MSDS 458: Artificial Intelligence and Deep Learning

Final Project

December 2, 2025

Predicting single guide RNA on-target efficacy in CRISPR-Cas9 system using neural networks

Yingyu Mao

Abstract

In this study, I investigated the use of modern neural networks, including Recurrent Neural Networks (RNNs), 1D Convolutional Neural Networks (1D CNNs), and Attention Mechanisms, to predict single guide RNA (sgRNA) on-target efficacy for the CRISPR-Cas9 system, aiming to eliminate manual feature engineering. Initial experiments confirmed that models trained on larger, more diverse datasets yield higher performance. Among the neural network architectures, the combination of bidirectional LSTM with a Multi-head Attention layer demonstrated superior prediction performance, achieving the highest Spearman correlation of 0.847 and the lowest MSE of 0.011. The Parallel 1D CNN also provided a strong, computationally efficient alternative, highlighting the importance of capturing both local k-mer patterns and long-range sequence dependencies. This study establishes that attention-based neural networks are the most effective models for *in silico* sgRNA prediction, providing a powerful guide for designing gene-editing experiments.

Keywords: recurrent neural network, 1D convolutional neural network, attention mechanism, single guide RNA, CRISPR-Cas9

1.0 Introduction

Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR), is a natural immune system in bacteria that has been adapted into a powerful gene-editing tool widely used for disease model generation, and genome-wide perturbation studies in biomedical research. CRISPR functions by using a single guide RNA (sgRNA) to direct the Cas9 nuclease to a specific genomic locus to perform editing. Not all sgRNAs are equally effective. Editing efficiency is influenced by a complex interplay of factors encoded in the sequence (made up of A,C,G,T/U bases), including nucleotide composition (such as GC content), position-specific effects (such as 3' complementarity has more weight than the 5'), thermodynamic properties (such as tendencies to form hairpins or self-dimers), and the presence of seed sequence that initiate RNA-DNA binding.

Suppose a gene-editing company developed a new variant of Cas9 that has enhanced activity and specificity. The management wants to build a machine learning model that reliably predicts on-target activity of sgRNAs specific to the new variant. This *in silico* drug screening step can greatly reduce the cost of the drug development process. Before generating a library of sgRNAs and running sequencing experiments, they want to find out how big a sgRNA library is needed, and which machine learning algorithm should be used to generate a robust model. I propose testing machine learning methods using publicly available datasets.

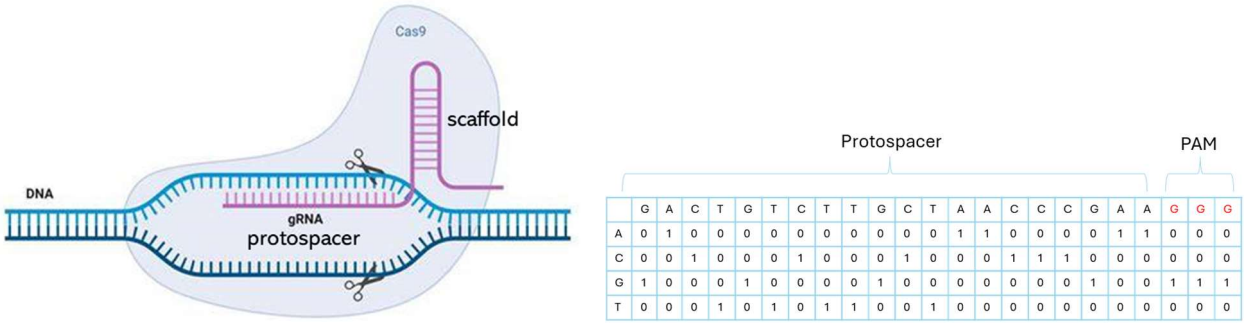
This study seeks to apply modern neural network architectures for learning sequential data, such as RNN, 1D CNN, and attention mechanism, to sgRNA efficiency prediction, with the goals of avoiding manual feature engineering that might miss nuance or long-range dependencies and providing preliminary results that guide model training that facilitates sgRNA design.

2.0 Literature review

Structurally, sgRNAs are made up of a protospacer region that guides Cas9 to the specific region of genomic DNA, followed by a protospacer adjacent motif (PAM) that marks the DNA cut site, and a scaffold sequence that embeds the sgRNA in the Cas9 protein (Figure 1 left). Early work on sgRNA on-target efficiency predication relied on manual featurization to engineer order 1 (mononucleotides, A/C/G/T) and order 2 (dinucleotides, AC/CG/TC...) sequence features in the protospacer region, GC count features, and computed melting temperatures. Traditional machine learning algorithms such as random forest regression, gradient boosted regression trees, and SVM with logistic regression had achieved above 0.5 in Spearman rank correlation for the predicted and true test values. Gradient boosted regression trees even achieved higher performance of 0.7 correlation with the held-out test set (Doench, et al. 2016). Later, Zarate et al. (2022) engineered similar k-mer (k=1,2,3) features in the protospacer region, the number of poly-T segments (defined as contiguous stretches of 3 or more Ts), four melting temperatures calculated from different regions of the 20 nt sgRNA spacer sequence and the first letter of the PAM. Using a combination of datasets from two different biological experiments as the training set, the authors trained a model, BoostMEC, based on a LightGBM regression algorithm that achieved similar performance on the held-out test set as Doench's model and performed slightly better than Doench's model on other new datasets. The reduced performance of both models on other datasets indicates strong variance common in biological experiments.

