Predicting TCR-Epitope Binding with Fine-Tuned Protein Language Models and Contrastive Learning

Yichuan Zhang
Carnegie Mellon University
Pittsburgh, PA 15213
yichuan2@andrew.cmu.edu

Shubham Kachroo
Carnegie Mellon University
Pittsburgh, PA 15213
skachroo@andrew.cmu.edu

Isaac Martinotti
Carnegie Mellon University
Pittsburgh, PA 15213
imartino@andrew.cmu.edu

Adam Hunt
Carnegie Mellon University
Pittsburgh, PA 15213
aahunt@andrew.cmu.edu

Abstract

T-cell receptor (TCR) binding prediction is a critical task in immunotherapy and vaccine development, yet it remains challenging due to the diversity of TCR-epitope interactions. This study introduces a novel approach that integrates contrastive learning with fine-tuned protein language models, such as ESM-2, to enhance classical binding prediction accuracy. We investigate various model extensions, including fine-tuning strategies (e.g., top-layer fine-tuning, LoRA, and adapter tuning), structural features of molecules (SMILES), and inference techniques tailored for TCR-epitope binding prediction. This work aims to contribute to a growing body of research that advances our understanding of TCR-epitope interactions. Our experiments demonstrate that combining contrastive learning with fine-tuned protein language models achieves robust predictive performance, unlocking new possibilities for advancing immunotherapy research. The complete codebase for this study is publicly available at **https://github.com/Yichuan0712/11785-TCR**.

1 Introduction

T-cell receptors (TCRs) are fundamental components of the adaptive immune system, playing a primary role in the recognition of antigens and in the initialization of immune responses. Located on the surface of T-cells, TCRs bind to antigenic peptides, enabling the immune system t1 o detect and respond to foreign pathogens [1]. The interaction occurs at the complementary-determining regions (CDRs) of the TCR [2], which constitutes the antigen-binding sites, and their diversity generated through recombination and hyper-mutation allows TCRs to recognize a vast array of antigens. Understanding the relationship between the amino acid sequence of TCRs is crucial as the sequence dictates the antigen specificity and binding affinity. Studies have shown that even subtle changes in the TCR sequence can lead to significant differences in antigen recognition, emphasizing the importance of sequence-structure relationships in immune responses [3] and their therapies.

The specificity of TCR-antigen interactions is central to immune function as it determines the ability of T-cells to distinguish between self and non-self molecules. Antigens are processed into smaller fragments called epitopes - the specific regions recognized by TCRs. Thus, predicting whether an epitope will effectively bind to specific TCRs is a key challenge in immunology due to the immense diversity of both TCRs and epitopes. Accurate prediction of TCR-epitope interactions has significant applications in vaccine development, immunotherapy and, in particular, the identification of specific isotopes created by tumor cells can lead to the development of T-cell therapies that specifically target cancerous cells without harming normal tissues [4].

In this study, we propose a novel approach that combines contrastive learning with fine-tuned protein language models to improve the accuracy of TCR-epitope binding prediction. By fine-tuning models like ESM-2 [5] within a contrastive learning framework, we aim to capture the underlying patterns in TCR sequences and their binding isotopes. Specifically, we focus on the antigen-binding sequences to train the model on the most critical regions involved in antigen recognition. This approach leverages the strengths of protein language models in understanding sequence data and the advantages of contrastive learning in representation learning. By doing so, we enhance the model's ability to generalize to unseen TCRs, providing a more robust and comprehensive prediction system. This study aims to advance predictive modeling in bioinformatics and seeks to bridge gaps that could impact vaccine development and immunotherapy, providing tools that can accelerate personalized treatment strategies.

2 Literature Review

Traditional methods for predicting TCR binding have primarily relied on analyzing amino acid sequences to estimate binding affinity to specific antigens. Tools like TCRmatch [6] and NetTCR [7] use sequence similarity to make their predictions. However, these models often fall short in fully capturing the diversity and complexity of TCR-epitope interactions. Methods relying on simple CNNs and transformers often oversimplify these intricate dynamics, leading to suboptimal representation quality and reduced predictive accuracy in real-world scenarios.

