, some methods still face convergence issues due to their high computational complexity. Overall, parameters such as walk length and vector dimensions were chosen to ensure the models' maximum efficiency while avoiding time complexity and computational limitations.

4.2.2 Evaluation Metrics for IsoGloVe Experiments

The proposed IsoGloVe method was evaluated on various PPI networks [106], which are summarized in Table 3. By displaying PPI networks in geodesic space, researchers can use graph-based techniques to analyze protein interactions and uncover hidden biological mechanisms. In this study, IsoGloVe is evaluated in three applications: graph reconstruction, network visualization, and node classification. For graph reconstruction, we reconstruct and rank node vector representations based on their proximity. We then calculate the reconstruction accuracy by predicting the top K neighbors as edges. Additionally, since graph embeddings consider the graph structure, they can be useful for node classification.

Therefore, in this study, we compare the quality of the vector embeddings by using them as node features for classification. Node features are given to a Leave-one-out cross-validation classifier. The classifiers used in this study include: Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Gaussian Naive Bayes (NB) [128]. We chose these methods due to their strong performance in handling high-dimensional feature spaces and their widespread use in graph-based classification tasks. These algorithms offer a good balance between accuracy, interpretability, and computational efficiency. Although other classification techniques such as rough set-based methods [10] are also available and have shown promise in some biological data analysis scenarios, we focused on the selected classifiers for their robustness, ease of implementation, and suitability for evaluating the quality of learned embeddings. Moreover, visualizing the learned vectors from the graph helps in better understanding the topological features of the network. IsoGloVe learns a 100-dimensional embedding and feeds it into the t-Distributed Stochastic Neighbor Embedding (t-SNE) [52] tool to reduce the vector dimensions to two and visualize nodes in a 2D space. To

[1] Refers to data where protein-protein interactions are studied in different biological tissues.

visualize the network, 25% of the yeast PPI nodes are randomly sampled.

4.2.3 Datasets Used in IsoGloVe Experiments

In this section, the datasets used for evaluating IsoGloVe are described. The datasets include three PPI networks, each of which is detailed in Table 3.

Table 3: Detail information about PPI networks used in this study.

Datasets Name	Edges	Nodes	Labeled
Tissue PPI network	1,612,348	56,944	Yes
Human PPI network	1,062,675	6,526	No
Yeast PPI network	7,182	2,361	Yes

These datasets were used as input for the IsoGloVe model to generate node (protein) vector embeddings. These vectors model the network structure and protein interactions and are used for graph reconstruction, node classification, and visualization of PPI networks.

4.2.4 IsoGloVe Experiments

In this study, we performed Kruskal-Wallis statistical tests to evaluate the performance of the models IsoGloVe, node2vec, GF, and HOPE on all datasets.

The H statistic [1] is 96.8 (3, N = 44), and the p-value is 0.029. We also evaluated the impact of vector dimensions on node classification with Naive Bayes and graph reconstruction in the tissue PPI network. Table 4 shows the model performance on the three datasets. The visual results produced in Figure 18 display the features of each embedding method. In Figures 19 (a) and 19 (b), the performance of different embedding models for node classification and graph reconstruction in the tissue PPI network is shown.

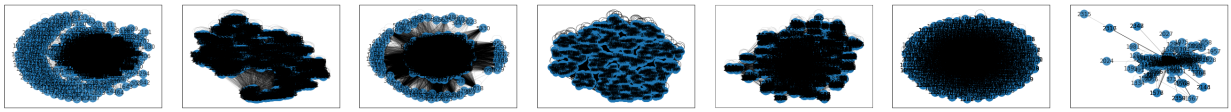
Table 4: Performance comparisons (Model score and MAP) on three PPI

	Y. PPI					T. PPI					H. PPI
	GR	KNN	SVM	RF	NB	GR	KNN	SVM	RF	NB	GR
IsoGloVe	**0.39**	0.20	0.35	**0.35**	**0.55**	0.006	**0.44**	**0.49**	**0.48**	0.41	**0.031**
node2vec	0.30	**0.25**	0.35	0.23	0.44	0.002	0.40	**0.49**	0.37	0.27	0.027
GloVe	0.30	0.24	0.34	0.23	0.44	0.001	0.40	0.38	0.37	0.27	0.025
GF	0.06	0.18	**0.36**	0.20	0.36	0.004	0.43	**0.49**	0.44	**0.49**	0.029
LLE	0.01	0.20	0.35	0.31	0.07	-	-	-	-	-	0.029
LE	0.03	0.23	0.35	0.32	0.09	-	-	-	-	-	0.026
HOPE	0.01	0.22	**0.36**	0.19	0.05	**0.04**	0.06	0.11	0.11	0.018	0.029

4.2.5 Results and Discussion

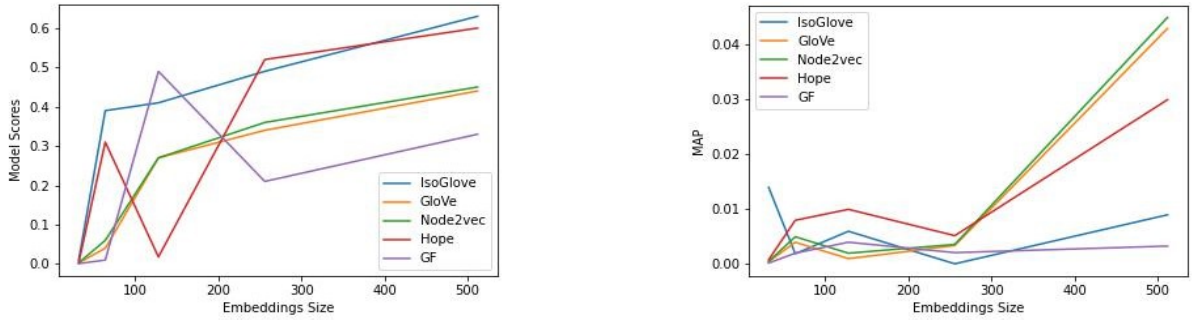
Based on the results of the experiments conducted for graph reconstruction, the average accuracy of IsoGloVe outperforms all baseline models across all datasets. This model has shown better performance in the yeast

[1] H is the statistic used in the Kruskal-Wallis test to examine significant differences between several models. If the p-value is less than 0.05, it indicates a significant difference between the models.



(a) IsoGloVe (b) node2vec (c) GloVe (d) GF (e) Hope (f) LE (g) LLE

Figure 18: Visualization of the embeddings for Yeast PPI network.



(a) The model score for Naive Bayes classifier (b) The MAP of graph reconstruction

Figure 19: Assessing different embedding models for Tissue PPI network at different dimensions.

PPI network for all classifiers except for SVM and KNN. Additionally, in the human PPI network, IsoGloVe achieved similar scores to other models for SVM.

It is worth mentioning that the SVM classifier did not perform optimally due to the distribution of the data being in the form of a manifold [1]. This data characteristic makes it difficult to separate classes using a linear boundary. On the other hand, KNN may not perform as desired due to its reliance on the local structure of the data. In a manifold, the local structure can change as the distance from a point increases. Furthermore, IsoGloVe's better performance compared to baseline models in the tissue PPI network is more significant than the results from the two other PPI networks. This outcome can be explained by the different cellular complexities and the size of the tissue PPI network compared to other PPI networks. Visualization of the vector placement for a sample network like the yeast PPI network shows which features each embedding method can retain. IsoGloVe, node2vec, and GloVe cluster structurally equivalent nodes together, while GF, LE, LLE, and HOPE only preserve the population structure of nodes, keeping connected nodes close to each other. Additionally, IsoGloVe has the ability to distinguish between high-degree hubs and central nodes.

Larger dimensions provide more capacity to represent relationships between nodes, but they also require longer training times and have a higher likelihood of overfitting, resulting in poorer model performance on new networks. Moreover, larger dimensions make it harder to interpret and visualize the vectors, and the relationship between dimensions may not be intuitive. The optimal dimensions depend on the specific application and the quantity and quality of the training data. Since IsoGloVe is an embedding method that records relationships between nodes based on patterns of co-occurrence in random walks, the specific dimensions of the vectors used in IsoGloVe are not as crucial as in other methods, though changing the vector

[1] Manifold

size can still impact the quality of the embeddings.

In the experiment related to Table 3, two important observations can be made. First, in both cases, IsoGloVe is stable against changes in vector size. One reason for this is that with more parameters, models overfit to the observed connections and cannot predict class labels. Second, the relative performance of baseline methods depends on the vector dimensions. For graph reconstruction, node2vec outperforms other methods at higher dimensions, while the vectors generated by IsoGloVe score higher for lower dimensions.

4.3 Experiments Related to Embedding Models

4.3.1 Embedding Model Settings

Hyperparameters play a crucial role in the performance of vector learning models. In Word2Vec, the window size, word vector dimensions, and the number of training iterations affect the model's performance. A limited window size restricts the amount of textual information, while a large window size may lead to overfitting. Increasing the vector dimensions improves the quality of the embeddings but also increases the risk of overfitting.

CNN performance is influenced by the number of filters, filter size, stride, and padding. A large number of filters may lead to overfitting, while a small number may result in underfitting [1]. Similarly, a small filter size creates limitations in the receptive field, while a large size increases the risk of overfitting. AGC performance depends on convergence layers in the graph, filters, activation functions [2], and pooling strategies. Choosing the layers and filters should strike a balance between detecting nucleotide relationships and preventing overfitting.

LSTM and GRU performance depends on the number of hidden units, activation function, and dropout rate. Too many hidden units can cause overfitting, while too few may lead to underfitting. The choice of activation function also impacts performance. ELMo's performance is affected by the number of layers, the number of hidden units, activation function, and dropout rate. Too many or too few hidden units can lead to overfitting or underfitting. Choosing the activation function is also important. In BERT, the number of layers, attention heads [3], hidden units, activation function, and dropout rate all impact performance. Too many hidden units and attention heads can cause overfitting, while too few may result in underfitting. Choosing the activation function is also crucial. The embedding quality in VAE depends on hyperparameters such as embedding dimensions, encoder and decoder architecture, reconstruction loss function [4], regularization techniques, and the quality and quantity of the training data. The optimal values depend on the needs and constraints of the problem.

Given the variability of optimal hyperparameters depending on the application and the datasets used, fine-tuning them is essential [108]. Hyperparameter optimization can be done using grid search or random search.

[1] Underfitting
[2] Activation
[3] Attention heads
[4] Reconstruction loss

Table 5: Settings and Hyperparameters Used for Different Models

Model	Hyperparameters	
Word2Vec	Window Size	5
	Vector Dimensions	50
	Number of Training Iterations	10
ELMo	Number of Layers	3
	Hidden Units per Layer	256
	Activation Function	tanh
	Dropout Rate	0.4
	Transformation Layer	sigmoid
LSTM/GRU	Number of Hidden Units	256
	Activation Function	Swish
	Number of Layers	3
	Dropout Rate	0.4
CNN	Kernel Size	3
	Filters per Layer	mean
	Activation Function	ReLU
	Pooling Size	300
VAE	Latent Dimensions	256, 128, 64, 32, and 16.
	Activation Function	LeakyReLU and sigmoid
	Dropout Rate	0.3, 0.2, 0.1
BERT	Vocabulary Size	14
	Number of Hidden Layers	4
	Number of Attention Heads	2
	Attention Dropout Rate	0.5
	Number of Hidden Units	120
	Activation Function	sinh
AGC	Number of Filters	256
	Filter Size	7
	Stride Size	1
	Number of Layers	4
	Dropout Rate (Embedding Layer)	0.6
	Dropout Rate (Other Layers)	0.4

The experimental results in this study, as shown in Table 5, display the optimal hyperparameters determined through grid search.

4.3.2 Evaluation Metrics and Methods for Embedding Models

This section explains the approach of this research for evaluating model interpretability. The following sections provide a deeper understanding of the nature and application of these techniques, examining the architecture and unique features of each model under investigation. At the end of this section, a comprehensive

table (Table 10) is provided, summarizing the key performance metrics for RNA sequence embedding along with their explanations to give an overview of the evaluation methods discussed. **Evaluation of Embedding Quality through Regression for mRNA degradation Prediction**

One of the common methods for evaluating the effectiveness of embedded vectors is to examine their application in a regression problem, which assesses their performance [76]. In these setups, the embedded vectors are used as input features for a regression model aimed at predicting a continuous variable. Model performance metrics are used as indicators of the quality of embedding. This evaluation is effective because it directly measures the ability of embedding methods to capture meaningful relationships in the data and make accurate predictions. The selected regression problem and performance metrics should align with the embedding problem at hand.

In this study, the vectors generated by various models for the mRNA degradation prediction problem are evaluated.

In this problem, the main goal is to predict the target features **reactivity**, **deg_Mg_pH10**, **deg_pH10**, **deg_Mg_50C**, and **deg_50C** for each RNA molecule. Essential information for each RNA molecule includes its sequence, its dot-bracket structure (which represents nucleotides predicted as paired or unpaired), and the type of predicted loop, which describes its structural details.

The importance of this field has significantly increased due to the COVID-19 pandemic. Although efforts to stabilize mRNA vaccines were ongoing before the disease outbreak, the SARS-CoV-2 emergency accelerated this research [88]. The embedding models evaluated in this study are the result of one of the machine learning challenges called *OpenVaccine* on Kaggle. This competition was dedicated to precise measurements for 6043 RNA structures with lengths between 102 and 130 nucleotides, all taken from the RNA design platform RNA Eterna [115].

For comparison, several datasets from the *OpenVaccine* database were used to evaluate vector learning models against RNA sequences. This database contains sequences, structures, and loop information for each RNA, which are essential for generating RNA vectors. The raw data is available in RDAT format [25]. Key inputs include `SHAPE_RYOS_0620`, `RYOS1_NMD_0000`, and `RYOS1_PH10_0000`. The dataset formats for the *OpenVaccine* challenge are available on the official site [29]. Additionally, subresources, scripts, and pretrained models are publicly available [115].

Figure 20 provides an overview of the RNA molecule and the data related to the mRNA degradation prediction problem, illustrating the distribution of data and noise in the dataset, and showing an image of the RNA molecule along with the feature values used for prediction.

In this regression problem, various metrics have been used to evaluate the embedding vectors, which are summarized in Table 6. Metrics such as Mean Squared Error (MSE) or Mean Absolute Error (MAE) provide an effective comparison between different embedding methods and help identify the optimal approach for a specific problem. For evaluating predictions with multiple outputs requiring the prediction of degradation rates under different conditions, the Mean Column Root Mean Squared Error (MCRMSE) is used [24]. This is because RMSE essentially generates multiple values, each corresponding to a column of predictions.

MCRMSE solves this problem by providing a consolidated evaluation metric.

MAE calculates the mean absolute error between actual and predicted data and indicates overall accuracy but does not account for large errors. On the other hand, MSE calculates the mean squared errors and effectively quantifies the mean squared differences [1] between the estimated and actual values [104]. The feature of MSE to emphasize large errors makes it suitable for balancing both large and small errors. Both MAE and MSE have specific uses, and it is often helpful to calculate and compare both for a particular model. MAE provides a simple evaluation of the average distance between actual and predicted data, while MSE focuses on the squared differences between estimates and actual values.

The Pearson Correlation Coefficient, ranging from -1 to 1, measures the relationship between variables. A correlation coefficient of -1 indicates a strong negative relationship, 0 indicates no relationship, and 1 indicates a strong positive relationship. The magnitude of the absolute value of the correlation coefficient indicates the strength of the relationship; in other words, higher values indicate stronger correlations.

Importance of k-mer through Sensitivity Mapping

In bioinformatics, a k-mer refers to consecutive segments of length k in a biological sequence. k-mer attribution techniques are used to find these important segments, which are then employed by neural networks for specific tasks. The sensitivity map for k-mers is represented as a vector whose dimensions match the input sample. In this map, each value represents the impact of a k-mer from the input on the prediction result [83]. High values indicate significant impact, and low values indicate minimal impact. These methods are inspired by pixel attribution approaches and are also referred to as sensitivity maps, saliency maps, or gradient-based attribution methods [51].

To better understand the features identified by the models mentioned, we calculated scores for each part of an input sample. These scores help us create a map that shows how much each part contributes to the final prediction. In this map, high-scoring values indicate nucleotides that are crucial for predicting mRNA.

Activation Patterns in RNA Sequence Interpretation

Activation patterns in the layers of neural networks with RNA input show how the network processes sequences. These patterns indicate which parts of the sequence the network focuses on. In the early layers, simple features such as nucleotide pairs and basic structures are identified. In the deeper layers, more complex information, such as three-dimensional structure and intermolecular relationships, is extracted. Analyzing these changes helps researchers better understand how the model processes and predicts. Of course, interpreting these patterns depends on the model type and the specific task. Therefore, it is essential to understand the model architecture and its input data before interpreting these diagrams [63].