Manual feature engineering allows easy model interpretation that identifies important features crucial to understanding the mechanism of sgRNA-DNA-Cas9 interactions. However, it is not as useful in the application workflows that rely on quick ranking of all potential sgRNAs targeting a gene. Neural network architectures for sequential data analysis, such as RNN, 1D CNN and attention mechanisms have been

gradually adopted for this task with the advantage of eliminating the manual feature engineering step. Example models include DeepSpCas9 (Kim, et al. 2019) derived from GoogLeNet (2D CNN), a bidirectional RNN DeepHF (Wang, et al. 2019), and an ensemble model CRISPR_HNN that combined ResNet, RNN and attention mechanism (Li, et al. 2025).



	G	A	C	T	G	T	C	T	T	G	C	T	A	A	C	C	C	G	A	A	G	G	G
A	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0
C	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0
G	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	1
T	0	0	0	1	0	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Figure 1 sgRNA-DNA-Cas9 complex (left). A one-hot encoding example of the sequence of interest for sgRNA prediction.

3.0 Method

3.1 Data preprocessing and exploratory data analysis

Two datasets of different sizes were selected from published literature for this study. The SpCas9 dataset used for training DeepSpCas9 (Kim, et al. 2019) has 12833 entries of sgRNA sequences and their corresponding indel frequencies (a measure of DNA cutting efficiency through deep sequencing). The dataset generated by Wang et al. (2019) contains 55603 entries.

The nucleotides (nt) of the protospacer region (20 nt), and the first nucleotide of the PAM were parsed as separate tokens (A, C, G, T). The last two tokens of PAM are always GG, hence ignored. The distributions of the four nucleotides at each position in the two datasets are shown in the heatmaps in Figure 2. The proportions of the four nucleotides are generally even in most positions except for 1-2 positions in both datasets. The first token of the sequences in the Wange dataset was made to be either A or G, due to the biological reason that these two nucleotides promote U6 promoter activity that initiates sgRNA expression.

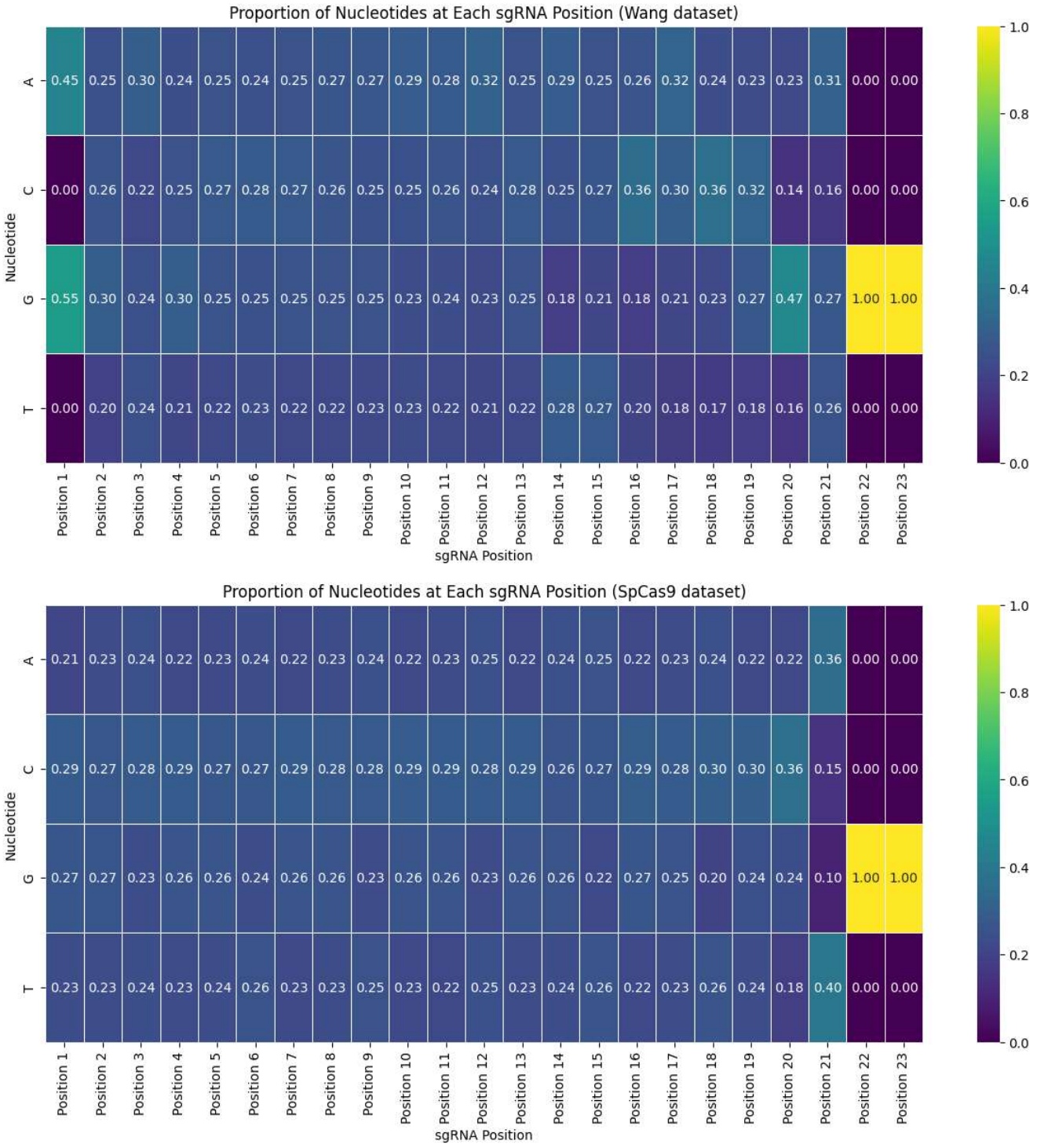


Figure 2 The nucleotide token proportion at each position in the two datasets.