To improve the quality of representations from protein sequences, a major advancement in the field of bioinformatics has been the development of protein language models [8], particularly ESM-2 [5]. These models have demonstrated strong capabilities in encoding rich, meaningful representations from protein sequences and have set new benchmarks across a variety of biological tasks. By training on vast amounts of protein data, ESM-2 captures both local and global sequence patterns, making it potentially effective for TCR binding prediction. Its ability to manage the complexities of biological data makes it a foundational tool for improving prediction models that rely solely on sequence information. To further enhance the performance of protein language models like ESM-2, several fine-tuning strategies can be considered, such as fine-tuning the top layers [9], LoRA (Low-Rank Adaptation) [10], and adapter tuning [11]. These methods offer ways to improve the adaptability and efficiency of protein language models, making them better suited for specific tasks like TCR binding prediction and potentially improving overall prediction accuracy.

To address the challenges posed by the vast diversity of proteins, we use contrastive learning [12], which has proven particularly effective in bioinformatics, especially for tasks like protein function classification. Methods such as CLEAN [13] have demonstrated success in handling large, diverse bioinformatics datasets. By bringing similar data points closer and pushing dissimilar ones apart, contrastive learning enables models to differentiate subtle variations across classes. In the context of TCR binding prediction [14] and clustering [15], contrastive learning has the potential to enhance the model's ability to capture complex patterns and improve generalization, particularly for unseen TCR-epitope interactions.

Several recent studies have applied contrastive learning techniques to TCR research, adopting varied approaches and objectives. Fang et al. introduced the ATMTCR model [14], which incorporates an attention-based encoder with contrastive learning to predict TCR-antigen binding specificity. Despite its improved performance on TCR-pMHC binding datasets by highlighting critical amino acids, ATMTCR's methodology is somewhat dated, potentially limiting its current applicability. The TouCAN model [15] employs a combination of ESM embeddings and triplet loss to cluster TCR sequences by antigen specificity, which effectively groups TCRs with different sequences into antigen-specific clusters. However, TouCAN did not explore various efficient fine-tuning techniques, which could potentially improve clustering performance, and its complex dataset poses limitations, as comparable high-quality data are often challenging to source. More recently, Nagano et al. developed SCEPTR [16], a small-scale TCR-specific language model that leverages autocontrastive learning. SCEPTR focuses on generating high-quality embeddings with minimal computational overhead, yet it does not target downstream tasks like binding specificity prediction. Although these studies represent significant advancements, differences in datasets, model architecture, and specific tasks indicate the need for further research to evaluate their direct applicability or comparability to our tasks and methods.

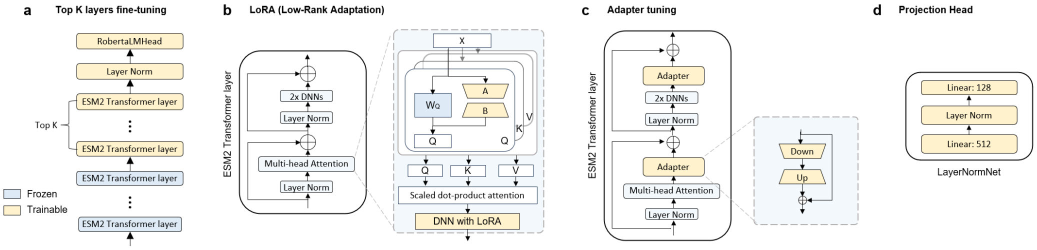

Figure 1: Architectural Variants for Fine-Tuning ESM2 with Projection Head Design

In summary, predicting TCR-epitope binding is challenging due to the diversity of TCRs and epitopes. Traditional sequence-based methods struggle with feature extraction and generalization, particularly in real-world settings. Advances like protein language models and contrastive learning offer promising solutions. By incorporating fine-tuning techniques such as LoRA and adapter tuning, these models can become more robust and accurate in capturing TCR-epitope interactions, holding great potential for applications in immunotherapy and personalized medicine.

3 Model Description

The architecture of our implemented model for predicting TCR-epitope binding comprises three primary components: (1) an encoder based on the ESM-2 protein language model [5], (2) a projection head that maps the encoder outputs to a lower-dimensional embedding space optimized for contrastive learning, and (3) a prediction model that takes these embeddings as input to determine binding probability. Additionally, we explore various fine-tuning strategies for the encoder and experiment with different models for the final prediction task, including MLPs, CNNs, and Extreme Gradient Boosting (XGBoost).