If activation values at specific positions in the RNA sequence are higher than others, it indicates that these positions are more important for the model's prediction. For example, if the RNA sequence contains crucial

[1] Squared differences

Metric	Formula	Description
MCRMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$	Mean RMSE across columns for multiple predictions, e.g., mRNA degradation rates.
MAE	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i-\hat{y}_i\rvert$	Mean absolute error for overall accuracy evaluation.
MSE	$\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$	Mean squared error for balanced error significance.
Pearson Correlation	$\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$	Measures the linear relationship between predicted and actual values.
Saliency Value	$S_i=\frac{\partial f(\mathbf{x})}{\partial x_i}$	Quantifies the importance of sequence positions for model predictions using gradient-based saliency.
Activation Patterns	-	Assesses how the neural network manages RNA sequence vectors.
Cosine Similarity	$\frac{A\cdot B}{\lVert A\rVert\lVert B\rVert}$	Measures similarity between vectors, useful for RNA sequence analysis.
Correlation Coefficient	$\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$	Measures the linear relationship between variables and is valuable for studying biological data correlations.

Table 6: Evaluation Metrics in mRNA degradation Prediction Problem

information for protein synthesis, high activation values at positions corresponding to amino acid residues indicate that the model is focusing on these positions to predict protein functionality.

It should be noted that interpreting activation values is a complex process that depends on the model's specific architecture, the nature of the input data, and the task at hand. A thorough analysis of activation values in the context of the specific problem being studied is essential, and caution should be exercised when making inferences based on activation values.

Model Discrimination Ability through Cosine Similarity of RNA Sequences

The use of cosine similarity matrices plays a fundamental role in discovering relationships and patterns within RNA sequences in this research. These matrices assist in identifying motifs and common subse-

quences that are essential for tasks like mRNA degradation prediction, providing a foundation for embedding-based models such as GRU and LSTM to effectively identify common features. It is important to note that high similarity values in this metric indicate the presence of repetitive motifs. Moreover, similarity matrices help in detecting model over-generalization. Additionally, these matrices act as a reference for assessing similarity metrics and reveal dependencies and repetitions within the dataset. These tools aid in comparing models and play a crucial role in evaluating neural networks as embedding models.

More precisely, the interpretability of embedding models, including Word2Vec, BERT, and ELMo, as well as neural encoder-based models like LSTM, GRU, CNN, and VAE, plays a key role in identifying patterns revealed by cosine similarity matrices. The interpretability of embedding models involves various methods for examining how models respond to different inputs and uncover relationships between data points. Common approaches here include gradient-based methods and visualization techniques such as t-SNE and Principal Component Analysis (PCA).

Furthermore, analyzing the weights of the attention mechanism helps better understand which parts of the data sequence the model focuses on for each input. It is worth mentioning that the choice of the optimal interpretability method depends on the problem, dataset, and nature of the predictions [14].

Additionally, visualizing the predicted features provides a deeper understanding of the interpretability of neural networks. This visualization allows for identifying input patterns that generate the highest activation values in each neural network unit. Although, due to the vast number of units, visualizing features for each of them is not possible, we can visualize features for each layer of these units [6].

Since embedding models represent sequences as low-dimensional vectors, they allow for the visualization and understanding of sequence features. Each embedding method represents different features within a sequence, leading to varying interpretations of the sequence [87].

Concept-Based Explanations for RNA Sequences

Recently, a set of techniques has emerged in sequence analysis that help understand the inner workings of so-called ”black box” models. However, similar to limitations in image processing, where individual pixels have limited interpretability, conventional feature-based approaches in RNA sequence analysis may lack sufficient interpretability [42]. For example, knowing the importance of individual nucleotides in a sequence may not provide much meaningful information. Moreover, as the number of features (such as nucleotides or specific positions in the sequence) increases, these methods may face limitations. These limitations can include computational complexity, reduced accuracy, or even decreased model ability to identify meaningful patterns.

Concept-based approaches offer a solution to tackle these challenges in RNA sequence analysis. A concept here refers to a sequence motif, a structural motif, or a functional element. Although the embedding models in this study were not directly trained with these concepts, they are encoded in the sequence embeddings. Concept-based explanations for RNA sequence vectors are valuable as they enable us to discover hidden patterns and relationships that are not directly apparent in raw sequence data [61].

To achieve these explanations for RNA sequence embeddings, we calculated the distance between them in a two-dimensional embedding space. This distance was used to discover correlations between various features of RNA sequences, such as sequence length, nucleotide composition, and structural motifs. Visualization techniques, such as heat maps, were then created to provide an intuitive view of the interactions between different RNA sequence features.

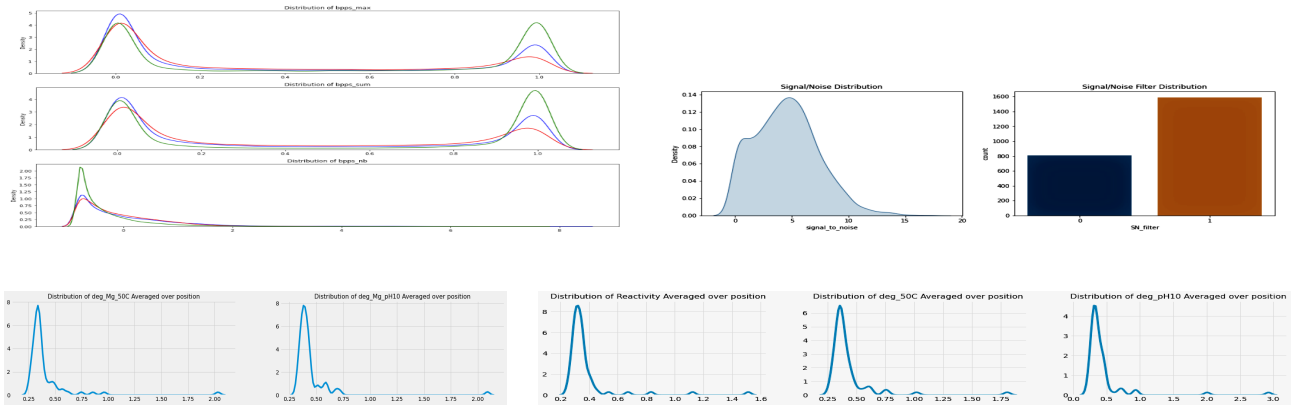
In this context, studies were conducted to uncover relationships between structural motifs present in RNA sequences. This involved calculating correlation coefficients between the distances of pairs of sequences and the criteria representing structural similarities or differences. Furthermore, the relationships between the distances of sequence pairs and the number of nucleotide pairs in each structural motif were examined to determine whether RNA sequences with similar nucleotide pairs tend to cluster in the two-dimensional embedding space. We then calculated the distances between structural motifs to provide a comprehensive view of the relationships encoded in RNA sequences and structural embeddings.

4.3.3 Data Used in the Embedding Models Experiments

The embedding models that have been evaluated are the results of the *OpenVaccine* machine learning challenge on Kaggle. *OpenVaccine* was held with the aim of examining 6043 RNA molecules, whose sequence lengths varied between 102 and 130 nucleotides, and all were obtained from the community-based RNA design platform Eterna [115].

Various datasets from the *OpenVaccine* database have been used for evaluating RNA sequence embedding models. This database includes a comprehensive set of sequences, structures, and loop information for each RNA, which is essential for generating their vector representations [25]. The *OpenVaccine* dataset is available from its official page [29]. Additionally, supplementary resources such as added datasets, scripts, and pretrained models are also available [115].

Figure 20 provides an overview of the RNA molecule and data related to mRNA degradation prediction, data distribution, and noise in the third *OpenVaccine* dataset. It also shows an image of an RNA molecule along with the target feature values associated with the prediction.



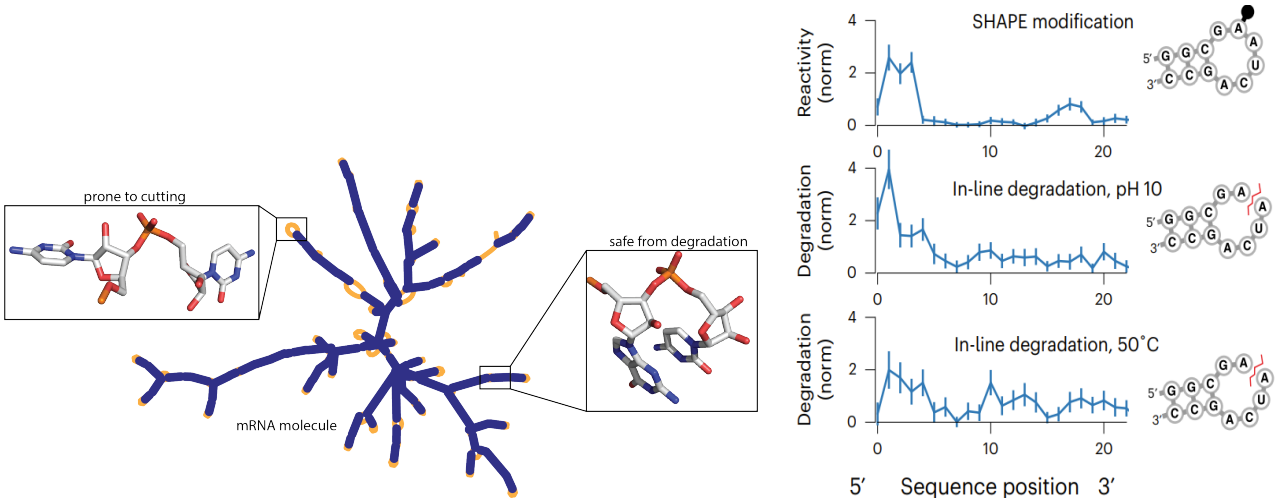


Figure 20: RNA Degradation Prediction: Row 1 and 2: Data distribution and noise in the *OpenVaccine* dataset. Row 3 right side: View of an RNA molecule, left side: Target feature values for prediction [115].

4.3.4 Evaluating the Performance of Embedding Models in Sequence and Structure Analysis

This section examines machine learning models for the analysis of biological sequences, especially RNA, and evaluates the performance of the models from various perspectives.

4.3.5 Analysis of Performance Metrics: MCRMSE, MAE, MSE, and Pearson Correlation Coefficient

The analysis of embedding techniques used for the mRNA degradation prediction problem reveals complex behaviors among different models and their compatibility with the degradation issue. As shown in Figure 4.3.5, the Word2Vec model fails to fit the training data effectively. This is because the Pearson correlation coefficient for the training set increases only slightly, and the validation correlation plot, although showing higher values than the training set, does not reach an acceptable level.

Additionally, the chart comparing predicted labels and actual values for the 5 target features shows significant prediction discrepancies, in contrast to other models, which cluster points diagonally. These results suggest that the Word2Vec model does not perform well in generalizing across most target features.

On the other hand, the ELMo model shows a significant improvement over other models. It efficiently encodes the structure of the training data, with continuous improvements in the Pearson correlation coefficient for both the training and validation datasets. Furthermore, substantial improvements in all error metrics indicate its adaptability to input sequences. However, this model has some weaknesses in learning vector representations. In particular, the predicted value for the target feature **deg_Mg_50C** shows the greatest discrepancy from the actual value, indicating challenges related to this feature.

LSTM and GRU models exhibit significant capability in learning from training data but are prone to overfitting, as shown in their plots. This overfitting becomes evident after several epochs, where the correlation in the training set exceeds that in the validation set. Although the 2D plots of these models show good performance in predicting most target features, the predicted values for the target features **deg_Mg_pH10** and

deg_Mg_50C appear as outliers, highlighting the complexity of these features.

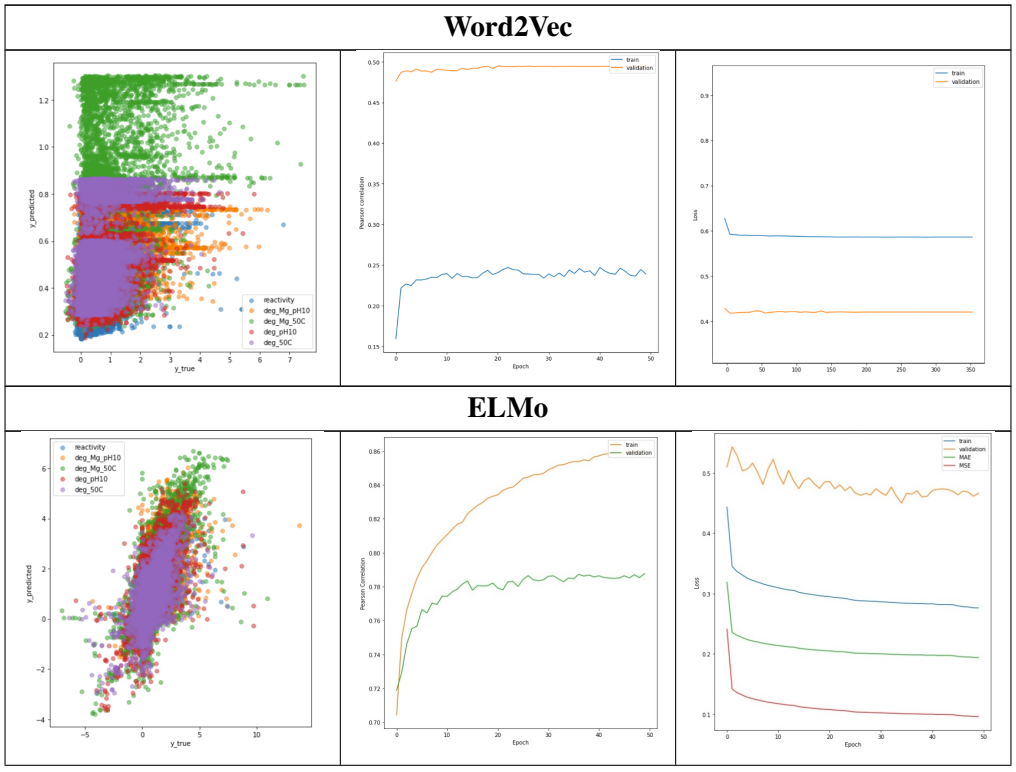
The CNN model shows a unique plot and demonstrates continuous learning with a steady increase in correlation. However, what stands out is the higher correlation among the validation data results, which suggests underfitting. The notable reduction in training set error over several epochs likely indicates progress in learning at that stage.

The VAE model, due to its extremely high MAE and MSE error values, is not suitable for embedding this type of sequence data. The formation of two clusters in its 2D plot suggests that it either recognized two distinct data types or misinterpreted the problem altogether.

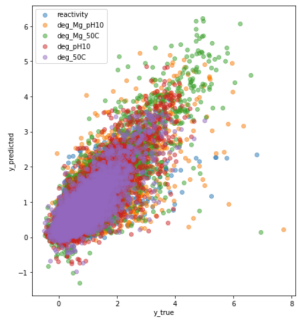
The BERT model performs well in vector representation of sequences. It shows its high-quality learning throughout the epochs, with significant reductions in prediction error values. However, the predicted values for the target feature **deg_Mg_50C** appear to be challenging.

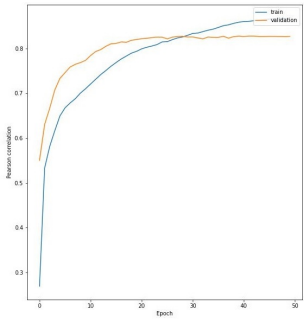
Finally, the AGC model stands out due to its balanced performance. The steady increase in Pearson correlation coefficients for both the training and validation sets, along with a reduction in error metrics, makes it a suitable model for this problem. The predicted values for the target features are predominantly aligned with y=x, indicating the efficiency of this model.

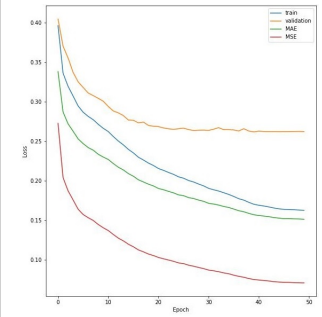
In summary, the target features **deg_Mg_50C** and **deg_Mg_pH10** have appeared as challenging features across most models, likely reflecting their inherent complexity. While some models like LSTM and GRU are prone to overfitting, others like CNN seem to be underfitting or are unsuitable for this specific problem. The ELMo and AGC models, with their balanced performance, seem very promising for future research.