Each type of nucleotide at each position was one-hot encoded (feature dimension = (84,)) for training a gradient boost regression baseline model. Label-encoded tokens were used as inputs for neural network models. A one-hot encoding layer (Figure 1 right) or a trainable embedding layer was included in the neural network architecture.

The distribution of the predictor variable, the indel frequency, of the two datasets were shown in Figure 3. The Wang dataset is dominated by sgRNA of high efficiency (>0.8), while the SpCas9 dataset has

more evenly distributed sgRNA with efficiency < 0.8, and low representation of sgRNAs with efficiency >0.8.

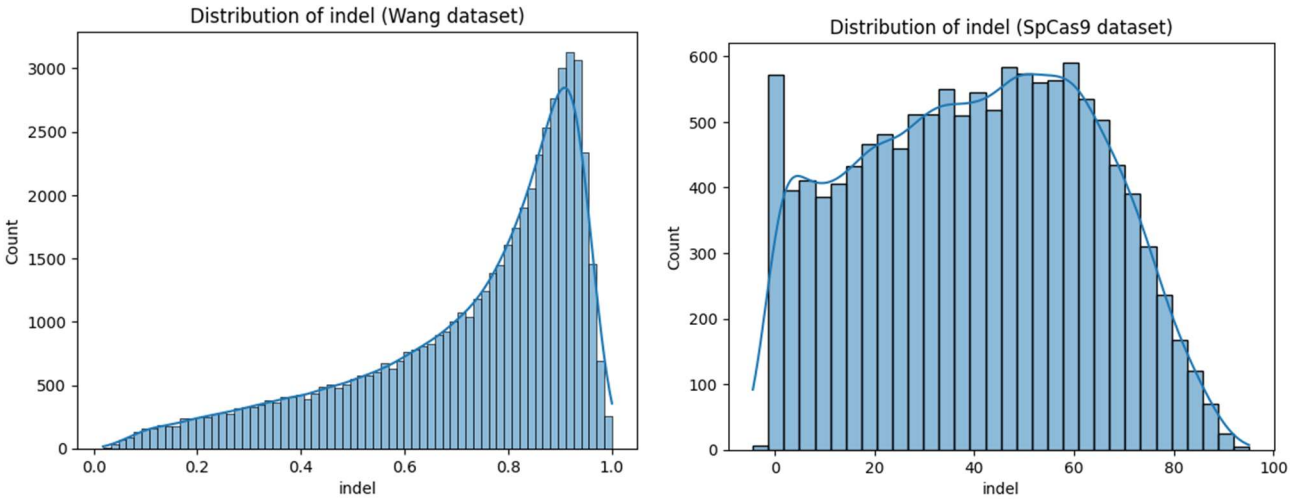


Figure 3 Distribution of indel frequencies in the two datasets.

The training, validation, and testing split at a ratio of 80:10:10 were stratified in accordance with the binned counts of the predictor variable to ensure the best representation of different sgRNA efficiency.

3.2 Experimental design

To understand the size of the training set on the learning outcome of independent datasets generated by separate experiments, I first used both datasets of different sizes (listed below) to train a gradient boost regressor, XGBoost (Chen and Guestrin 2016).

Dataset	**Training set size**	**Test set size**
Wang full	80% of 55603	10% of 55603
Wang subset	same as SpCas9	same as above
SpCas9	80% of 12833	10% of 12833

The Wang dataset was then used for training all neural network models, the architectures of which are listed below.

Models	**Architecture**
Bidirectional simple RNN	Input: label-encoded sequence (21 nt) Embedding: one-hot encoding Bidirectional SimpleRNN: 32 units Dense: 64 units Dense: 1 unit
Bidirectional LSTM (BiLSTM)	Input: label-encoded sequence (21 nt) Embedding: one-hot encoding Bidirectional LSTM: 32 units Dense: 64 units Dense: 1 unit
Bidirectional GRU	Input: label-encoded sequence (21 nt) Embedding: one-hot encoding Bidirectional GRU: 32 units