3.1 Encoder Architecture

Our encoder leverages the ESM-2 (esm2_t33_650M_UR50D) model, which is a state-of-the-art protein language model known for its ability to generate rich representations of protein sequences. The default encoder architecture is detailed in Table 1. It consists of an embedding layer followed by 33 Transformer layers, each with multi-head attention mechanisms and feed-forward networks, culminating in a LayerNorm layer.

Table 1: Default Encoder Architecture

Layer	Output Shape	Parameters
Embedding	[1, 1280, 1280]	42,240
Transformer Layers (33x)	[1280, 1, 1280]	19,677,440 each
Linear	[1, 1280, 1280]	1,639,680
LayerNorm	[1280, 1, 1280]	2,560

3.2 Encoder Adjustments for Fine-Tuning

We explore three fine-tuning strategies to adapt the encoder to our specific task:

- **Fine-Tuning Last Layers**: We fine-tune the last few Transformer layers of the ESM-2 model while keeping the rest frozen. This allows the model to adapt higher-level representations to our task without overfitting.
- **Adapter Layers**: We introduce adapter layers [11] In the Transformer architecture, adapter layers are lightweight modules inserted within each Transformer layer, enabling task-specific fine-tuning with a minimal number of additional parameters. As an example with $N = 12$, adapter layers are applied starting from layer $34 - N$ (i.e., layer 22) onwards, up to layer

33. This approach ensures that only the top N layers of the Transformer are enhanced with adapters, leaving the rest of the architecture unchanged.

Table 2: Encoder Architecture with Adapter Layers

Layer	**Output Shape**	**Parameters**
Transformer Adapter Layers (Lower Layers)	[1280, 1, 1280]	19,677,440 each
Transformer Adapter Layers (Top N Layers)	[1280, 1, 1280]	22,963,200 each

- **Low-Rank Adaptation (LoRA)**: We apply LoRA [10] to the attention weights of the Transformer layers. LoRA reduces the number of trainable parameters by decomposing the weight updates into low-rank matrices, significantly decreasing computational requirements while maintaining performance.

3.3 Projection Head

The projection head maps the high-dimensional outputs of the encoder to a lower-dimensional space suitable for contrastive learning. The architecture of the projection head is presented in Table 3. It consists of two fully connected layers with LayerNorm and Dropout, followed by a final linear layer that outputs 128-dimensional embeddings.

Table 3: Projection Head Architecture

Layer	**Output Shape**	**Parameters**
Linear Layer 1	[1, 512]	655,872
LayerNorm	[1, 512]	1,024
Dropout	[1, 512]	–
Linear Layer 2	[1, 512]	262,656
LayerNorm	[1, 512]	1,024
Dropout	[1, 512]	–
Linear Layer 3	[1, 128]	65,664

3.4 Prediction Models

For the final prediction of TCR-epitope binding, we experiment with three different models that take the embeddings generated by the encoder and projection head, as shown in Table 4:

- **Extreme Gradient Boosting (XGBoost)**: A tree-based ensemble method effective for structured data classification. It does not utilize hidden layers, and the model is tuned with 100 estimators and a maximum tree depth of 5 for optimal performance.
- **Multi-Layer Perceptron (MLP)**: A neural network with 5 fully connected (dense) layers of sizes 64, 128, 256, 128, and 64, using ReLU activation. The model employs adaptive learning rates and early stopping to enhance performance.
- **Convolutional Neural Network (CNN)**: A model with 3 convolutional blocks, each consisting of Conv1D, Batch Normalization, Max Pooling, and Dropout layers. The architecture is followed by a Flatten layer and two fully connected layers with Dropout. The filters are set to 64, 128, and 256, with kernel size 3 and dense layers of size 512 and 256, leading to an output layer.

4 Dataset Description

The dataset utilized in this paper is derived from the Weber et al. dataset, a well-known resource for TCR-epitope binding prediction within the Therapeutics Data Commons (TDC) framework [17]. Below are some of the key columns in the dataset.