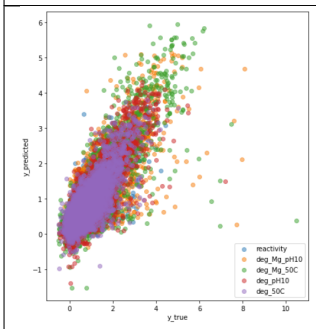
LSTM

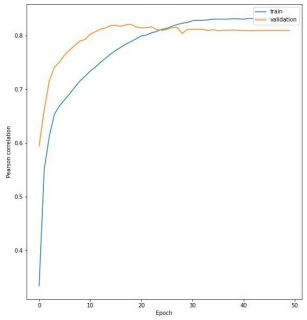


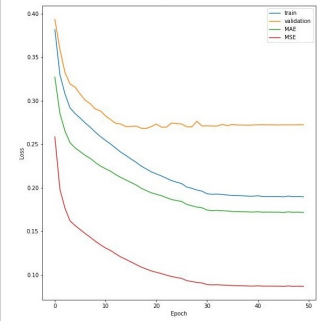




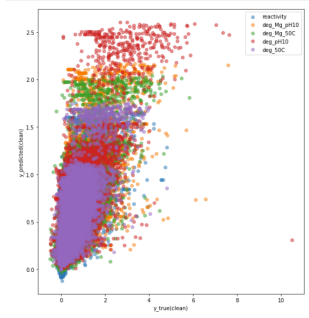
GRU

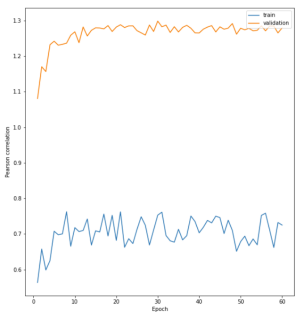


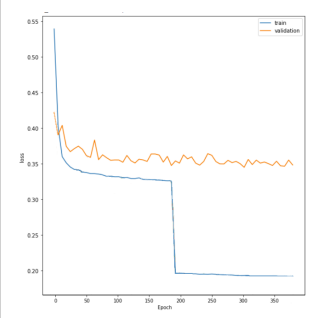




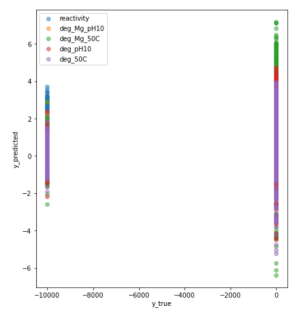
CNN

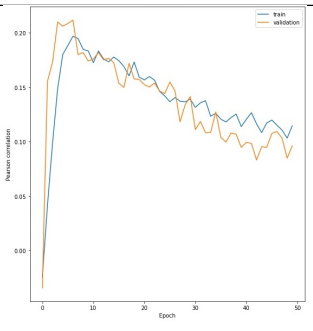


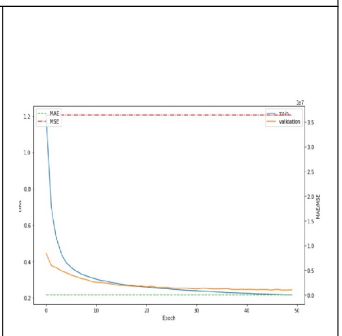




VAE







BERT

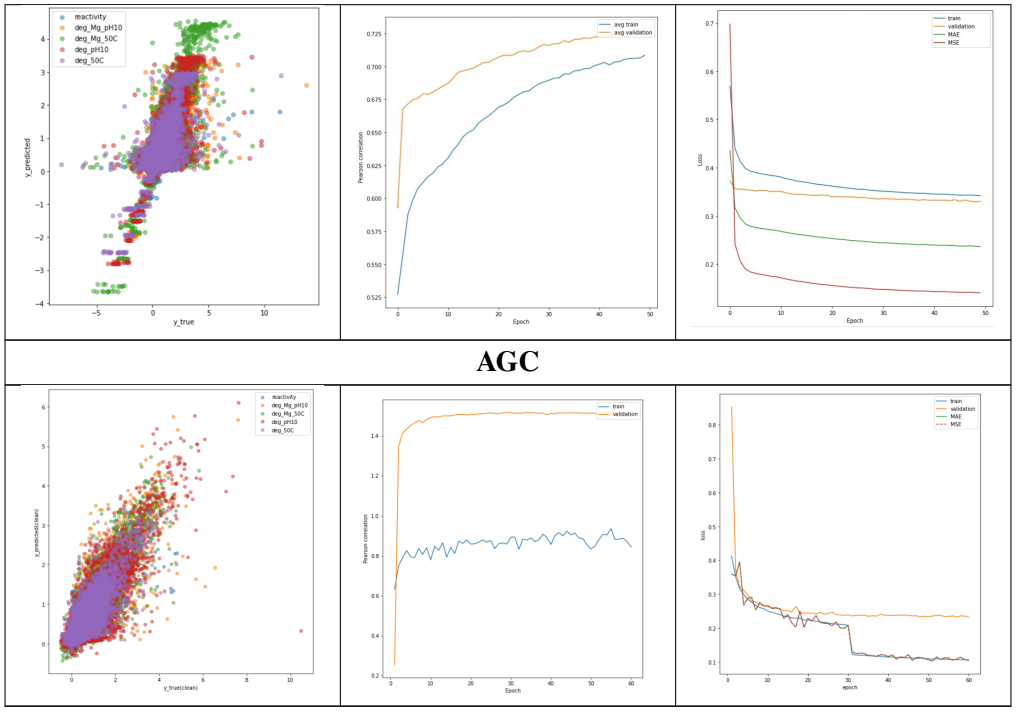


Table 7: Performance of different sequence embedding models in predicting mRNA degradation. From right to left: The first column for each model illustrates the two-dimensional plot of predicted versus actual values for five target features (**reactivity**, **deg_Mg_pH10**, **deg_pH10**, **deg_Mg_50C**, and **deg_50C**). The second column presents the average Pearson correlation coefficient, and the third column shows the mean errors.

4.3.6 Saliency Pattern Analysis

In the analysis of different embedding models and their interaction with RNA sequences, distinct patterns and behaviors emerge based on the respective architectures. The analysis of these patterns is essential to understand the behavior and potential applications of these models in sequence and structure analysis of RNA. Figure 21 shows the saliency patterns for each of the embedding models.

The Word2Vec model chart for the target features **reactivity**, **deg_Mg_pH10**, and **deg_pH10** shows uniform values close to zero. This indicates the inability of this model to identify distinct patterns in RNA sequences for predicting these three target features. In contrast, significant saliency values are observed for the features **deg_Mg_50C** and **deg_50C** at the beginning and end of the sequence. This suggests the identification of distinct structural or functional patterns in these sequence sections.

The LSTM and GRU models, despite their ability to capture long-term dependencies, show similar saliency values to Word2Vec for the features **reactivity**, **deg_Mg_pH10**, and **deg_pH10**. However, for the two target features **deg_Mg_50C** and **deg_50C**, these models place greater importance on the first half of the sequence. This may indicate that these models deem the sequence's end less significant for these features or

that they have encountered the vanishing gradient problem, meaning that in longer sequences, the influence of initial sequence elements diminishes over time.

The ELMo model, known as a context-based and neighborhood embedding method, has a high capacity for sequence processing. However, due to its neural network structure, the saliency map interpretation results of this model closely resemble those of the LSTM and GRU models.

The CNN model operates differently. This model assigns nearly equal and high importance to the entire RNA sequence when predicting the features **deg_Mg_pH10**, **deg_pH10**, and **deg_Mg_50C**. This behavior indicates the model's ability to detect patterns regardless of their position within the sequence. However, it is notable that despite uniform importance across the sequence, there is a greater emphasis on the sequence's terminal sections. This points to the presence of specific secondary structures or distinct motifs at the sequence's ends.

The VAE model shows significant importance to all positions along the sequence for certain target features due to its generative nature. The main purpose of using embeddings from this model in the degradation task is to leverage its ability to discover data distributions and understand a wide range of features present in the RNA sequence.

BERT, which uses an attention mechanism-based architecture, is capable of identifying dependencies across longer distances within the sequences. The saliency values of this model confirm that it not only looks at individual sequence components but also considers the role of each section within the entire sequence in relation to the problem being examined. This means the model understands which sections are important for the final outcome, with darker regions representing parts where key information for predicting the target features is hidden.

Finally, the results from the AGC model resemble those of the CNN model in many respects. AGC considers the entire RNA sequence but places more focus on the sequence's end. This behavior indicates that AGC effectively distinguishes the sequence's terminal section. Additionally, due to its use of attention mechanisms and graph-based data processing, it has uncovered complex relationships in the RNA sequence that other models were unable to identify.

In summary, the interpretations of the saliency maps indicate that while models such as LSTM, GRU, and ELMo focus on specific positions within the sequence, other models like CNN, BERT, and AGC have a more uniform understanding of all sequence components. These saliency maps play a crucial role in the model interpretation process and reveal which parts of the sequences the models consider essential for their predictions.

4.3.7 Analysis of Input and Output Layer Activations [1]

In this section, we analyze and compare the activations of input and output layers for different embedding models. The inputs consist of nucleotide sequences, secondary structure, and RNA loop structure, while the

[1] Input and Output Layer Activations

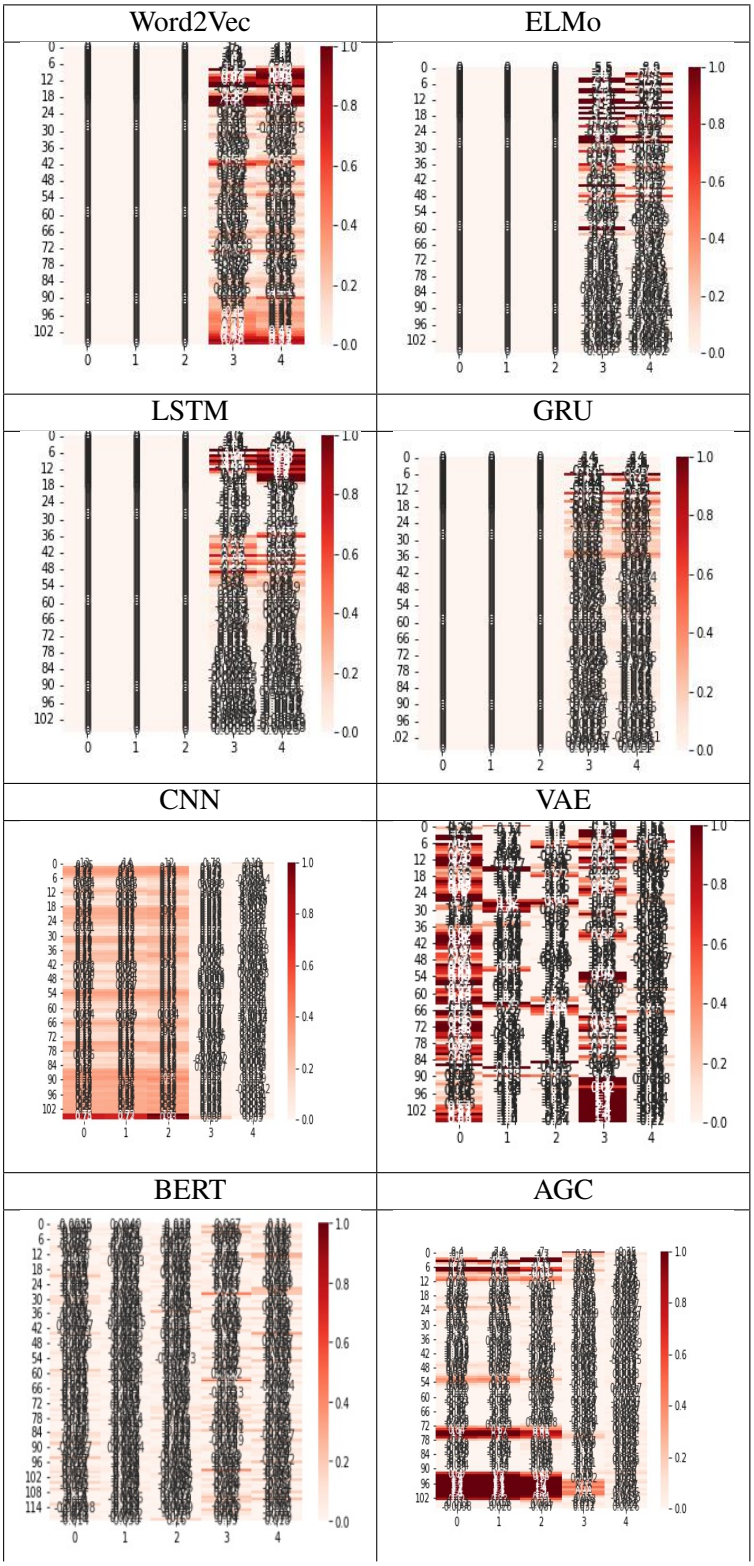


Figure 21: Saliency maps of embedding models for target features and input sequences based on prediction error are presented. The horizontal axis represents each of the target features as follows: `reactivity` = 0, `deg_Mg_pH10` = 1, `deg_pH10` = 2, `deg_Mg_50C` = 3, and `deg_50C` = 4.

outputs correspond to the five target features predicted using vectors obtained from these models.

Figure 22 displays the activations of the first layer for the embedding models. This figure shows how these models identify constant sequences and may reach a saturation state at early positions. Figure 23 shows the activations of the last layer for models applied to RNA sequences. This figure highlights changes in activation patterns and their impact on biological processes and pattern recognition.

We then examine each of the models to more accurately analyze their performance in predicting target features and identifying biological patterns related to RNA sequences.

For the LSTM, GRU, Word2Vec, and ELMo models, the input layer activations stabilize to a constant value after position 90, indicating the ability of these models to recognize sequences with similar patterns or stable structures in the input data. The activation values for RNA structure data remain around 2 after position 90, which points to stable structures or possibly model saturation [1].

The activations for RNA loop structure data fluctuate between 0 and 1, but they also stabilize after position 90. All output activations for GRU, LSTM and ELMo show a downward trend, which represents the shared target features of the input. The activation for the **reactivity** target decreases significantly, while for **deg_pH10** it remains higher. Word2Vec detects local changes with ReLU activations, while LSTM and GRU, using tanh or Sigmoid, recognize longer sequences and result in more stable activations.

The consistent trend across the predictions of these five target features indicates shared information or a strong correlation between these features. Therefore, examining shared layers in multi-task architectures [2] would be beneficial. The stabilization of activations after position 90 indicates model saturation, and further research in this area could focus on optimizing the model for processing these regions.

Analysis of CNN layers reveals diverse activation patterns. RNA sequence activations vary between 0 and 2, with significant increases at positions 40 and 60. RNA structure activations stabilize after position 40, while RNA loop structure data activations fluctuate between 0 and 1. Output activations for all five features show similar trends, ranging from 75.0 to 5.2, peaking at position 35. This indicates the model's ability to identify common features in RNA sequences. The AGC input activations follow a similar trend to CNN, indicating stability in their behavior when encountering different patterns in the input data.
It is worth noting that AGC activations increase significantly toward the end of the sequence positions. This increase may be due to the detection of unique features or changes in the sequence's final section. Interestingly, despite the unique input activation pattern, the output activations of AGC resemble those observed in LSTM, GRU, BERT, and VAE. This suggests that AGC, like the other models, can identify and display crucial information related to target features, despite differences in input activations. This alignment in output activations indicates that AGC, like other models, may effectively learn and embed biological patterns and underlying correlations in the data.

The input activations for VAE fluctuate (0-1) compared to LSTM and GRU, indicating its embedding ability.

[1] Model saturation refers to a state in machine learning models where the model no longer has the capacity to learn or extract new information from the data.

[2] Multi-task

Output activations for all target features start from 75.0 to 2, peak after position 40, and then fluctuate between 0 and 5.1, which refers to similar patterns in the data in the predictions. BERT input activations exhibit diverse patterns across different regions of the sequence. In the first 20 positions, values fluctuate from 3 to 6, decreasing to 0 to 2 between positions 20 and 40. Between positions 40 and 60, values fluctuate between 7 and 12, and after position 60, they approach zero. Output activations for features like **reactivity** start from 75.0 to 2 and decrease to zero, which could indicate saturation or stability in model simulation. This means the model no longer attends to changes in those areas or no longer needs to process these regions.

In summary, models like LSTM and GRU identify constant patterns, with activations remaining stable after position 90. Word2Vec exhibits notable variations in activations, while VAE provides a more diverse representation. BERT effectively identifies sequence variations. The activation patterns suggest that RNA processes are influenced by common factors, indicating the deep learning model's ability to identify biological patterns.

4.3.8 Cosine Similarity Analysis

An ideal similarity matrix displays common motifs or subsequences, which are essential for more detailed analyses, such as mRNA degradation prediction. Figure 24 shows the performance of the various models used as embedding models in this research.

In all heatmaps, a diagonal line is visible, indicating the highest similarity between similar arrays. This value can be used to assess other similarity measures. Blue colors indicate that most arrays have high similarity, usually with values between 7.0 and 1. Yellow spots refer to more specific or less similar sequences in this study.