	Dense: 64 units Dense: 1 unit
BiLSTM with trainable embedding layer	Input: label-encoded sequence (21 nt) Embedding: 16/64 dimensions Bidirectional LSTM: 32 units Dense: 64 units Dense: 1 unit
1D CNN	Input: label-encoded sequence (21 nt) Embedding: 16 dimensions Conv1D: 32 filters, kernel size = 3 Dropout: 0.5 MaxPooling1D: pool size = 2 Flatten Dense-1: 64 units Dense-2: 1 unit
Parallel 1D CNN with different kernel size	Input: label-encoded sequence (21 nt) Embedding: 16 dimensions Parallel Conv1D: 32 filters, kernel size = [2,3,4,5,6] Flatten Dense-1: 64 units Dense-2: 1 unit
BiLSTM with attention layer	Input: label-encoded sequence (21 nt) Embedding: 16 dimensions Bidirectional LSTM: 32 units Attention Flatten Dense: 64 units Dense: 1 unit
BiLSTM with multi-head attention layer	Input: label-encoded sequence (21 nt) Embedding: 16 dimensions Bidirectional LSTM: 32 units Multi-head attention: num_heads = 8, key_dim = 32 Flatten Dropout: 0.4 Dense: 64 units Dense: 1 unit

All layers used default kernel initiation. All Dense and Conv1D layers used ReLU activation, while RNN units used tanh activation for output and sigmoid activation for recurrence. For performance assessment, I compared the training time (on Colab default CPU), mean squared error (MSE) of predicted sgRNA efficiency, the coefficient of determination (R^2) and the Spearman correlation between the predicted and true efficiency. R^2 indicates how much of the total variance of the predicated efficiency is explained by the true efficiency, while the Spearman rank correlation reveals the monotonic relationship between the predicted and true values. Ranking is essential for identifying candidate sgRNAs with high efficiency.

4.0 Model Training Results

All models were trained using TensorFlow default Nadam optimizer, a variant of Adam in combination with the Nesterov trick. The loss function used was the mean squared error.

Early stopping was used to prevent overfitting:

```
tf.keras.callbacks.EarlyStopping(monitor='val_accuracy', patience=3)
```

4.1 Experiment 1: Test the prediction performance of datasets with different sizes

The performance of XGBoost trained with different datasets are summarized below:

Dataset	MSE on held-out test set	R^2 on held-out test set	Spearman Correlation on held-out test set
Wang full	0.013	0.720	0.834
Wang subset	0.017	0.642	0.781
SpCas9	0.028	0.472	0.685

As expected, models built with smaller training sets performed worse. Their performance was also dataset/experiment dependent, as biological factors, such as sgRNA expression level, deep sequencing variations, can affect the indel frequency readout. The XGBoost model trained with the full Wang dataset achieved Spearman correlation of 0.624 on the SpCas9 dataset (Figure 4 left). There were about 13% data points that had high predicted efficiency (> 0.6) while their true efficiencies were < 0.4 (false positives), comparing to only 1.4% of the Wang held-out test set. This discrepancy may come from biological variances between experiments and differences of the predictor variable distribution which results in limited performance on out-of-distribution predictions. To prepare the training data for our business problem, one should consider the sgRNA expression level, variance of deep sequencing experiments, and ensure a good coverage of sgRNA efficiency at all levels, besides generating a large training dataset.