Table 4: Prediction Models

Model	Hidden Layers	Hyperparameters
XGB	No Hidden Layers	N Estimators 100, Max Depth 5
MLP	5 × Dense	ReLU Activation Dense (64, 128, 256, 128, 64)
CNN	3 × {Conv1D, BatchNorm, MaxPool, Dropout}, Flatten, 2 × {Dense, Dropout}	Filters (64, 128, 256), Kernel Size (3) Dropout (0.3, 0.5), Dense (512, 256)

- **epitope_aa**: Represents the amino acid sequences of the epitopes. These are short peptides presented on the surface of cells by Major Histocompatibility Complex molecules, allowing the immune system to recognize foreign invaders [18].
- **epitope_smi**: Provides SMILES (Simplified Molecular Input Line Entry System [19]) notation of epitope's structure. SMILES gives compact machine-readable representations of chemical molecules that support the exploration of structural similarities and binding properties [20].
- **tcr**: Represents the binding site sequence of T-Cell Receptors, which plays a key role in antigen recognition [21].
- **tcr_full**: Includes the full receptor sequence, covering variable, diversity, and joining gene segments, providing a more comprehensive view of TCRs [21].
- **label**: Binary indicator where 1 denotes a successful binding interaction between a TCR and an epitope, and 0 indicates no binding.

5 Evaluation Metric

We evaluate the models using four metrics:

- **Accuracy**: Measures the proportion of correctly predicted TCR-epitope bindings.
- **AUROC**: Evaluates the model's ability to distinguish binding from non-binding pairs across thresholds.
- **AUPR**: Assesses precision and recall to highlight the model's performance in identifying binding pairs without excessive misclassification.
- **F1 Score**: Provides a balanced measure of precision and recall for binding predictions.

6 Loss Function

We employed two contrastive loss functions during training:

The Triplet Margin Loss selects an anchor embedding z_a, a positive embedding z_p from the same class, and a negative embedding z_n from a different class [22]. The loss is defined as:

$$\mathcal{L}^{TM} = \|z_a - z_p\|_2 - \|z_a - z_n\|_2 + \alpha \tag{1}$$

where α is the margin hyperparameter, set to 1 in our experiments.

The SupCon-Hard Loss is a modification of the supervised contrastive loss [22], tailored to use a fixed number of hard positives and negatives per batch, similar to the approach in the CLEAN architecture [9]. For each class e in the set of classes E, we define $P(e)$ as the set of positive samples and $N(e)$ as the set of hard negative samples. The loss function is formulated as:

$$\mathcal{L}^{sup} = \sum_{e \in E} \frac{-1}{|P(e)|} \sum_{z_p \in P(e)} \log \frac{\exp(z_e \cdot z_p / \tau)}{\sum_{z_a \in A(e)} \exp(z_e \cdot z_a / \tau)} \tag{2}$$

where $A(e) = P(e) \bigcup N(e)$ and τ is the temperature parameter, set to 0.1. The embeddings z are L2-normalized to unit length to stabilize training.

7 Baseline Selection and Implementation

7.1 CLEAN-Based Approach: Adapting Enzyme Function Prediction Methods to TCR-Epitope Binding with Contrastive Learning

Our approach draws inspiration from the CLEAN method [13], a highly successful contrastive learning model in bioinformatics. Originally designed for enzyme function prediction, CLEAN's architecture offers valuable insights for protein-related tasks like TCR-epitope binding prediction. Its strong performance on datasets with limited information underscores its ability to generalize effectively to sparse datasets, including ours. Building on this foundation, we fine-tune ESM-2 to refine TCR embeddings and apply contrastive learning to separate binding from non-binding TCR-epitope pairs, tailoring CLEAN's framework to get high quality TCR sequence representations.

7.2 Reimplementing CLEAN with Enhanced Batch Flexibility

To ensure a robust foundation and enable future improvements, we have reimplemented the full functionality of the CLEAN method. Our code can be found here **https://github.com/Yichuan0712/11785-TCR**. Starting from the CLEAN architecture, we integrated the ESM-2 model directly into our implementation. Unlike CLEAN, which only uses the official ESM-2 model to generate embeddings without including the model as part of their program, our approach embeds ESM-2 within the framework. In addition, we added adapter layers, trainable top layers, and LoRA-related code, enhancing the adaptability and fine-tuning capabilities of the model specifically for TCR-epitope binding tasks. We also improved CLEAN's data loading strategy. CLEAN samples each batch according to class types, which works well for large datasets. Building on this, we implemented regular batch sampling, offering greater flexibility and adaptability for smaller or less balanced datasets.