When considering the relative performance of different models as embedding models, it becomes clear that different models provide different results. For example, the VAE model shows the least similarity between vectors of RNA sequences and identifies most sequences as dissimilar. On the other hand, the ELMO, AGC, and CNN models are almost identical in identifying similar motifs but detect more similar sequences than the VAE. However, they still identify most sequences as dissimilar. The LSTM and GRU models, which are almost functionally similar, also identify more similar motifs in sequences. Finally, the BERT and Word2Vec models perform equally in identifying similar sequences.

These findings emphasize the importance of selecting an appropriate embedding model for RNA sequence analysis tasks, as different models exhibit varying sensitivities to sequence similarity.

In summary, high values of similarity metrics indicate the presence of motifs in RNA sequences. Therefore, vector embedding learning models like GRU and LSTM are considered suitable models for detecting these motifs. It can also be concluded that the degradation prediction problem is prone to overfitting, and the use of a diverse training dataset is essential.

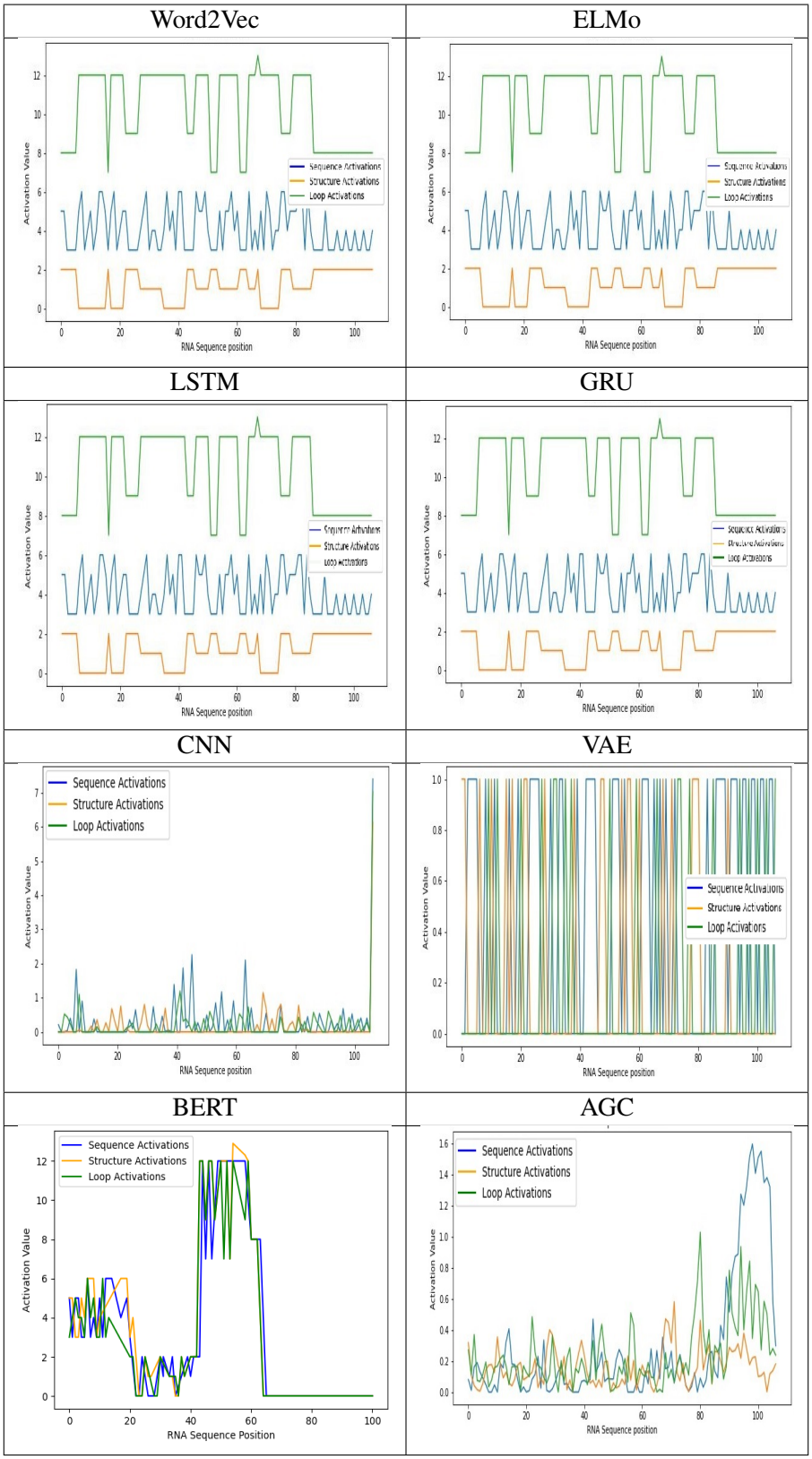


Figure 22: First-layer activations in different embedding models. Activation values are shown for input data including nucleotide sequences, secondary structure, and RNA loop structure.

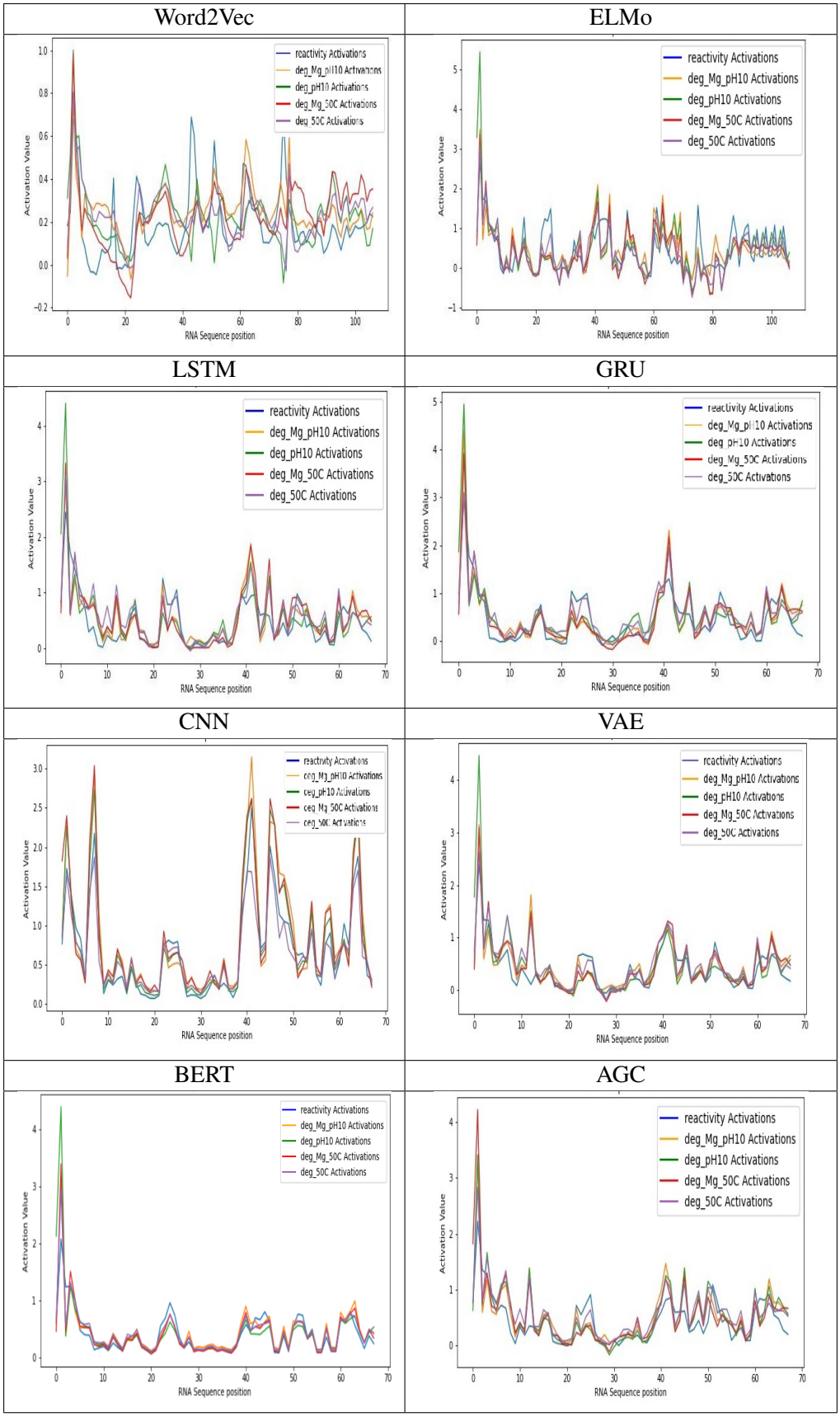


Figure 23: Final-layer activations for different embedding models. Activation values for each of the predicted target features in the final layer are displayed.

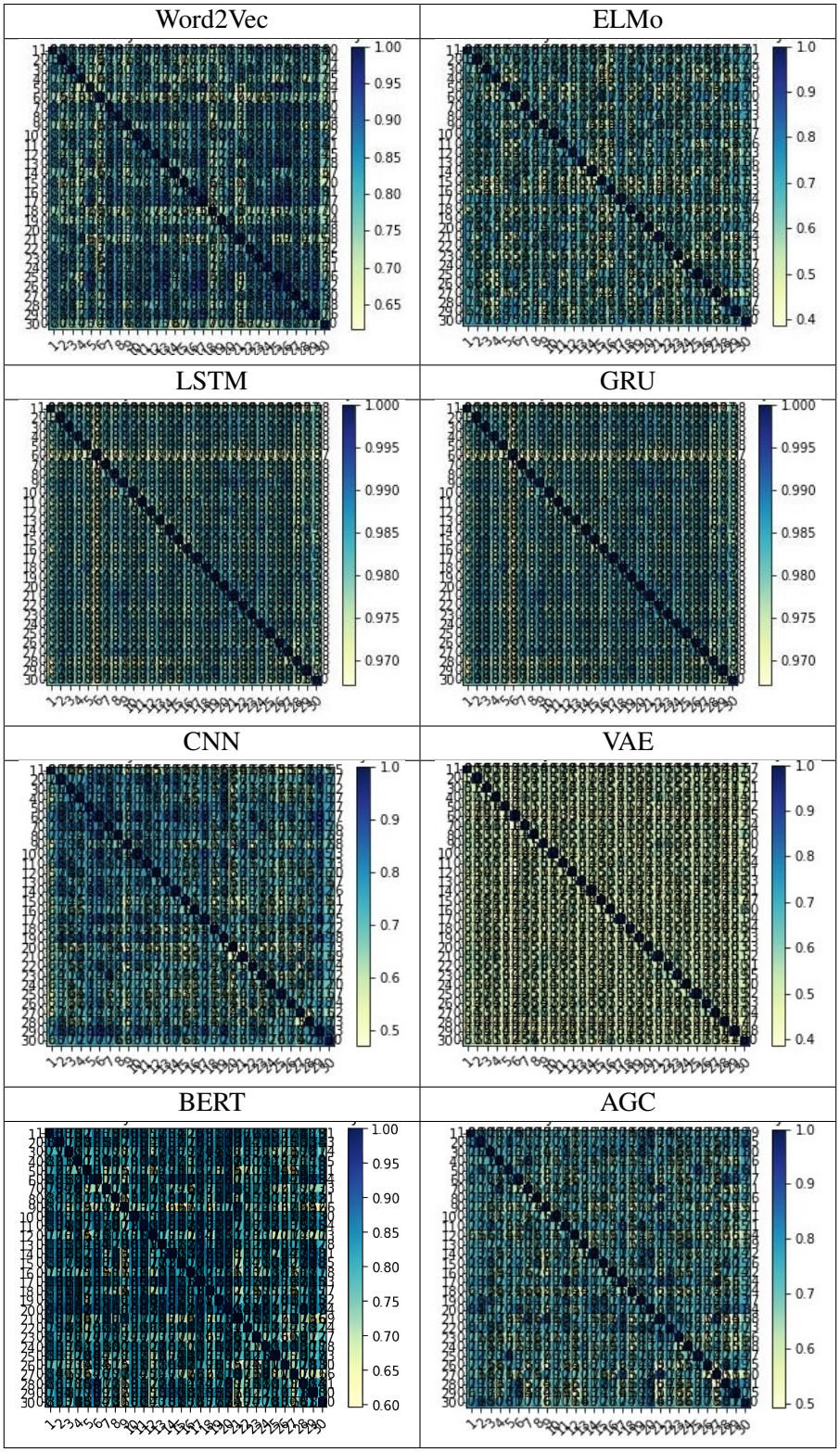


Figure 24: Comparison of cosine similarities between RNA vectors for 30 sampled sequences.

4.3.9 Heatmap of Correlation Analysis: Nucleotide Sequences, Secondary Structures, and Loop Structures

In RNA sequence analysis, finding meaningful relationships between sequences, structures, and various features is important for a better understanding of the RNA molecule's nature. As mentioned earlier, the purpose of these analyses is to examine whether the embeddings generated by different models can adequately represent nucleotide sequences, secondary structures, and loops, and correctly transfer the main features of the initial data. For this, we use various computational methods and tools to measure the relationships between RNA sequences, their structural features, and loop structures.

In this section, we introduce the methods used to derive these relationships and create heatmaps to represent them. The cosine similarity matrix is represented as a square matrix, where each entry indicates the cosine similarity between two RNA vector representations generated by embedding models. Figure 25 displays the comparative performance of different neural networks in this analysis.

Sequence similarity is evaluated by calculating the Levenshtein distance, also known as edit distance, on RNA sequences. This measurement is precisely implemented using the Levenshtein programming tool, which has powerful capabilities for comparing sequences. In the implementation process, we examine all RNA sequence pairs in the dataset and compute their Levenshtein distance. To ensure proper comparison, we normalize the calculated distances by dividing them by the maximum length between the two sequences.

For structural similarity analysis, RNA secondary structure for each sequence is predicted using RNAfold, which is part of the ViennaRNA package. Then, for all sequence pairs, we perform this prediction and evaluate the structural similarity by comparing the obtained structures and normalizing the results.

To measure loop similarity, we first calculate the number of similar loops and divide it by the total number of loops in the loop structure of the sequence. The result of this calculation is a value between 0 and 1, representing the degree of similarity. For this analysis, we have created a matrix that shows the similarities between the RNA sequence, secondary structure, and loop structure embeddings. In each of these matrices, at each intersection point, if the row and column are the same, the correlation is equal to 1 and is colored red.

For example, the intersection of the row and column **cosine-cosine** indicates the similarity of the RNA embedding vector with itself. For other rows and columns, the correlations are displayed with different colors. These colors indicate the intensity of similarity (correlation value) and the direction of the correlation (whether it is positive or negative). In summary, the colors indicate how similar two features are and which direction this similarity tends towards.

The colors in the **cosine** column of the heatmap represent the correlation between the cosine similarity matrix and each of the other similarity matrices. This simplification provides a clear understanding of the interactions and patterns among these matrices in the dataset and facilitates the exploration of relationships and structures within RNA sequences.

Among the models examined, **Word2Vec** shows a moderate positive correlation with sequence-based vector embeddings, indicating its ability to embed RNA sequence information. It also shows weaker positive corre-

lations with structural and loop embeddings, revealing its limited capacity in embedding structural and loop features.

ELMo has been very successful in extracting RNA sequence information. This is demonstrated by its stronger positive correlation with sequence-based vector embeddings compared to **Word2Vec**. Although ELMo also has positive correlations with structural and loop embeddings, these correlations are weaker, suggesting that this model focuses more on sequence information.

Recurrent neural network-based models, such as LSTM and GRU, behave similarly to the **Word2Vec** model in terms of correlations. These recurrent models show moderate positive correlations with sequence vector embeddings but weak correlations with structural and loop vector embeddings, aligning with trends observed in **Word2Vec** and ELMo.

In contrast, the CNN-based model shows a weak correlation with sequence-based vector embeddings compared to other models, indicating its limited capacity to capture RNA sequence information. Additionally, its near-zero correlations with structural and loop embeddings raise questions about its efficiency in capturing structural features and loop aspects of RNA.

The VAE model performs well in this area, showing a strong connection with RNA sequence vector features and a moderate connection with structural and loop features. This indicates that VAE can effectively represent various aspects of RNA sequence and structure. The BERT model shows weak to moderate correlations with sequence, structure, and loop vector features, suggesting a balanced performance. In contrast, the AGC model focuses more on RNA sequence information, with a moderate correlation with sequence vector features, but shows less correlation with structural and loop features.

In general, heatmaps show that the models **VAE**, **GRU**, **LSTM**, **CNN**, and **AGC** perform better in identifying similarities. These models demonstrate a high ability to discover patterns from RNA sequence, structure, and loop structure, emphasizing the importance of selecting the appropriate model in biological problems.