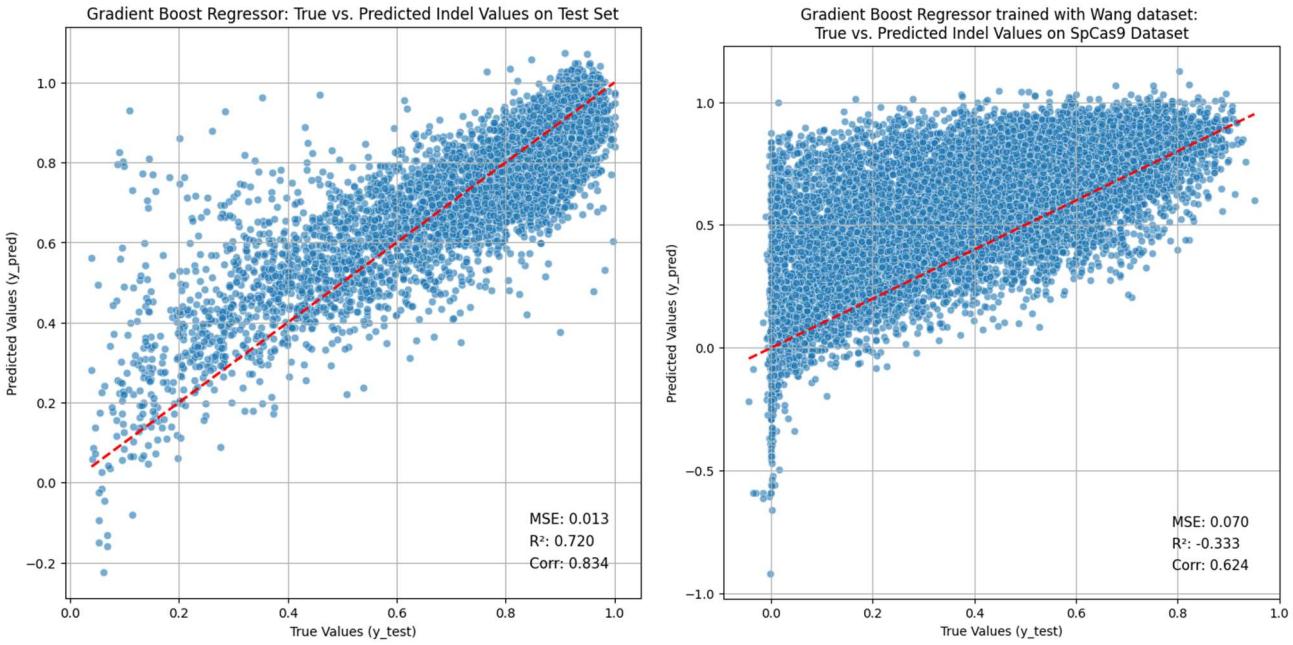


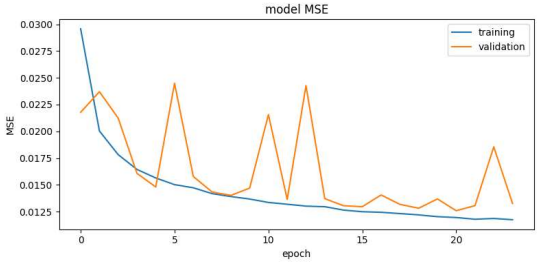
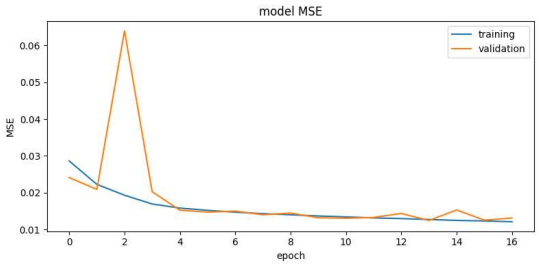
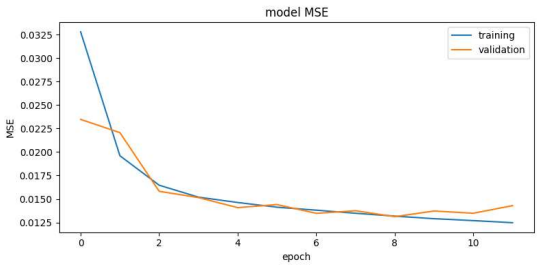
Figure 4 True vs predicted indel frequencies of gradient boost regressor trained on the full Wang dataset. (Left) performance on the held-out test set. (Right) performance on the SpCas9 dataset.

Since the Wang dataset has better coverage of all efficiency level and has sufficient training data, I performed neural network training and evaluations on this dataset.

4.2 Experiment 2: bidirectional simple RNN, LSTM, GRU

The RNA sequence data is time-independent, which means the context could be extracted from either direction. Bidirectional recurrent architectures, including simple RNN, LSTM, and GRU, were compared. The summary of their performance can be found in Table 1.

Table 1 Performance summary of Bidirectional RNNs.

Experiments	Training time	Training history	Performance on test set
Bidirectional simple RNN	6min 33s		MSE: 0.013 R^2: 0.730 Correlation: 0.827 Number of false positives: 73
BiLSTM	6min 32s		MSE: 0.013 R^2: 0.728 Correlation: 0.827 Number of false positives: 79
Bidirectional GRU	7min 14s		MSE: 0.014 R^2: 0.714 Correlation: 0.817 Number of false positives: 79

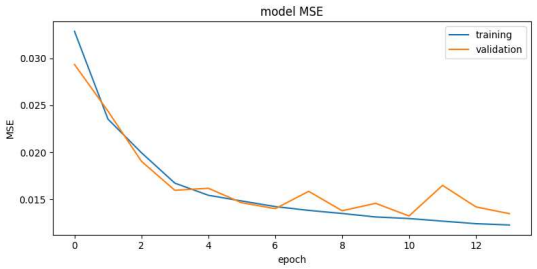
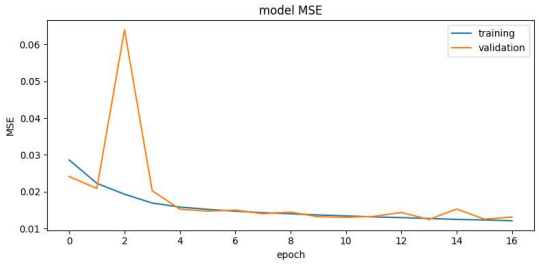
With one-hot encoded tokens, the bidirectional simple RNN and LSTM had similar performance that was better than GRU.

4.3 Experiment 3: BiLSTM with trainable embeddings

Swapping out the one-hot encoding layer with a 16- or 64-dimensional embedding layer allows richer information to represent the four nucleotides. With the same underlying BiLSTM architecture, the effect of embedding dimensions was assessed and shown in Table 2.

Table 2 Performance summary of BiLSTM with different trainable embeddings.

Experiments	Training time	Training history	Performance on test set

BiLSTM with 16-dimension embedding	6min 17s		MSE: 0.013 R^2: 0.725 Correlation: 0.828 Number of false positives: 64
BiLSTM with 64-dimension embedding	11min 6s		MSE: 0.013 R^2: 0.734 Correlation: 0.830 Number of false positives: 71

The 64-dimensional embedding needed almost twice the training time without generating much improvement on the overall prediction performance. The subsequent models used the 16-dimensional embedding layer.