7.3 VIBTCR: Attentive Variational Information Bottleneck for TCR–peptide interaction prediction

To compare our approach with TCR binding specificity prediction methods, we selected VIBTCR as an additional baseline. The Attentive Variational Information Bottleneck (AVIB) model [23] advances TCR-epitope interaction prediction by incorporating multi-head self-attention into the classical Variational Information Bottleneck (VIB) framework, capturing complex interdependencies across multiple input sequences. Its Attention of Experts (AoE) mechanism integrates latent encodings from individual sequences into a unified multi-sequence representation, enabling AVIB to handle missing input sequences without significant performance loss. This robust approach effectively predicts TCR-epitope binding while addressing the inherent diversity of biological data.

8 Explored Model Extensions

8.1 Top Layer Fine-Tuning

To efficiently adapt ESM-2 for TCR-epitope binding prediction while maintaining its pre-trained knowledge of protein sequences, we implemented top-layer fine-tuning. This approach involves updating only the top K Transformer layers of the ESM-2 backbone while keeping the lower layers unchanged. This method significantly reduces the computational requirements compared to full-model fine-tuning while helping prevent overfitting and alteration of the heavily specialized foundational protein knowledge. The value of K serves as a hyperparameter, allowing us to balance computational efficiency with task-specific performance optimization.

8.2 LoRA

Low-rank Adaptation (LoRA) offers a parameter-efficient approach to fine-tuning ESM-2 by introducing trainable low-rank matrices while keeping original pre-trained weights frozen [10]. Instead of updating the entire weight matrices, LoRA represents updates through low-rank decomposition:

$$\Delta W = BA$$

where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, with $r \ll \min(d, k)$. Here, $W_0 \in \mathbb{R}^{d \times k}$ represents the original weight matrix. During training, only A and B are updated, substantially reducing the number of trainable parameters, thus reducing the overhead and time to converge. For an input x, the forward pass with LoRA is expressed as:

$$h = W_0 + \Delta W x = W_0 + BAx$$

In practice, $\Delta W x$ is scaled by $\frac{\alpha}{r}$, where α is a constant controlling the scale factor. We apply the low-rank decomposition matrices to the input, key, value, and output projection matrices within the self-attention layers of the top K Transformer layers of ESM-2. This approach maintains the model's capacity to adapt to TCR-epitope binding without a substantial hit to resources. The hyperparameters K, α, and r can be tuned to optimize performance for our specific intentions.

8.3 Adapter Tuning

Adapter tuning provides another efficient approach to fine-tuning ESM-2 by inserting specialized adapter modules into the top K Transformer layers [11]. Each adapter employs a bottleneck architecture with skip connections to compress and reconstruct input features, all using fewer parameters. The overall transformation for each layer with an adapter is described by:

$$h' = h + \sum_{i=1}^{N} a_i(h)$$

where h represents the input features for a Transformer layer, and the adapter transformation $a(h)$ is defined as:

$$a(h) = W_{\text{up}} \cdot \sigma(W_{\text{down}} \cdot h + b_{\text{down}}) + b_{\text{up}}$$

Here, W_{up} and W_{down} are trainable matrices for up-projection respectively, while b_{up} and b_{down} are the corresponding biases. The ReLU activation function σ is applied to the down-projection. This bottleneck structure effectively reduces the number of trainable parameters while maintaining the models ability to adapt to TCR-epitope binding prediction tasks.

The number of transformer layers K where adapters are inserted serves as a hyperparameter. This method enables efficient fine-tuning for TCR-epitope binding prediction while keeping the overall number of trainable parameters relatively low, making it particularly suitable for our computational constraints.

8.4 Inference Strategy

In the original CLEAN model, the inference methods—max-sep and p-value scoring—were designed for datasets where a single enzyme is associated with all its known functionalities, leveraging the dataset's completeness. However, for our TCR dataset, predicting binding specificity is inherently challenging since it is nearly impossible to comprehensively enumerate all epitopes that a TCR can bind to. This fundamental difference makes CLEAN's inference strategies less suitable for our task and explains the poor performance of our previous adaptations.

To address this, we are exploring alternative inference strategies, such as XGBoost, MLP, and CNN, which are better suited for predicting the binding specificity of each TCR-epitope pair, aligning more closely with the biological reality and the limitations of our dataset.