4.3.10 Sensitivity to Hyperparameters

Sensitivity to hyperparameters refers to the impact of the model's hyperparameter settings on its performance. Unlike parameters that are automatically adjusted during the training process, hyperparameters must be specified before the training begins. Proper selection of hyperparameters can play a crucial role in enhancing model performance. In this section, our goal is to examine the stability of embedding methods against hyperparameter changes and analyze the impact of these changes on the performance of the models.

4.3.11 Impact of Vector Dimensions on Model Performance

The dimensions of vectors can affect their quality in various ways. A larger dimension means a larger model and longer training time, but it also provides greater capacity for the model to represent relationships between entities, which may lead to improved accuracy. However, having excessively large dimensions can lead to overfitting, which weakens performance on new datasets. Furthermore, larger dimensions make it more challenging to interpret and visualize the vector embeddings, and the relationship between dimensions

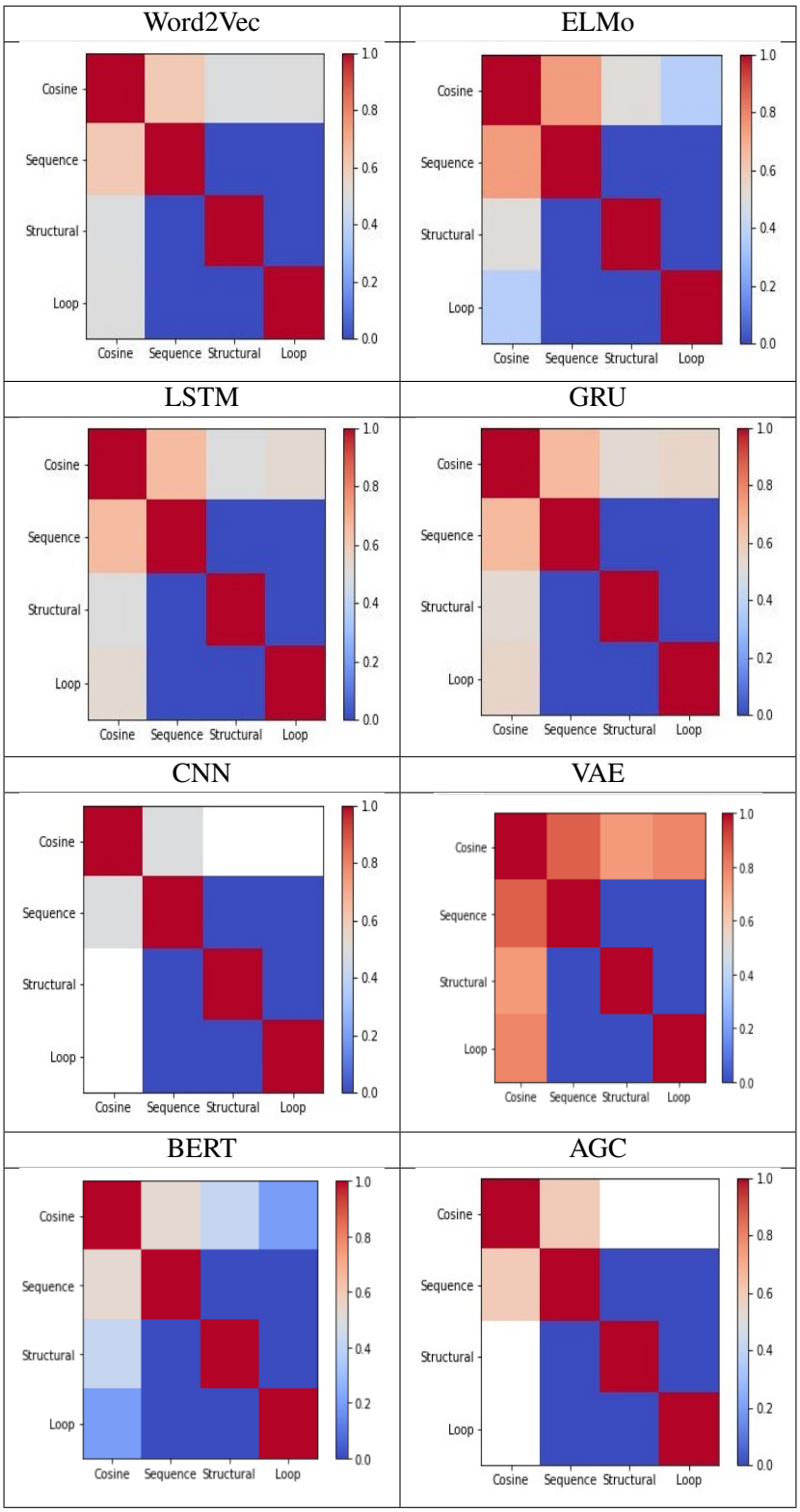


Figure 25: Heatmap of correlation coefficients between RNA vector embeddings.

Table 8: Predicted Model Performance for Different Vector Embedding Lengths. Performance Metrics: MCRMSE (MC.), MAE (MA), MSE (MS.), and Pearson Correlation (P.).

Model	**30-Dimension**				**50-Dimension**				**100-Dimension**			
	MC.	**MA.**	**MS.**	**P.**	**MC.**	**MA.**	**MS.**	**P.**	**MC.**	**MA.**	**MS.**	**P.**
Word2Vec	0.65	0.40	1.05	0.20	0.56	0.35	0.91	0.24	0.50	0.31	0.81	0.27
ELMo	0.32	0.22	0.10	0.77	0.28	0.19	0.09	0.86	0.25	0.17	0.08	0.97
LSTM	0.17	0.16	0.08	0.79	0.15	0.14	0.07	0.88	0.13	0.12	0.06	0.99
GRU	0.21	0.20	0.08	0.79	0.18	0.17	0.07	0.88	0.16	0.15	0.06	0.99
CNN	0.20	0.21	0.21	0.70	0.18	0.19	0.19	0.73	0.16	0.17	0.17	0.77
VAE	0.25	4148	42443	0.05	0.22	3645	36451	0.06	0.20	3314	33483	0.07
BERT	0.46	0.25	0.14	0.68	0.40	0.22	0.12	0.76	0.36	0.20	0.11	0.85
AGC	0.14	0.13	0.13	0.75	0.12	0.11	0.11	0.84	0.11	0.10	0.10	0.94

may become less clear. The optimal dimension depends on the specific application, the amount and quality of the training data, and the computational resources available. Therefore, it is essential to carefully select the vector dimensions when building machine learning models [77].

In some cases, techniques like PCA or SVD are used to reduce the data's dimensionality while preserving important information. Additionally, using regularization techniques such as dropout or early stopping, even with large dimensions, can help prevent overfitting. Ultimately, the appropriate dimension for a specific problem depends on a combination of the data's size and complexity, the available computational resources, and the desired balance between accuracy and interpretability.

Table 8 shows the evaluation results of various models' embeddings for different vector dimensions. This table indicates that increasing the vector size improves the models' performance, which is due to richer vector representations and the ability to embed more complex features from the data. The LSTM and GRU models are significantly superior to the other models due to their capability to process sequential data effectively. The high error rate in VAE suggests that there may be issues such as overfitting or convergence problems during training, requiring further investigation of the training process. However, in this study, 50-dimensional vectors have been used as the output of the embedding model. This decision was mainly influenced by time constraints and computational complexity, and this configuration may not be the optimal solution.

4.3.12 Time Complexity Analysis in mRNA Sequence Representation Models

In this section, a comparison of the time complexity and training duration of various models designed for processing RNA sequence data is conducted. Table 9 presents a comparison of the training time complexity and duration for different vector representation models.

Several key insights emerge from this comparison:

- Models such as LSTM and GRU have relatively low time complexity, with training durations of 4.12 and 7.10 minutes, respectively.
- CNN, with a time complexity of $O(N \cdot F \cdot D^2)$ and a training duration of 4.2 minutes, demonstrates

efficient training.

- More complex models such as VAE, BERT, and ELMo have higher time complexity, resulting in longer training durations.
- AGC, with time complexity related to the number of neighbors, has a training duration of 6.11 minutes.

This comparison helps researchers select models suitable for transforming data into vector representations based on computational resources and required training time. These insights contribute to more precise decision-making in efforts to obtain useful results from RNA data.

Table 9: Comparison of Time Complexity and Training Duration

Model	**Time Complexity**	**Training Time (in minutes)**
Word2Vec	$O(N \cdot T \cdot D)$	71.0
ELMo	$O(N \cdot T \cdot D)$	0.74
LSTM	$O(N \cdot T \cdot D^2)$	4.12
CNN	$O(N \cdot F \cdot D^2)$	4.2
GRU	$O(N \cdot T \cdot D)$	7.10
VAE	$O(N \cdot E \cdot D)$	6.23
BERT	$O(N \cdot S \cdot D^2)$	8.45
AGC	$O(N \cdot D \cdot$ Max Number of Neighbors$)$	6.11

4.3.13 Results and Discussion

Table 10 presents a summary of the comparative analysis conducted in Section 4.3.4. This table highlights various aspects, including the model used, evaluation metrics, reasons for using salient maps, activation patterns in input and output layers, cosine similarity of vectors, correlations (between vectors, sequences, structures, and loops), and vector representation quality.

Models such as ELMo, LSTM, and GRU have demonstrated significant performance and generally provide high-quality embeddings. In contrast, models like Word2Vec and VAE exhibit weaker performance. Other models, such as CNN and BERT, perform adequately, though their results may have some shortcomings. Additionally, despite BERT's strong performance in natural language processing, in certain conditions, it may experience a notable decrease in embedding quality compared to other models, especially when dealing with more complex or out-of-distribution data. Generally, models that can attend to different parts of the data yield better results. Models like BERT and ELMo, which can focus on different sections of the data, perform better in identifying important words or features. These models outperform simpler models due to their specialized attention to specific sequence segments.

4.4 Experiments on StructmRNA

In this study, various models for embedding RNA sequences have been introduced and analyzed in detail in Section 4.3. Some of these models, such as LSTM and GRU, have been recognized as top choices for

Table 10: Summary of Comparative Analysis for RNA Sequence Encoding

Model	Performance Metrics		Saliency	Layer Act.	Cosine Sim.	Correlation			Embedding Quality
	PC	Losses				Seq.	Str.	Loop	
Word2Vec	Limited	Decreases, remains relatively high	Varies, with limited focus on some targets and strong on others.	Captures localized variations with ReLU	Moderate	Relatively high	Moderate	Moderate	Limited
ELMo	High	Significant improvement	Limited saliency for certain targets	Stable input act. at seq's end, declining trend for output	Moderate	Relatively high	Moderate	Relatively weak	Effective
LSTM	High	Significant improvement	Limited saliency for certain targets	Stable input act. at seq's end, declining trend for output	High	Relatively high	Moderate	Moderate	Effective
GRU	High	Significant improvement	Limited saliency for certain targets	Stable input act. at seq's end, declining trend for output	High	Relatively high	Moderate	Moderate	Effective
CNN	High	Sudden reduction	Global importance, emphasis on sequence's end	Diverse act. patterns	Moderate	Moderate	Moderate	Moderate	Effective
BERT	High	Significant reduction	Adaptability through attention to various sequence parts	Diverse, effective interpretation of various sequence regions	Moderate	Moderate	Moderate	Relatively weak	Effective
VAE	Low	High	Broad importance across the entire sequence	Dynamic fluctuations in input layer act.	Limited	Low	Relatively high	Relatively high	Limited
AGC	Balanced	Sudden and significant reduction	Global importance for specific targets, emphasis on sequence's end	Increase on sequence's end, declining trend for output	Moderate	Relatively high	Moderate	Moderate	Effective

embedding biological sequences due to their high capability in processing sequential data. However, these models face challenges such as overfitting. Additionally, models like Word2Vec do not perform well when dealing with the complexity of biological sequences, highlighting the need to balance deep and shallow learning techniques for specific applications. In this section, we compare the StructmRNA model with the aforementioned baseline models.

4.4.1 StructmRNA Experiment Data

The StructmRNA model has been trained using a large dataset of human mRNA sequences, consisting of 46.3 billion nucleotides. This dataset, obtained from the NCBI database [1], provides sequence and structural data with high diversity from various sources.

Additionally, to enhance the generalization capability of the StructmRNA model, we have incorporated data from the *OpenVaccine* project by Stanford University. This subset, which includes sequence and structural information of 2,400 mRNA molecules with a length of 107 nucleotides, contributes to improving prediction accuracy. To ensure efficient processing and enhance model training, all sequences in this study's dataset have been standardized to a fixed length of 107 nucleotides.

Moreover, to generate synthetic mRNA sequences using the proposed GAN model, we utilized DNA sequences from the NCBI database. In total, we examined 125.1 million nucleotides of human DNA extracted from NCBI.

Dataset Source	Sequence	Purpose	Data Availability
NCBI Homo sapiens mRNA	46.3 billion nucleotides	Training the embedding model	https://www.kaggle.com/datasets/spnahali/sequences-ncbi-auto/dataStructmRNA model
NCBI Homo sapiens DNA	125.1 million nucleotides	Training the GAN model	https://www.kaggle.com/datasets/spnahali/rna-seq-strGAN dataset
Synthetic mRNA sequences generated by GAN	300 000	Generated from NCBI DNA sources	https://www.kaggle.com/datasets/spnahali/rna-seq-strGAN dataset
OpenVaccine	2400	Predicting mRNA degradation	https://www.kaggle.com/competitions/stanford-covid-vaccine/dataKaggle competition

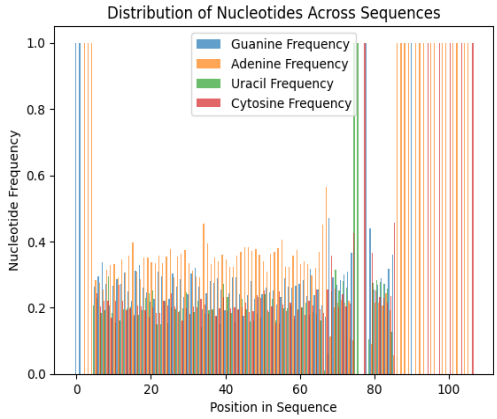
Table 11: Summary of datasets used in this study.

4.4.2 Evaluation Criteria of StructmRNA Experiments

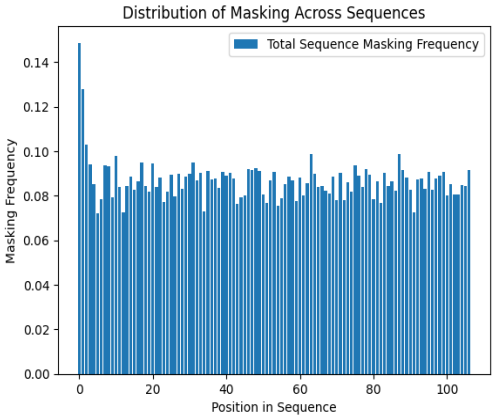
To fairly evaluate the proposed StructmRNA model and compare it with existing baseline models, we utilized datasets introduced as the **Public Dataset** [1] in the Kaggle competition for all models.

The baseline models from the Kaggle competition include ensemble models [2], genetic algorithms, Nullrecurrent, Kazuki2, DegScore-XGBoost, and DegScore [44]. In addition to these models, reference models

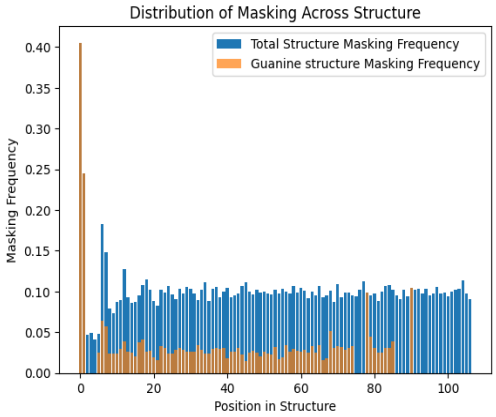
[1] Public Dataset
[2] Ensemble



(a) Nucleotide frequency distribution among sequences.



(b) Frequency distribution of masking in RNA sequences.



(c) Frequency distribution of masking in RNA structures.

Figure 26: An overview of the data preparation process for RNA sequences and structures.

thoroughly reviewed in Section 4.3 were also used to evaluate the proposed model. This diverse selection of reference models allows for comprehensive assessment within different computational paradigms. The evaluation metric chosen for the models is MCRMSE, enabling a direct comparison of model prediction capabilities with established benchmarks in this study.

The performance and generalization capability of the StructmRNA model were evaluated against various machine learning models based on three criteria: (1) improvement in errors for the training set and public test dataset, (2) absolute difference between the final training error and **public test dataset error**, referred to as [1], and (3) the lowest achieved final errors.