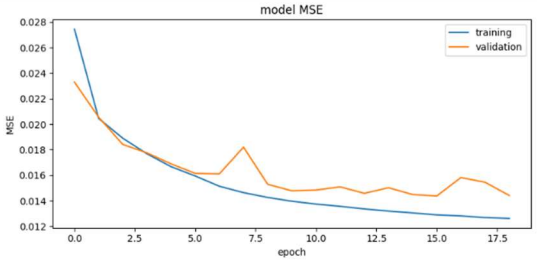
4.4 Experiment 4: 1D CNN architectures

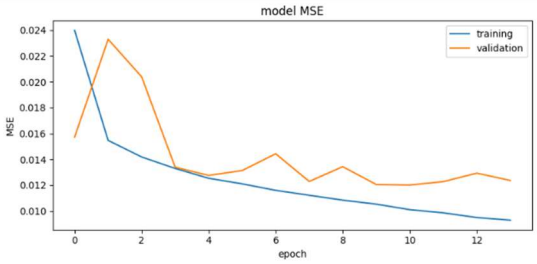
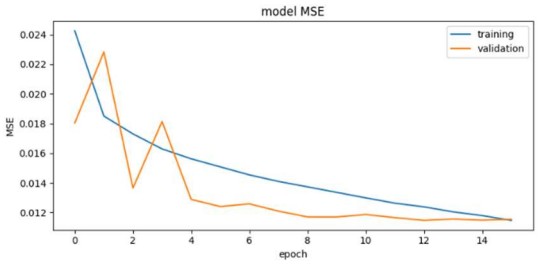
Like bidirectional RNNs, 1D CNN learns the context independent of the sequence direction. The kernel size of 1D CNN controls the level of granularity each neuron focuses on. In the case of RNA sequence, it captures local relationships within different k-mers of nucleotides (corresponds to n-grams of tokens in text sequences). With kernel size 1 and stride 1, the network functions as a fully connected model. As comparison, this model performed worse than a 1D CNN with kernel size 3 (Table 3). This indicates that higher order motifs are likely more important than single tokens in determining the sgRNA efficiency.

To capture different levels of granularity of the RNA sequence, I then trained a parallel 1D CNN with different kernel sizes. This model improved both the R^2 and the Spearman correlation. Although there were similar false positives as the simple 1D CNN (based on the criteria mentioned above), the parallel model was effective at reducing false negatives.

Even the simple 1D CNN had better performance than all the RNNs tested, suggesting that learning local patterns through kernels extracts more relevant information than nonlinearly transformed weighted sum of all hidden states in an RNN. In addition, its training time was only 1/3 of that of a simple RNN or LSTM, as CNNs are parallelizable and independent of time steps.

Table 3 Performance summary of Bidirectional 1D CNNs.

Experiments	**Training time**	**Training history**	**Performance on test set**
1D CNN kernel size 1 (like a fully connected NN)	2min 19s		MSE: 0.015 R^2: 0.695 Correlation: 0.813 Number of false positives: 78

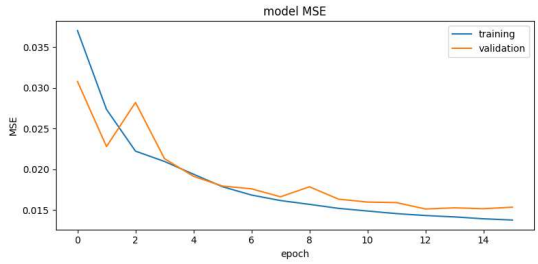
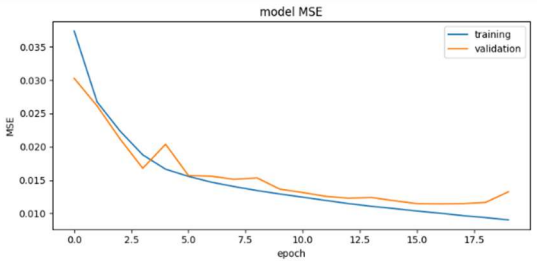
1D CNN kernel size 3	1min 39s		MSE: 0.013 R^2: 0.735 Correlation: 0.833 Number of false positives: 64
Parallel 1D CNN with different kernel size	4min 18s		MSE: 0.012 R^2: 0.753 Correlation: 0.841 Number of false positives: 65

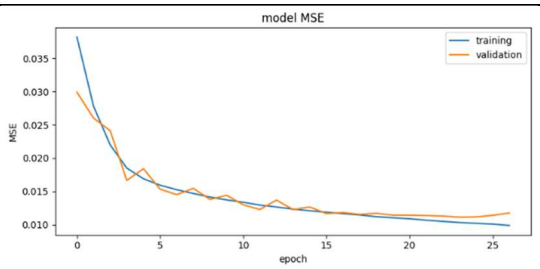
4.5 Experiment 5: LSTM in combination with self-attention mechanism

To address the limitations of RNNs, I then tested whether addition of attention mechanism will improve its performance, with the idea that dot product self-attention enables the model to focus on most similar positions in the sequence. Translating it into the biological context, this means to have the model learn representations of different parts of the RNA sequence that might have similar functions in sgRNA-Cas9-DNA complex formation and subsequent cutting. Unlike 1D CNN which focuses on motifs formed by local k-mers, self-attention mechanism learns positional dependences that can be adjacent to each other or several nucleotides away. The attention weights transform the LSTM sequence output to incorporate context information.

Adding a default TensorFlow `Attention` layer after the LSTM layer resulted in a model with better prediction outcome although it took twice as long to train (Table 4). As a comparison, a model with BiLSTM layer replaced by the attention layer (attention only) was also evaluated.

Table 4 Performance of LSTM in combination with self-attention mechanism.