8.5 SMILES

Our initial goal was to integrate multimodal data, including 3D molecular structures, to improve model predictions. However, challenges in obtaining high-quality 3D data led us to adopt SMILES [19] representations as a practical alternative.

Using SMILES, we extracted key molecular features such as molecular weight, partition coefficient, polar surface area, hydrogen bond donors, and acceptors. These features, combined with sequence-derived embeddings, expanded the feature space, allowing the prediction model to leverage complementary information. This approach balances data availability with predictive performance, enabling more robust modeling outcomes.

9 Experiments and Results

Our framework integrates an encoder, projection head, and prediction models within a contrastive learning paradigm. The encoder and projection head generate embeddings that align similar TCR-epitope pairs in the feature space while maximizing the separation of dissimilar pairs, enabling the prediction models to classify binding interactions with high accuracy.

We expanded CLEAN's data batch handling method -C by introducing a standard, regular batch sampling method -R (CLEAN-style sampling uses a single batch per epoch, and the batch size is the number of classes.) To ensure fair comparisons, the total model updates were kept constant, though CLEAN-style required more epochs (34,400) than the regular approach (400).

To manage the computational demands of the ESM-2 encoder (650M parameters), we employed mixed-precision training to optimize memory usage. Training was performed on NVIDIA A100 80GB GPUs, with durations ranging from 2–7 days per model. We present only the most informative results here. The complete experiment results are available at **https://github.com/Yichuan0712/11785-TCR/blob/main/final_report/results.csv**.

9.1 Baselines

Table 5 summarizes baseline models, primarily our reimplementations of CLEAN variations using the 650M ESM-2 as the base model. The VIBTCR model, mentioned earlier, performs significantly worse than methods utilizing the protein language model encoder. Among predictors, XGBoost shows inferior performance compared to CNN and MLP, which deliver similar results across most metrics. The batch sampling strategy (CLEAN-style or regular) has minimal impact on overall performance.

Table 5: Baseline Models

Encoder	Predictor	Accuracy	AUROC	AUPR	F1
N/A	VIBTCR	0.6350	0.6927	0.6702	0.6525
650M-R	XGB	0.6670	0.7287	0.7219	0.6325
650M-R	MLP	0.6997	0.7588	0.7404	0.7132
650M-R	CNN	**0.7022**	**0.7636**	0.7427	0.6922
650M-C	XGB	0.6774	0.7443	0.7352	**0.6952**
650M-C	MLP	0.6984	0.7606	**0.7464**	0.6921
650M-C	CNN	0.6956	0.7493	0.7280	0.7072

9.2 Top Layer Fine-Tuning

Table 6 presents the results of top-layer fine-tuning with various predictors, batch strategies, and numbers of fine-tuned layers. Fine-tuning fewer layers (e.g., top 2 or 4) consistently delivers better performance, with MLP and CNN achieving competitive results. CLEAN-style batches slightly outperform regular batches, but the differences are minimal. The best configurations are 650M-C-top2 with MLP for accuracy and 650M-C-top4 with CNN for F1, demonstrating the effectiveness of shallow fine-tuning and CLEAN-style batching.

Table 6: Top Layer Fine-Tuning

Encoder	Predictor	Accuracy	AUROC	AUPR	F1
650M-R-top2	MLP	0.7471	0.8124	0.8006	0.7559
650M-R-top4	MLP	0.7223	0.7833	0.7716	0.7316
650M-R-top6	MLP	0.6880	0.7388	0.7041	0.6928
650M-R-top2	CNN	0.7427	0.8177	**0.8083**	0.7517
650M-R-top4	CNN	0.7268	0.7940	0.7796	0.7279
650M-R-top6	CNN	0.6897	0.7443	0.7113	0.6785
650M-C-top2	MLP	**0.7526**	0.8145	0.7952	0.7570
650M-C-top4	MLP	0.7516	0.8205	0.7975	0.7565
650M-C-top6	MLP	0.6867	0.7459	0.7141	0.6894
650M-C-top2	CNN	0.7503	**0.8220**	0.8062	0.7547
650M-C-top4	CNN	**0.7526**	0.8216	0.8004	**0.7660**
650M-C-top6	CNN	0.6929	0.7438	0.7080	0.7034

9.3 Adapter Tuning

Table 7 presents the results of adapter tuning with varying adapter dimensions (12, 16, 20), predictors, and batch strategies. Overall, CLEAN-style batches consistently outperform regular batches across most metrics. Larger adapter dimensions (e.g., 20) generally lead to better results, with the 650M-C-adp20 configuration and CNN predictor achieving the highest AUROC and AUPR. MLP and CNN predictors perform similarly. These results highlight the importance of both adapter dimension selection and batch strategy in optimizing performance.