4.4.3 Results and Discussion

The **StructmRNA + OpenVaccine_Data** model, trained using StructmRNA and mRNA sequences along with secondary structure data from the *OpenVaccine* dataset, demonstrated superior performance compared

[1] generalization gaps

to other models. This model achieved the lowest MCRMSE score of **0.07**, indicating high prediction accuracy.

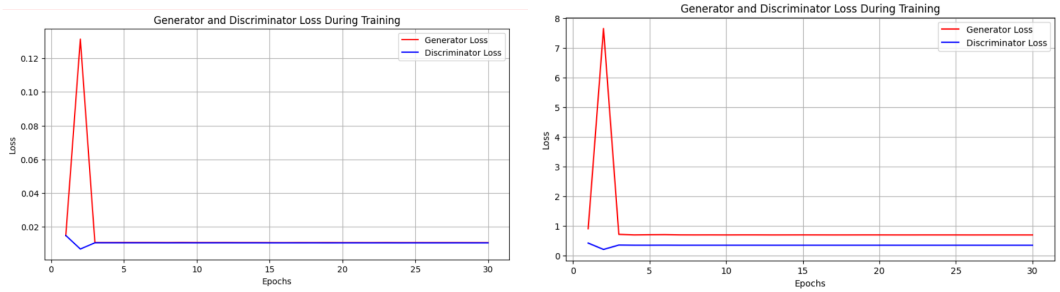
Following this, the **StructmRNA + GAN_Data** and **StructmRNA + NCBI_Data** models further demonstrated the effectiveness of StructmRNA. These models achieved significant MCRMSE scores of **0.11** and **0.10**, respectively. The **StructmRNA + GAN_Data** model utilizes StructmRNA trained on synthetic mRNA sequences generated by GAN, along with secondary structures provided by the ViennaRNA tool.

Similarly, the **StructmRNA + NCBI_Data** model employs the StructmRNA architecture trained on mRNA sequences alongside a secondary structure dataset from NCBI.

Among the *OpenVaccine* models, genetic algorithm and Kazuki2 models exhibited similar performance with error rates of approximately **0.22** and **0.23**, respectively. In contrast, traditional embedding methods such as Word2Vec and ELMo resulted in higher MCRMSE values (**0.41** and **0.44**), highlighting the challenge of achieving high accuracy in mRNA degradation prediction. Figure 27 illustrates the progression of training and validation errors for different sequence embedding models used in mRNA degradation prediction.

These models predict five target features: **reactivity**, **deg_Mg_pH10**, **deg_pH10**, **deg_Mg_50C**, and **deg_50C**. The average errors of all models were computed over four repetitions to compare the predicted labels with actual values.

Furthermore, while pretraining StructmRNA with synthetic data does not yield significant improvements in mRNA degradation prediction, the convergence observed in the proposed model pretrained with real data exhibits similar convergence patterns after 30 epochs, indicating model stability and statistical similarity between synthetic and real sequences (Figure 28). These results demonstrate the effectiveness of advanced machine learning techniques, particularly those leveraging complex embedding methods, in enhancing prediction performance.



(a) (b)

Figure 28: Comparison of BERT model convergence on real and synthetic RNA sequences. (a) Convergence on real data. (b) Convergence with 50% real data and 50% synthetic data.

According to the points raised in Section 4.4.2, the performance and generalization capability of the StructmRNA model have been evaluated against various machine learning models based on three criteria: 1) Improvement in training and general test set errors, 2) The absolute difference between final training errors and

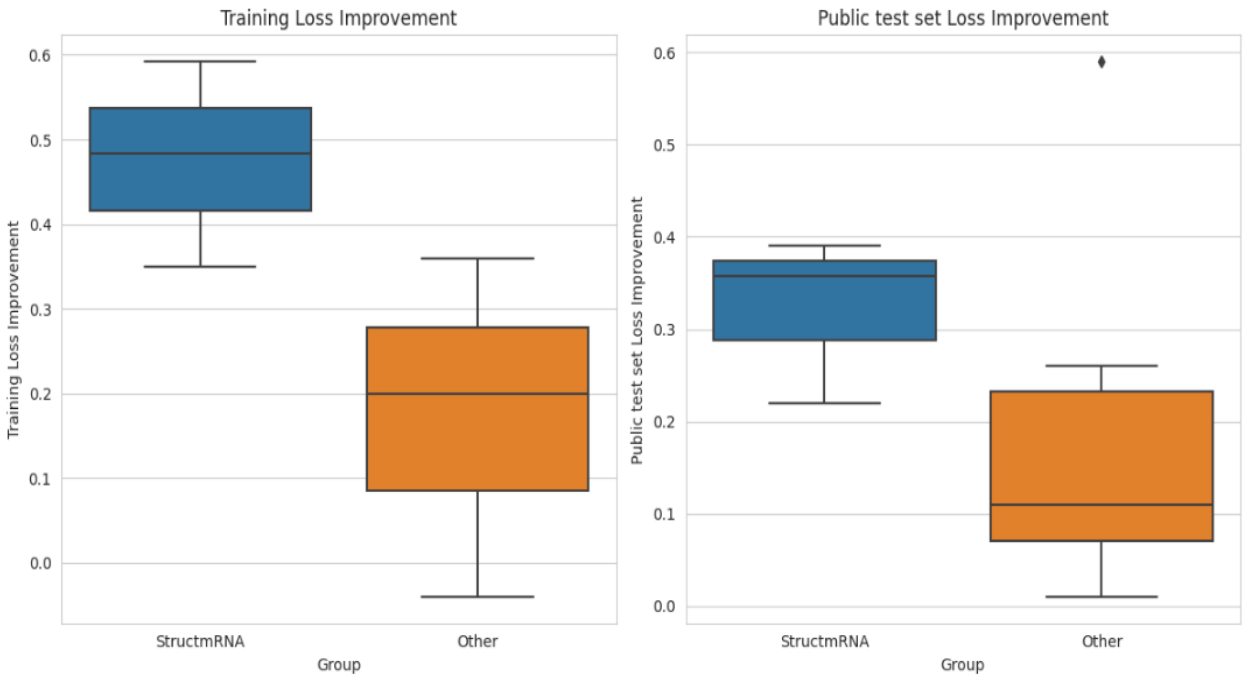
Table 12: Performance outcomes of mRNA degradation prediction models assessed on the OpenVaccine public dataset. The final three rows denote results from our StructmRNA model, pre-trained on the OpenVaccine, NCBI, and GAN datasets, respectively. All numerical values have been rounded to two decimal places.

Models	MCRMSE (Public test set)
Models From Kaggle OpenVaccine Competition	
Experimental error	0.12
DegScore	0.39
DegScore-XGBoost	0.36
Nullrecurrent	0.23
Kazuki2	0.23
Genetic algorithm (10 of top 100 selected)	0.22
Ensemble top two models	0.22
Models in respect to embedding methods	
Word2vec	0.41
ELMo	0.44
LSTM	0.26
CNN	0.35
VAE	0.21
AGC	0.25
StructmRNA + OpenVaccine_Data	**0.07**
StructmRNA + NCBI_Data	**0.10**
StructmRNA + GAN_Data	**0.11**

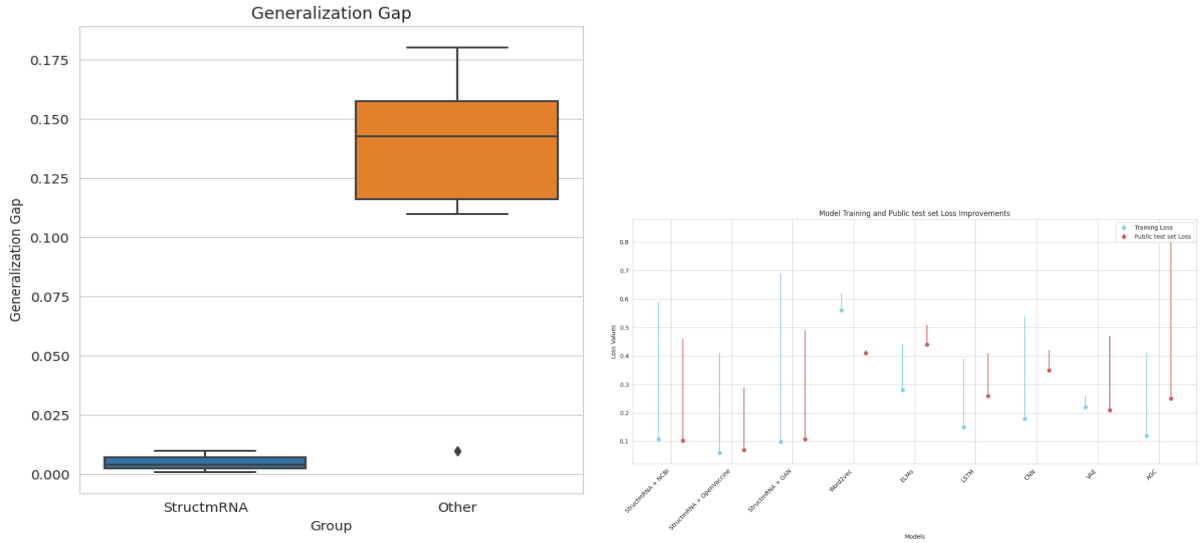
general test set errors, known as the generalization gap, and 3) The lowest final errors obtained.

To evaluate and confirm the statistical significance of the observed superior performance of the StructmRNA model across the previously discussed evaluation criteria, we employed a one-way Analysis of Variance (ANOVA) test. This statistical method was chosen due to its effectiveness in comparing the means of multiple groups to determine whether at least one model performs significantly differently from the others. In our case, the ANOVA test was conducted to compare the training errors of StructmRNA against those of the alternative models included in our study. The results of this analysis demonstrated that StructmRNA yields a significantly lower training error relative to the other models, with an F-statistic of 8.76 and a corresponding p-value of 0.021. These results indicate that the difference in performance is unlikely to have occurred by chance and provide strong statistical support for the effectiveness of StructmRNA.

However, no significant difference was observed in the improvement of **general test set** errors, indicating similar error reductions in the **general test set** across models. The generalization gap also showed a significant difference, highlighting variations in models' ability to generalize from training datasets to the **general test set**. Notably, the lowest final errors recorded for training and **general test set** were 0.60 and 0.07, respectively, demonstrating the models' effectiveness in minimizing errors at the end of the training phase (Figure 29 (a) and (b)).



(a)



(b) (c)

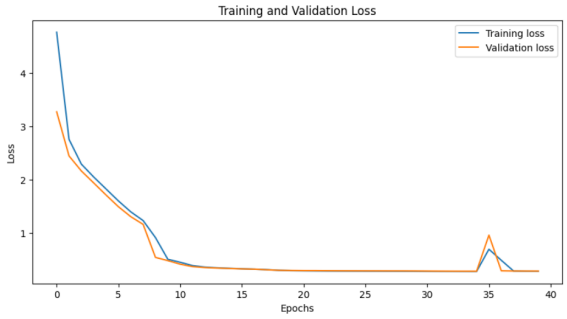
Figure 29: Overview of error reduction in the training set and **general test set** and the difference between training and test results. (a) Error reduction in the training set and general test set. (b) Comparison of error reduction in training and general test set across different models. (c) Models' generalization ability and potential overfitting.

Additionally, Figure 29 (c) presents the final training and **general test set** errors for a collection of machine learning models. The error bar vector illustrates the reduction in error from the beginning (top of the bar) to the end (point) of the training process, indicating each model's learning quality. The blue and red markers represent the final training and **general test set** errors, respectively, across nine models, including

StructmRNA (covering all models pretrained on NCBI **OpenVaccine** and GAN data), Word2Vec, ELMo, LSTM, CNN, VAE, and AGC. The error bars effectively show the models' learning progress and their ability to generalize from training data to **general test set** data, where shorter bars indicate less improvement and longer bars indicate significant error reduction. Comparing the training and **general test set** error points for each model allows direct assessment of their performance and generalization differences. These findings suggest a promising outlook for the StructmRNA model's performance compared to other models in similar applications.

The significant difference in training error improvements indicates that the StructmRNA model performs better during training and reduces training errors more effectively. However, this superiority does not always translate to the **general test set**, meaning there is no substantial difference in error reduction within the **general test set**. This observation suggests that enabling models to generalize to new data is a complex challenge, and models may sometimes experience overfitting or underfitting.

During the training process of the proposed model, while the GPU handled most of the computations, the CPU was also significantly utilized for data management, preprocessing, and pretraining StructmRNA (Figure 30).



Criterion	Value
Training Time	220 hours
CPU Usage	70%
GPU Usage	GPU T4 x 2
Batch Size	64

Figure 30: Left table: Performance of the proposed model pretrained on the structures and sequences of the extitOpenVaccine mRNA dataset. Right chart: Model training efficiency and scalability.

4.4.4 Conclusion

In this chapter, various machine learning models for graph embedding and their significance in bioinformatics were reviewed and evaluated. The results demonstrated that the proposed IsoGLoVe model performed better in tissue PPI networks compared to baseline models such as SVM and KNN.

Subsequently, the evaluation of mRNA sequence embedding models using various metrics revealed that models such as LSTM, GRU, CNN, and BERT performed well in improving embedding quality and reducing errors. Then, the StructmRNA model was introduced based on the obtained results for effective sequence and structural embedding of mRNA. The proposed model was further evaluated using multiple criteria in the mRNA degradation prediction task, demonstrating notable performance in learning accuracy and generalization compared to other models.

The outcomes of this study demonstrated that the StructmRNA model, when trained using a combination

of synthetically generated data produced by Generative Adversarial Networks (GANs) and authentic, real-world data, was able to achieve a level of performance that was comparable to training with real data alone. This observation suggests that the StructmRNA model exhibits a strong capacity for generalization, allowing it to effectively learn from and adapt to multiple data sources, including both artificial and naturally obtained datasets. Such capability is particularly significant in the context of bioinformatics, where high-quality annotated data can often be limited or difficult to obtain.

Moreover, the integration of synthetic data into the training process provides several practical advantages. Not only does it offer a means to artificially expand the available dataset, thereby reducing the reliance on scarce real data, but it also presents opportunities for exploring a broader distribution of biological patterns that may not be fully captured in real datasets. This can potentially lead to the discovery of new features or relationships within biological sequences that would otherwise remain unnoticed.

Despite these promising results, the findings also underscore the importance of conducting further investigations aimed at optimizing the use of synthetic data in machine learning applications for bioinformatics. Future research should focus on identifying the most effective methods for generating high-fidelity synthetic data, determining the optimal ratios of synthetic to real data during model training, and understanding the specific characteristics of synthetic samples that contribute most significantly to model performance.

In conclusion, this research not only demonstrates the potential effectiveness of incorporating synthetic biology data into computational models but also establishes a foundation for future advancements in the application of synthetic data within machine learning frameworks. This is especially relevant for domains where data scarcity poses a significant barrier to the development of accurate and robust predictive models.