Experiments	Training time	Training history	Performance on test set
Attention only	2min 2s		MSE: 0.015 R^2: 0.677 Correlation: 0.799 Number of false positives: 107
BiLSTM with attention layer	12min 14s		MSE: 0.012 R^2: 0.754 Correlation: 0.844 Number of false positives: 62

BiLSTM with multi-head attention layer	27min 22s		MSE: 0.011 R^2: 0.767 Correlation: 0.847 Number of false positives: 60

In text processing, a word in a sentence can have different characteristics, such as function (verb, noun, adjective), or tense (present, past, future), or position. Learning all characteristics in one shot with one attention layer misses the nuances between different characteristics that need more dimensions to accurately represent. Multi-head attention is parallelization of scaled dot-product attention (Vaswani, et al. 2017) which allows each head to focus on a different characteristic of a token. In the biological context, the different characteristics of a token could be translated into the different functions of nucleotides in the process of CRISPR-Cas9 induced DNA editing, such as mediating RNA-DNA hybrid formation, sgRNA folding, or facilitating the nuclease activity.

Adding an 8-head attention layer after the LSTM layer resulted in a model with even better prediction outcome although with longer training time (Table 4). Both BiLSTM-attention models performed better than the parallel 1D CNN model.

4.6 Visualize outputs of hidden layers

To understand how the embeddings at different hidden layers affect the model performance, I then used t-SNE to reduce the hidden feature dimensions and visualized the relationships between the data points in the test set. I compared the hidden layer output of three models: the LSTM layer from the BiLSTM model with a trainable embedding layer, the flattened layer of the parallel 1D CNN, and the flattened layer of the LSTM with multi-head attention. Figure 5 shows the t-SNE projection of the data distribution into a 2-dimensional latent space, with the data points colored by the true sgRNA efficiency. The embeddings coming out of the BiLSTM layer resulted in local clustering of the data: sgRNA with high efficiency were scattered into multiple different clusters, and some clusters contained a mixture of sgRNA with low, mid, and high efficiency. This means that the hidden features in the embeddings did not effectively capture some differences between sgRNAs of high and low efficiencies. In addition, since the prediction was not a class but a value of any real number, the data distribution should be continuous instead of forming discrete clusters.

On the contrary, the data projections from the parallel 1D CNN and LSTM with multi-head attention both formed a shape of Gaussian distribution. With the embeddings from the multi-head attention model, the data points of low sgRNA efficiencies were aggregated more closely and separated from those of high sgRNA efficiencies. This visualization explains the performance differences among these models: the BiLSTM model has worse performance, while the multi-head attention model outperformed the others.