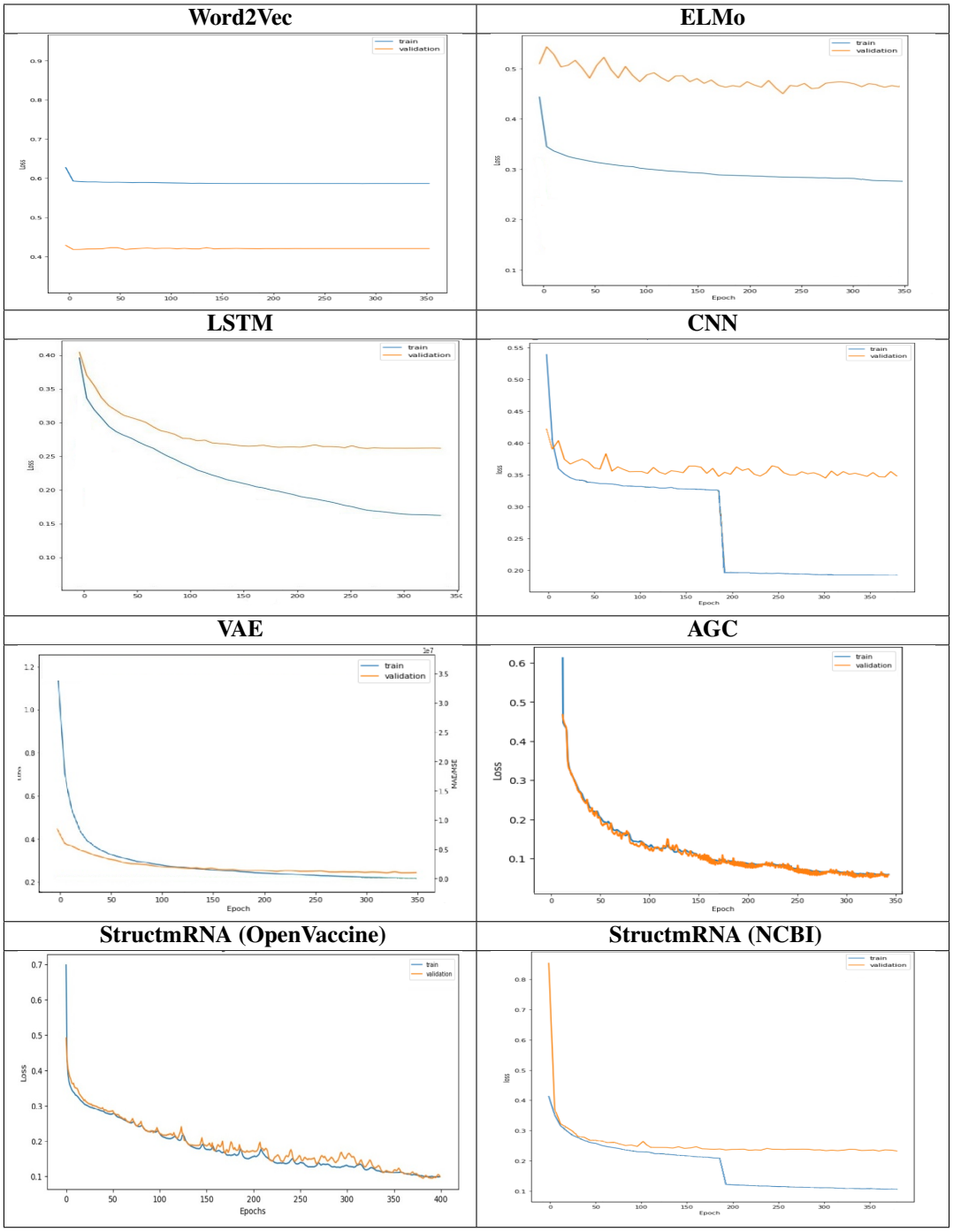


Figure 27: Performance of various sequence embedding models in the mRNA degradation prediction task, aiming to predict five target features (**reactivity**, **deg_Mg_pH10**, **deg_pH10**, **deg_Mg_50C**, and **deg_50C**).

5 Conclusion

5.1 Conclusion

Predicting mRNA degradation and analyzing its structures are key topics in bioinformatics with significant impacts on the design of RNA-based therapies. In this research, the challenges of predicting mRNA degradation using machine learning methods have been explored.

The goal of this study was to develop a model named StructmRNA, which leverages the BERT architecture and advanced dual-level and conditional masking techniques. This model demonstrated outstanding performance in analyzing and vector representation of RNA sequences and structures, effectively capturing the complex relationships between RNA structure and sequence. Specifically, by utilizing vector representations of sequences and advanced masking techniques, it has accurately simulated the interactions between RNA sequences and structures without sequence length limitations.

Experimental results showed that StructmRNA significantly outperformed other machine learning models in predicting mRNA degradation. This model achieved superior performance in RNA molecular representation and generalization to various datasets compared to all existing models while maintaining convergence and stability throughout training. However, evaluations indicated that adding synthetic data generated by GAN did not have a significant impact on improving the model's performance. Nevertheless, the obtained results confirm the high stability of StructmRNA and the accuracy of synthetic data in representing real mRNA structures. These findings highlight the necessity of further research to enhance the interaction between synthetic and real data.

The advancement of StructmRNA, through precise design and optimization techniques, represents a crucial step in improving the accuracy and speed of RNA-related predictions. The use of diverse datasets and advanced preprocessing methods has ensured model accuracy while preventing data bias.

Beyond introducing an innovative model, this study underscores the importance of deep learning-based textual analysis in RNA sequence analysis and clearly outlines the advantages and limitations of these methods. This analysis aids researchers in selecting the most suitable approach based on their specific challenges. These insights are particularly important in RNA bioinformatics, especially for predicting RNA structures and sequences. StructmRNA, as an efficient tool, holds potential applications in RNA-based vaccine design and new therapeutic strategies. The results of this study can serve as a foundation for the development of more advanced models and the intelligent utilization of synthetic data in future research, paving new ways for optimal data exploitation in bioinformatics.

5.2 Future Work

For further research and future developments in this field, it is recommended that the IsoGloVe method be considered in other complex networks such as social networks. Additionally, the impact of vector dimensions on evaluation results should be investigated, and the application of IsoGloVe pre-trained vectors in various information retrieval problems [21, 45, 47, 49, 50, 60, 74, 75, 91, 121, 131, 132] and other applications should

be analyzed [22, 48, 55, 69, 117, 133]. Regarding the StructmRNA model, it is suggested to examine the generalization gap between training and public test datasets, integrate synthetic and real data to enhance prediction accuracy without reducing model performance, and optimize the training process with a focus on model convergence and continuous updates in the model's pretraining phase to maintain data quality and reduce StructmRNA bias.

Furthermore, it is recommended to explore the application of the proposed StructmRNA model in mRNA-based drug development and its contribution to therapeutic research. In the field of vector embedding learning for biological sequences, existing models should be refined to address issues such as overfitting. Additionally, the use of diverse training datasets, including synthetic and heterogeneous data, should be emphasized.

Bibliography

[1] National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/.

[2] ABDEL-BASSET, M., MOUSTAFA, N., AND HAWASH, H. *Generative Adversarial Networks (GANs)*. 2023, pp. 271–285.

[3] AGARWAL, V., REDDY, N. J. K., AND ANAND, A. Unsupervised representation learning of DNA sequences. *arXiv preprint arXiv:1906.03087* (2019). Accepted at 2019 ICML Workshop on Computational Biology.

[4] AKIYAMA, M., AND SAKAKIBARA, Y. Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning. *NAR Genomics and Bioinformatics 4*, 1 (Feb 2022), lqac012.

[5] AKIYAMA, M., AND SAKAKIBARA, Y. Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning. *NAR Genomics and Bioinformatics 4*, 1 (Feb. 2022), lqac012.

[6] ALAMMAR, J. Finding the words to say: Hidden state visualizations for language models, 2021.

[7] ALI EZZAT, MIN WU, X.-L. L. C.-K. K. Drug-target interaction prediction using ensemble learning and dimensionality reduction. *Methods* (2017).

[8] ALLEY, E. C., KHIMULYA, G., BISWAS, S., ALQURAISHI, M., AND CHURCH, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods 16*, 12 (2019), 1315–1322.

[9] AMARIEI, G. *Document Clustering of Irish Government Circulars using Machine Learning Techniques*. PhD thesis, National College of Ireland, Dublin, 2024.

[10] AN, A., HUANG, Y., HUANG, X., AND CERCONE, N. Feature selection with rough sets for web page classification. *Trans. Rough Sets 2* (2004), 1–13.

[11] ASIM, M. N., IBRAHIM, M. A., ASIF, T., AND DENGEL, A. Rna sequence analysis landscape: A comprehensive review of task types, databases, datasets, word embedding methods, and language models. *Heliyon 11*, 2 (2025).

[12] ASIM, M. N., IBRAHIM, M. A., ZAIB, A., AND DENGEL, A. Dna sequence analysis landscape: a comprehensive review of DNA sequence analysis task types, databases, datasets, word embedding methods, and language models. *Frontiers in Medicine 12* (2025), 1503229.

[13] BACK, B., AND PAPERSUBMIT. Functional annotation of proteins using domain embedding based sequence classification. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)* (Vienna, Austria, 2019), SCITEPRESS, pp. 163–170.

[14] BELINKOV, Y., GEHRMANN, S., AND PAVLICK, E. Interpretability and analysis in neural NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts* (Online, 2020), Association for Computational Linguistics, pp. 1–5.

[15] BHATT, N., BHATT, N., AND PRAJAPATI, P. Empirical analysis of word embedding methods for estimating their. *ICT for Intelligent Systems: Proceedings of ICTIS 2024, Volume 5 1111* (2024), 131.

[16] BOJANOWSKI, P., GRAVE, E., JOULIN, A., AND MIKOLOV, T. Enriching word vectors with subword information. *MIT Press* (2017).

[17] BONANDRINI, R., GATTI, D., ET AL. fasttext (sub) word vectors. In *Reference Module in Social Sciences*. Elsevier Inc., 2024.

[18] BRENNECKE, J., STARK, A., RUSSELL, R., AND COHEN, S. Principles of microrna-target recognition. *PLOS Biology 3* (2005).

[19] BRYAN PEROZZI, RAMI AL-RFOU, S. S. Deepwalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, 2014), ACM, pp. 701–710.

[20] CHEN, C., GHERZI, R., ONG, S., CHAN, E., RAIJMAKERS, R., PRUIJN, G., STOECKLIN, G., MORONI, C., MANN, M., AND KARIN, M. Au binding proteins recruit the exosome to degrade are-containing mrnas. *Cell 107* (2001), 451–464.

[21] CHEN, Q., HU, Q., HUANG, J. X., HE, L., AND AN, W. Enhancing recurrent neural networks with positional attention for question answering. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, Aug 7-11, 2017* (2017), N. Kando, T. Sakai, H. Joho, H. Li, A. P. de Vries, and R. W. White, Eds., ACM, pp. 993–996.

[22] CHEN, X., XIONG, K., ZHANG, Y., XIA, L., YIN, D., AND HUANG, J. X. Neural feature-aware recommendation with signed hypergraph convolutional network. *ACM Trans. Inf. Syst. 39*, 1 (2020), 8:1–8:22.

[23] CHENG, F., WANG, Y., BAI, Y., LIANG, Z., MAO, Q., LIU, D., WU, X., AND XU, M. Research advances on the stability of mRNA vaccines. *Viruses 15*, 3 (2023), 668.

[24] COMPETITOR, J., AND PARTICIPANT, J. Evaluation metrics in the openvaccine competition: Embracing mcrmse. In *Proceedings of the 2022 International Symposium on Vaccine Research* (2022), pp. 310–315.

[25] CORDERO, P., LUCKS, J. B., AND DAS, R. An RNA Mapping DataBase for curating RNA structure mapping experiments. *Bioinformatics 28*, 22 (2012), 3006–3008.

[26] CROMMELIN, D. J., ANCHORDOQUY, T. J., VOLKIN, D. B., JISKOOT, W., AND MASTROBATTISTA, E. Addressing the cold reality of mRNA vaccine stability. *Journal of pharmaceutical sciences 110*, 3 (2021), 997–1001.

[27] CUI, F., ZHANG, Z., CAO, C., ZOU, Q., CHEN, D., AND SU, X. Protein-dna/rna interactions: Machine intelligence tools and approaches in the era of artificial intelligence and big data. *Proteomics 22*, 8 (2022), e2100197.

[28] CUI, F., ZHANG, Z., AND ZOU, Q. Sequence representation approaches for sequence-based protein prediction tasks that use deep learning. *Briefings in Functional Genomics 20*, 1 (2021), 61–73. PMID: 33527980.

[29] DAS, W. S., AND ET AL. `https://kaggle.com/competitions/stanford-covid-vaccine`, 2020.

[30] DEANA, A., CELESNIK, H., AND BELASCO, J. The bacterial enzyme rpph triggers messenger RNA degradation by 5' pyrophosphate removal. *Nature 451* (2008), 355–358.

[31] DEIGAN, K. E., LI, T. W., MATHEWS, D. H., AND WEEKS, K. M. Accurate SHAPE-directed RNA structure determination. *Proceedings of the National Academy of Sciences of the United States of America 106*, 1 (Jan. 2009), 97–102. Epub 2008 Dec 24.

[32] DENG, L., AND YU, D. *Deep Learning: Methods and Applications*. 2014.

[33] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Association for Computational Linguistics* (2019).

[34] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics* (2019), Association for Computational Linguistics.

[35] DOSHI, K. J., CANNONE, J. J., COBAUGH, C. W., AND GUTELL, R. R. Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for rna secondary structure prediction. *BMC Bioinformatics 5* (2004), 105.

[36] EDERA, A. A., SMALL, I., MILONE, D. H., AND SANCHEZ-PUERTA, M. V. Deepred-mt: Deep representation learning for predicting c-to-u RNA editing in plant mitochondria. *Computers in Biology and Medicine 136* (2021), 104682.

[37] ERASLAN, G., AVSEC, VZ., GAGNEUR, J., AND THEIS, F. J. Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics 20*, 7 (2019), 389–403.

[38] FASOULIS, R., PALIOURAS, G., AND KAVRAKI, L. E. Graph representation learning for structural proteomics. *Emerging Topics in Life Sciences 5*, 6 (Dec 2021), 789–802.

[39] GANEA, O.-E., AND BÉCIGNEUL, G. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems* (2018), pp. 5755–5764.

[40] GHOJOGH, B., KARRAY, F., AND CROWLEY, M. Anomaly detection and prototype selection using polyhedron curvature. vol. abs/2010.10702.

[41] GROVER, A., AND LESKOVEC, J. node2vec: scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, CA, 2016), ACM, pp. 855–864.

[42] GUIDOTTI, R., MONREALE, A., RUGGIERI, S., TURINI, F., GIANNOTTI, F., AND PEDRESCHI, D. A survey of methods for explaining black box models. *ACM computing surveys (CSUR) 51*, 5 (2018), 1–42.

[43] HAI-CHENG YI, ZHU-HONG YOU, L. C.-X. Z. T.-H. J. X. L. Y.-B. W. Learning distributed representations of RNA and protein sequences and its application for predicting lncrna-protein interactions. *15* (2019).

[44] HANNAH K. WAYMENT-STEELE, WIPAPAT K, A. M. W. D. S. K. B. T. W. R.-M. D. J. R. R. W.-O. J. J. N. J. G. K. O. K. F. H. M. G. V. M. T. B. S. T. I. T. N. S. H. K. I. Y. L. F. A. C. E. K. A. M. F. E. P. R. D. Deep learning models for predicting RNA degradation via dual crowdsourcing. *Nature Machine Intelligence 4*, 12 (2022), 1174–1184.

[45] HE, B., HUANG, J. X., AND ZHOU, X. Modeling term proximity for probabilistic information retrieval models. *Inf. Sci. 181*, 14 (2011), 3017–3031.

[46] HOFACKER, I. L., PRIWITZER, B., AND STADLER, P. F. Prediction of locally stable RNA secondary structures for genome-wide surveys. *Bioinformatics 20*, 2 (Jan. 2004), 186–190.

[47] HUANG, J. X., MIAO, J., AND HE, B. High performance query expansion using adaptive co-training. *Inf. Process. Manag. 49*, 2 (2013), 441–453.

[48] HUANG, X., CERCONE, N., AND AN, A. Comparison of interestingness functions for learning web usage patterns. In *Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, McLean, VA, USA, Nov 4-9, 2002* (2002), ACM, pp. 617–620.

[49] HUANG, X., HUANG, Y. R., WEN, M., AN, A., LIU, Y., AND POON, J. Applying data mining to pseudo-relevance feedback for high performance text retrieval. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 dec 2006, Hong Kong, China* (2006), IEEE Computer Society, pp. 295–306.

[50] HUANG, Y., AND HUANG, J. A survey on retrieval-augmented text generation for large language models. *CoRR abs/2404.10981* (2024).

[51] HUMPHREY, S., KERR, A., RATTRAY, M., DIVE, C., AND MILLER, C. A model of k-mer surprisal to quantify local sequence information content surrounding splice regions. *PeerJ 8* (2020), e10063.

[52] HUSNAIN, M., MISSEN, M. M. S., MUMTAZ, S., LUQMAN, M. M., COUSTATY, M., AND OGIER, J.-M. Visualization of high-dimensional data by pairwise fusion matrices using t-sne. vol. 11.

[53] HUYNEN, M. A., PERELSON, A., VIEIRA, W. A., AND STADLER, P. F. Base pairing probabilities in a complete HIV-1 RNA. *Journal of Computational Biology 3*, 2 (summer 1996), 253–274.

[54] ISKEN, O., AND MAQUAT, L. Quality control of eukaryotic mrna: safeguarding cells from abnormal mRNA function. *Genes & Development 21*, 15 (2007), 1833–1856.

[55] JAHAN, I., LASKAR, M. T. R., PENG, C., AND HUANG, J. X. Evaluation of chatgpt on biomedical tasks: A zero-shot comparison with fine-tuned generative transformers. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks, BioNLP@ACL 2023, Toronto, Canada, 13 July 2023* (2023), D. Demner-Fushman, S. Ananiadou, and K. Cohen, Eds., Association for Computational Linguistics, pp. 326–336.

[56] JI, Y., ZHOU, Z., LIU, H., AND DAVULURI, R. V. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics 37*, 15 (2021), 2112–2120.

[57] JIAN TANG, MENG QU, M. W. M. Z. J. Y. Q. M. Line: large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web* (Florence, Italy, 2015), ACM, pp. 1067–1077.