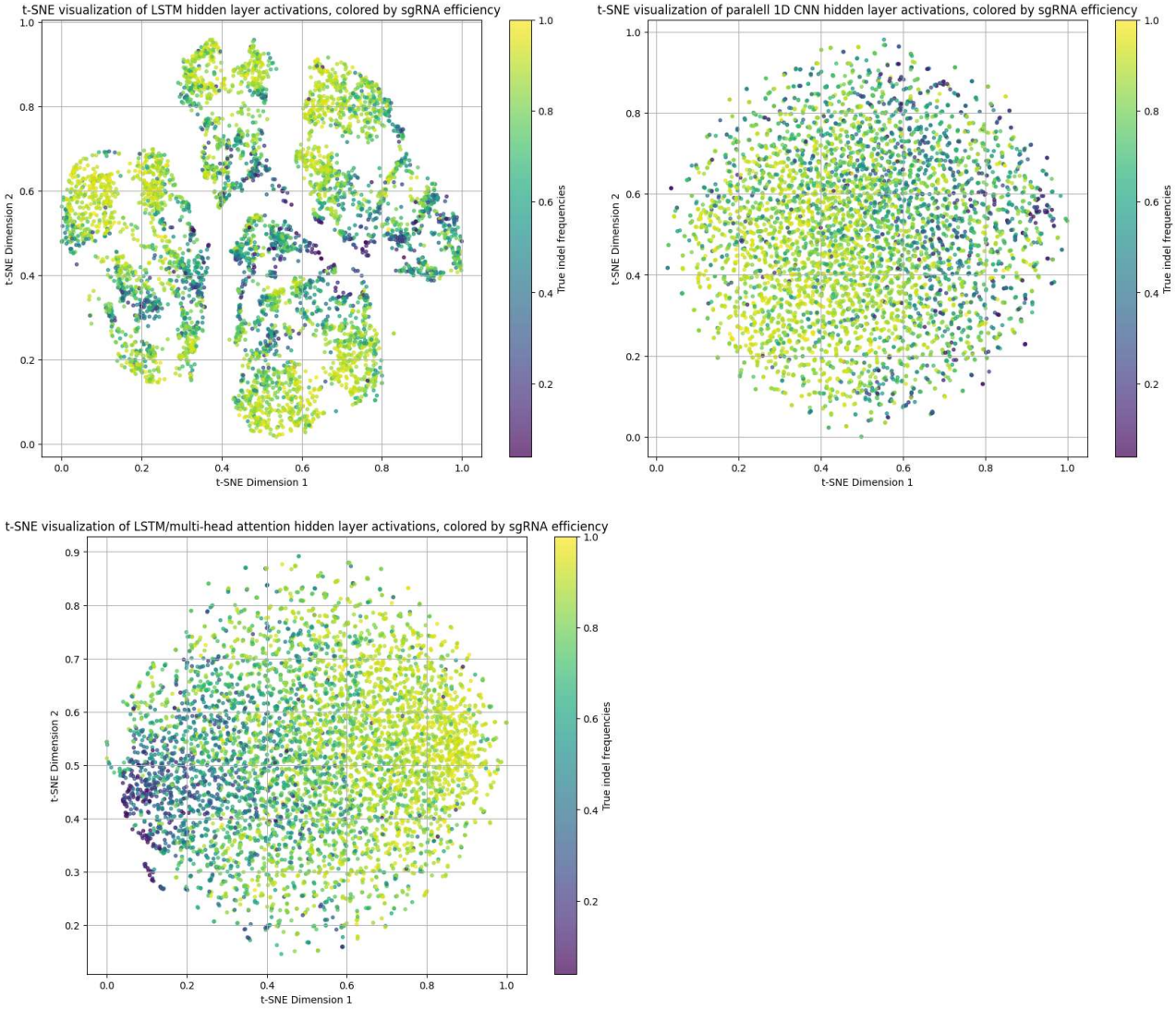


Figure 5 Dimension reduction of hidden features using t-SNE. Colors of data points indicate the true indel frequency (sgRNA efficiency).

5.0 Conclusions

I applied various modern neural network architectures, including Recurrent Neural Networks (RNNs), 1D Convolutional Neural Networks (1D CNNs), and Attention Mechanisms, to the task of predicting single guide RNA on-target efficiency for the CRISPR-Cas9 system. Initial experiments with the baseline XGBoost regressor demonstrated that models trained on larger datasets (Wang full, 55,603 entries) significantly outperformed those trained on smaller datasets (SpCas9, 12,833 entries), achieving a Spearman correlation of 0.834 on its held-out test set. Critically, the XGBoost model showed limited performance when applied to the out-of-distribution SpCas9 dataset, highlighting the strong variance between biological experiments and the need for diverse training data. The best performance among all neural network models was achieved by the Bidirectional Long Short-Term Memory (BiLSTM) layer followed by a Multi-head Attention layer, which yielded the highest Spearman correlation of 0.847.

The comparison of neural network architectures provided some insights into how they process sgRNA sequence data. The 1D CNNs demonstrated a significant advantage in training speed and competitive performance, suggesting that capturing local sequence patterns (k-mers) is highly effective for

determining sgRNA efficiency. The Parallel 1D CNN further improved performance by capturing multiple levels of granularity simultaneously. The biggest performance boost came from incorporating the Multi-head Attention mechanism with BiLSTM, which outperformed the parallel CNN. This suggests that learning long-range dependencies and focusing on the different functional characteristics of nucleotides, such as their roles in RNA-DNA hybridization or Cas9 nuclease activity, is crucial for high-accuracy prediction. Visualization via t-SNE reveals the inner workings of several models: the hidden embeddings from the Multi-head Attention model created the most distinct separation between high and low-efficiency sgRNAs, indicating better learned features compared to the clustering produced by the simple BiLSTM model.

Based on these results, I recommend the following guidelines to the management for their *in silico* drug screening efforts:

- **Training Data Requirement:** To build a robust machine learning model for the new Cas9 variant, the company should aim to generate a large and diverse library of sgRNAs with efficiency data, ensuring good coverage of sgRNA efficiency at all levels. Accounting for biological variance between sequencing experiments might also be needed to generate a diverse training dataset from multiple rounds of sequencing.
- **Optimal Model Selection:** The BiLSTM with Multi-head Attention model should be selected as the foundational architecture for predicting on-target activity due to its superior performance in both ranking (Spearman Corr = 0.847) and overall prediction error. Its ability to capture complex sequence relationships is essential for reliability.
- **Alternative for Speed**: If training time is a critical constraint, the Parallel 1D CNN offers an excellent balance of performance (Spearman Corr = 0.841) and significantly faster training.

References

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." *arXiv* 1603.02754.

Doench, John G, Nicolo Fusi, Meagan Sullender, Mudra Hegde, Emma W Vaimberg, Katherine F Donovan, Ian Smith, et al. 2016. "Optimized sgrNA design to maximize activity and minimize off-target effects of crisPr-cas9." *Nature Biotechnology* VOLUME 34 NUMBER 2: 184-195.

Kim, Hui Kwon, Younggwang Kim, Sungtae Lee, Seonwoo Min, Jung Yoon Bae, Jae Woo Choi, and Dongm Jinman Park. 2019. "SpCas9 activity prediction by DeepSpCas9, a deep learning–based model with high generalization performance." *Sci. Adv.* 5,eaax9249.

Li, C, Q Zou, J Li, and H Feng. 2025. "Prediction of CRISPR-Cas9 on-target activity based on a hybrid neural network." *Comput Struct Biotechnol J.* 27:2098-2106.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." *arXiv* 1706.03762.

Wang, D, C Zhang, B Wang, B Li, Q Wang, D Liu, H Wang, et al. 2019. "Optimized CRISPR guide RNA design for two high-fidelity Cas9 variants by deep learning." *Nat Commun.* 10(1):4284.

Zarate, Oscar A., Yiben Yang, Xiaozhong Wang, and Ji-Ping Wang. 2022. "BoostMEC: predicting CRISPR-Cas9 cleavage efficiency through boosting models." *BMC Bioinformatics* Volume 23, article number 446.