[58] JIN, S., ZENG, X., XIA, F., HUANG, W., AND LIU, X. Application of deep learning methods in biological networks. *Briefings in Bioinformatics 22*, 2 (05 2020), 1902–1917.

[59] JOHNSON, S. J., MURTY, M. R., AND NAVAKANTH, I. A detailed review on word embedding techniques with emphasis on word2vec. *Multimedia Tools and Applications 83*, 13 (2024), 37979–38007.

[60] KEYVAN, K., AND HUANG, J. X. How to approach ambiguous queries in conversational search: A survey of techniques, approaches, tools, and challenges. *ACM Comput. Surv. 55*, 6 (2023), 129:1–129:40.

[61] KIM, B., WATTENBERG, M., GILMER, J., CAI, C., WEXLER, J., AND VIEGAS, F. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning* (2018), PMLR, pp. 2668–2677.

[62] KIS, Z. Stability modelling of mRNA vaccine quality based on temperature monitoring throughout the distribution chain. *Pharmaceutics 14*, 2 (2022), 430.

[63] KOO, P., MAJDANDZIC, A., PLOENZKE, M., ANAND, P., AND PAUL, S. Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. *PLoS Comput Biol 17*, 5 (2021), e1008925.

[64] KOO, P. K., AND EDDY, S. R. Representation learning of genomic sequence motifs with convolutional neural networks. *PLoS Computational Biology 15*, 12 (Dec 2019), e1007560.

[65] KOPF, A., AND CLAASSEN, M. Latent representation learning in biology and translational medicine. *Patterns 2*, 3 (2021), 100198.

[66] KRISHNA, U. V., PREMJITH, B., AND SOMAN, K. P. A comparative study of pre-trained gene embeddings for covid-19 mRNA vaccine degradation prediction. In *Proceedings of the Seventh International Conference on Mathematics and Computing* (Singapore, 2022), D. Giri, K.-K. Raymond Choo, S. Ponnusamy, W. Meng, S. Akleylek, and S. Prasad Maity, Eds., Springer Singapore, pp. 301–308.

[67] LACAN, A., SEBAG, M., AND HANCZAR, B. GAN-based data augmentation for transcriptomics: survey and comparative assessment. *Bioinformatics 39*, Supplement_1 (2023), i111–i120.

[68] LANCZOS, C. Solution of a set of linear equations and least squares problems by optimization over a complete orthogonal set. No. 3, National Bureau of Standards, pp. 579–588.

[69] LASKAR, M. T. R., HOQUE, E., AND HUANG, J. X. WSL-DS: weakly supervised learning with distant supervision for query focused multi-document abstractive summarization. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), dec 8-13, 2020* (2020), D. Scott, N. Bel, and C. Zong, Eds., International Committee on Computational Linguistics, pp. 5647–5654.

[70] LENORE COWEN, TREY IDEKER, B. J. R. R. S. Review network propagation: a universal amplifier of genetic associations. *Nat Rev Genet 18*, 9 (2017), 551–562.

[71] LEWIN, B., KREBS, J. E., KILPATRICK, S. T., AND GOLDSTEIN, E. S. *Lewin's Genes X*. Jones and Bartlett, Sudbury, Mass., 2011.

[72] LIU, Q., WANG, D., ZHOU, L., LI, J., AND WANG, G. Mtgdc: a multi-scale tensor graph diffusion clustering for single-cell RNA sequencing data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2023).

[73] LU, W., TANG, Y., WU, H., HUANG, H., FU, Q., QIU, J., AND LI, H. Predicting RNA secondary structure via adaptive deep recurrent neural networks with energy-based filter. *BMC Bioinformatics 20*, Suppl 25 (Dec. 2019), 684.

[74] LUPU, M., HUANG, J. X., ZHU, J., AND TAIT, J. TREC-CHEM: large scale chemical information retrieval evaluation at TREC. *SIGIR Forum 43*, 2 (2009), 63–70.

[75] LUPU, M., PIROI, F., HUANG, X., ZHU, J., AND TAIT, J. Overview of the TREC 2009 chemical IR track. In *Proceedings of The Eighteenth Text REtrieval Conference, TREC 2009, Gaithersburg, Maryland, USA, Nov 17-20, 2009* (2009), E. M. Voorhees and L. P. Buckland, Eds., vol. 500-278 of *NIST Special Publication*, National Institute of Standards and Technology (NIST).

[76] MARA, A., LIJFFIJT, J., AND DE BIE, T. Evalne: A framework for network embedding evaluation. *SoftwareX 17* (2022), 100997.

[77] MENGHANI, G. Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Computing Surveys 55*, 12 (2023), 1–37.

[78] MENGYU GEHAOMIAO, YANGXIAN MING, K. X. Openvaccine: Covid-19 mRNA vaccine degradation prediction, 2020.

[79] MICHAEL HEINZINGER, AHMED ELNAGGAR, Y. W. C. D. D. N. F. M. . B. R. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics* (2019).

[80] MIKHAIL BELKIN, P. N. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput. 15* (2003), 1373–1396.

[81] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS)* (Lake Tahoe, Nevada, 2013), p. n/a.

[82] MIKOLOV, T., CHEN, K., CORRADO, G., AND DEAN, J. Efficient estimation of word representations in vector space. *arXiv* (2013).

[83] MOLNAR, C. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Leanpub, 2023. Online book, version 2023.

[84] MURAD, T., ET AL. Exploring the potential of gans in biological sequence analysis. *Biology 12*, 6 (2023), 854.

[85] MUZIO, G., O'BRAY, L., AND BORGWARDT, K. Biological network analysis with deep learning. *Briefings in Bioinformatics 22*, 2 (11 2020), 1515–1530.

[86] OBBARD, D., GORDON, K., BUCK, A., AND JIGGINS, F. The evolution of RNAi as a defence against viruses and transposable elements. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 364* (2009).

[87] ODRZYWOLEK, K., KARWOWSKA, Z., MAJTA, J., BYRSKI, A., MILANOWSKA-ZABEL, K., AND KOSCIOLEK, T. Deep embeddings to comprehend and visualize microbiome protein space. *Sci Rep 12*, 1 (2022), 10332.

[88] PARDI, N., HOGAN, M. J., PORTER, F. W., AND WEISSMAN, D. mRNA vaccines — a new era in vaccinology. *Nature Reviews Drug Discovery 17*, 4 (2018), 261–279.

[89] PARKER, R., AND SHETH, U. P bodies and the control of mRNA translation and degradation. *Molecular Cell* (2007), 635–646.

[90] PEDERSON, T. Review of "RNA: Life's indispensable molecule" by james e. darnell. In *RNA* (2011), vol. 17, Cold Spring Harbor Laboratory Press, pp. 1771–1774.

[91] PENG, F., HUANG, X., SCHUURMANS, D., AND CERCONE, N. Investigating the relationship between word segmentation performance and retrieval performance in chinese IR. In *19th Interna-*

tional Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, Aug 24 - September 1, 2002 (2002).

[92] PENNINGTON, J., SOCHER, R., AND MANNING, C. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha, Qatar, Oct. 2014), A. Moschitti, B. Pang, and W. Daelemans, Eds., Association for Computational Linguistics, pp. 1532–1543.

[93] PETERS, M. E., NEUMANN, M., IYYER, M., GARDNER, M., CLARK, C., LEE, K., AND ZETTLEMOYER, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana, 2018), M. Walker, H. Ji, and A. Stent, Eds., Association for Computational Linguistics, pp. 2227–2237.

[94] PETERS, M. E., NEUMANN, M., IYYER, M., GARDNER, M., CLARK, C., LEE, K., AND ZETTLEMOYER, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana, 2018), M. Walker, H. Ji, and A. Stent, Eds., Association for Computational Linguistics, pp. 2227–2237.

[95] PYLE, A. M., AND SCHLICK, T. Challenges in RNA structural modeling and design. *Journal of Molecular Biology 428*, 5 Pt A (2016), 733–735.

[96] RITZ, J., MARTIN, J. S., AND LAEDERACH, A. Evolutionary evidence for alternative structure in RNA sequence covariation. *PLoS Computational Biology 9*, 8 (2013), e1003171.

[97] SAVELLI, C., AND GIOBERGIA, F. Enhancing cross-lingual word embeddings: Aligned subword vectors for out-of-vocabulary terms in fasttext. In *2024 IEEE 18th International Conference on Application of Information and Communication Technologies (AICT)* (2024), IEEE, pp. 1–6.

[98] SCHLICK, T., AND PYLE, A. M. Opportunities and challenges in RNA structural modeling and design. *Biophysical Journal 113*, 2 (July 2017), 225–234. Epub 2017 Feb 2.

[99] SHAN, Y., YANG, J., LI, X., ZHONG, X., AND CHANG, Y. Glae: A graph-learnable auto-encoder for single-cellRNA-seq analysis. *Information Sciences 621* (2023), 88–103.

[100] SHENG, N., HUANG, L., GAO, L., CAO, Y., XIE, X., AND WANG, Y. A survey of computational methods and databases for lncrna-mirna interaction prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2023).

[101] SIEGFRIED, N. A., BUSAN, S., RICE, G. M., NELSON, J. A. E., AND WEEKS, K. M. RNA motif discovery by SHAPE and mutational profiling (SHAPE-MaP). *Nature Methods 11*, 9 (Sept. 2014), 959–965. Epub 2014 Jul 13.

[102] SINGH, J., HANSON, J., PALIWAL, K., AND ZHOU, Y. RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nature Communications 10*, 1 (Nov. 2019), 5407.

[103] SLOMA, M. F., AND MATHEWS, D. H. Improving rna secondary structure prediction with structure mapping data. *Methods in Enzymology 553* (2015), 91–114.

[104] SMITH, J., AND JOHNSON, A. On the utility of mean square error in prediction models. In *Proceedings of the 1985 International Symposium on Statistical Methods* (1985), pp. 256–260.

[105] SONG, B., LI, Z., LIN, X., WANG, J., WANG, T., AND FU, X. Pretraining model for biological sequence data. *Brief Funct Genomics 20*, 3 (2021), 181–195.

[106] STARK, C., ET AL. Biogrid: A general repository for interaction datasets. pp. D535–D539.

[107] STEFAN MAUTNER, SOHEILA MONTASERI, M. M. M. R. F. C. R. B. Shaker:RNA shape prediction using graph kernel. *Bioinformatics 35* (2019), i354–i359.

[108] TALBI, E.-G. Automated design of deep neural networks: A survey and unified taxonomy. *ACM Computing Surveys (CSUR) 54*, 2 (2021), 1–37.

[109] TIAN, S., AND DAS, R. RNA structure through multidimensional chemical mapping. *Quarterly Reviews of Biophysics 49* (Jan. 2016), e7. Research Support, N.I.H., Extramural; Research Support, Non-U.S. Gov't; Review.

[110] UDDIN, M. N., AND RONI, M. A. Challenges of storage and stability of mrna-based covid-19 vaccines. *Vaccines 9*, 9 (2021), 1033.

[111] VANDEWIELE, G. Predicting mRNA degradation using gnns and rnns in the search for a covid-19 vaccine, 2020. A write-up of the fourth place solution to the Kaggle OpenVaccine competition.

[112] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., AND POLOSUKHIN, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2017), NIPS'17, Curran Associates Inc., p. 6000–6010.

[113] WANG, J., HUANG, J. X., TU, X., WANG, J., HUANG, A. J., LASKAR, M. T. R., AND BHUIYAN, A. Utilizing BERT for information retrieval: Survey, applications, resources, and challenges. *ACM Comput. Surv. 56*, 7 (2024), 185:1–185:33.

[114] WANG, K., HU, J., AND ZHANG, X. Identifying drug–target interactions through a combined graph attention mechanism and self-attention sequence embedding model. In *Advanced Intelligent Computing Technology and Applications* (Singapore, 2023), Springer Nature Singapore, pp. 246–257.

[115] WAYMENT-STEELE, H., KLADWANG, W., WATKINS, A., ET AL. Deep learning models for predicting RNA degradation via dual crowdsourcing. *Nature Machine Intelligence 4* (2022), 1174–1184.

[116] WEN ZHANG, XIANG YUE, G. T. W. W. F. H. X. Z. Sequence-based feature projection ensemble learning for predicting lncrna-protein interactions. *PLoS Comput Biol 14* (2018).

[117] XIA, L., HUANG, C., XU, Y., ZHAO, J., YIN, D., AND HUANG, J. X. Hypergraph contrastive collaborative filtering. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022* (2022), E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, and G. Kazai, Eds., ACM, pp. 70–79.

[118] XIE, P., ZHUANG, J., TIAN, G., AND YANG, J. Emvirus: An embedding-based neural framework for human-virus protein-protein interactions prediction. *Biosafety and health 5*, 3 (2023), 152–158.

[119] YAMADA, K., AND HAMADA, M. Prediction of RNA-protein interactions using a nucleotide language model. *Bioinformatics Advances 2*, 1 (Apr. 2022), vbac023.

[120] YAO, Z., WEINBERG, Z., AND RUZZO, W. L. CMfinder–a covariance model based RNA motif finding algorithm. *Bioinformatics 22*, 4 (Feb. 2006), 445–452. Epub 2005 Dec 15.

[121] YE, Z., HUANG, J. X., AND LIN, H. Finding a good query-related topic for boosting pseudo-relevance feedback. *J. Assoc. Inf. Sci. Technol. 62*, 4 (2011), 748–760.

[122] YI, H.-C., YOU, Z.-H., HUANG, D.-S., AND KWOH, C. K. Graph representation learning in bioinformatics: trends, methods and applications. *Briefings in Bioinformatics 23*, 1 (2021), bbab340.

[123] YU, J., AND RUSSELL, J. Structural and functional analysis of an mrnp complex that mediates the high stability of human beta-globin mrna. *Molecular and Cellular Biology* (2001), 5879–5888.

[124] YU, Y., HE, W., JIN, J., XIAO, G., CUI, L., ZENG, R., AND WEI, L. iDNA-ABT: advanced deep learning model for detecting DNA methylation with adaptive features and transductive information maximization. *Bioinformatics 37*, 24 (10 2021), 4603–4610.

[125] YUAN GAO, D. G. Deep gate recurrent neural network. In *ACML* (2016).

[126] YUE, X., WANG, Z., HUANG, J., PARTHASARATHY, S., MOOSAVINASAB, S., HUANG, Y., LIN, S. M., ZHANG, W., ZHANG, P., AND SUN, H. Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics 36*, 4 (2019), 1241–1251.

[127] ZENG, M., ZHANG, X., LI, Y., LU, C., YIN, R., GUO, F., AND LI, M. Rnaloc-lm:RNA subcellular localization prediction using pre-trained RNA language model. *Bioinformatics 41*, 4 (2025), btaf127.

[128] ZHANG, C., LIU, C., ZHANG, X., AND ALMPANIDIS, G. An up-to-date comparison of state-of-the-art classification algorithms. pp. 128–150.

[129] ZHANG, J., FEI, Y., SUN, L., AND ZHANG, Q. C. Advances and opportunities in RNA structure experimental determination and computational modeling. *Nature Methods 19*, 10 (2022), 1193–1207.

[130] ZHANG, S., FAN, R., LIU, Y., CHEN, S., LIU, Q., AND ZENG, W. Applications of transformer-based language models in bioinformatics: a survey. *Bioinformatics Advances 3*, 1 (01 2023), vbad001.

[131] ZHAO, J., HUANG, J. X., AND HE, B. CRTER: using cross terms to enhance probabilistic information retrieval. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011* (2011), W. Ma, J. Nie, R. Baeza-Yates, T. Chua, and W. B. Croft, Eds., ACM, pp. 155–164.

[132] ZHAO, J., HUANG, J. X., AND YE, Z. Modeling term associations for probabilistic information retrieval. *ACM Trans. Inf. Syst. 32*, 2 (2014), 7:1–7:47.

[133] ZHOU, J., ZHAO, J., HUANG, J. X., HU, Q. V., AND HE, L. MASAD: A large-scale dataset for multimodal aspect-based sentiment analysis. *Neurocomputing 455* (2021), 47–58.

[134] ZHU, R., TU, X., AND XIANGJI HUANG, J. Chapter seven - deep learning on information retrieval and its applications. In *Deep Learning for Data Analytics*, H. Das, C. Pradhan, and N. Dey, Eds. Academic Press, 2020, pp. 125–153.

[135] ZHU-HONG YOU, XIAO LI, K. C. C. An improved sequence-based prediction protocol for protein-protein interactions using amino acids substitution matrix and rotation forest ensemble classifiers. *Neurocomputing 277* (2017), 228–2017.

[136] ZOU, Q., XING, P., WEI, L., AND LIU, B. Gene2vec: gene subsequence embedding for prediction of mammalian n6-methyladenosine sites from mrna. *RNA 25*, 2 (2019), 205–218.