Table 7: Adapter Tuning

Encoder	Predictor	Accuracy	AUROC	AUPR	F1
650M-R-adp12	MLP	0.7126	0.7696	0.7481	0.7284
650M-R-adp16	MLP	0.7011	0.7622	0.7394	0.7010
650M-R-adp20	MLP	0.6867	0.7416	0.7074	0.7066
650M-R-adp12	CNN	0.6944	0.7539	0.7284	0.6876
650M-R-adp16	CNN	0.7014	0.7685	0.7561	0.7054
650M-R-adp20	CNN	0.6910	0.7429	0.7080	0.6887
650M-C-adp12	MLP	0.7130	0.7757	0.7566	0.7255
650M-C-adp16	MLP	0.7238	0.7857	**0.7656**	0.7317
650M-C-adp20	MLP	0.7238	0.7844	0.7577	**0.7348**
650M-C-adp12	CNN	0.7143	0.7846	0.7631	0.7207
650M-C-adp16	CNN	0.7170	0.7807	0.7602	0.7014
650M-C-adp20	CNN	**0.7245**	**0.7869**	0.7630	0.7269

9.4 LoRA

Table 8 presents the performance of LoRA-based tuning with different configurations specifying the number of layers where LoRA is applied (e.g., 16 and 33), predictors, and batch strategies. Models with more layers incorporating LoRA (e.g., 33) consistently demonstrate better performance across most metrics. The 650M-R-lora33 configuration with CNN achieves the highest AUROC and Accuracy, while MLP excels in AUPR and F1. Regular batches generally outperform CLEAN batches slightly, though the differences remain minimal. These results highlight the significant impact of the number of LoRA-applied layers and predictor choice on model performance.

Table 8: LoRA

Encoder	Predictor	Accuracy	AUROC	AUPR	F1
650M-R-lora16	MLP	0.7293	0.8016	0.7848	0.7313
650M-R-lora33	MLP	0.7444	0.8181	**0.8025**	**0.7561**
650M-R-lora16	CNN	0.7321	0.8065	0.7959	0.7348
650M-R-lora33	CNN	**0.7484**	**0.8203**	0.8013	0.7438
650M-C-lora16	MLP	0.7151	0.7799	0.7575	0.7280
650M-C-lora33	MLP	0.7391	0.8116	0.7934	0.7479
650M-C-lora16	CNN	0.7179	0.7839	0.7612	0.7229
650M-C-lora33	CNN	0.7331	0.8089	0.7914	0.7310

9.5 SMILES

In this section, we focus on whether the features extracted from SMILES representations can further enhance the performance of the previously best-performing fine-tuning configurations. From the SMILES representations, we extracted a set of chemically relevant features, including molecular weight, partition coefficient, topological polar surface area, the number of hydrogen bond donors, and the number of hydrogen bond acceptors. These features were then integrated into the models to evaluate their impact on the predictive performance.

Table 9 shows that incorporating SMILES-derived chemical features enhances model performance across most metrics. Notably, models augmented with these features achieve higher AUROC, AUPR, and F1 scores, demonstrating that these chemically relevant descriptors provide complementary information to the encoder embeddings, improving prediction performance.

Table 9: SMILES

Encoder	Predictor	Accuracy	AUROC	AUPR	F1
650M-C-top4	CNN	0.7526	0.8216	0.8004	**0.7660**
650M-C-adp20	CNN	0.7245	0.7869	0.7630	0.7269
650M-R-lora33	MLP	0.7444	0.8181	0.8025	0.7561
650M-R-lora33	CNN	0.7484	0.8203	0.8013	0.7438
650M-C-top4	smi-CNN	**0.7548**	0.8265	0.8055	0.7556
650M-C-adp20	smi-CNN	0.7213	0.7898	0.7688	0.7193
650M-R-lora33	smi-MLP	0.7431	0.8156	0.8002	0.7467
650M-R-lora33	smi-CNN	0.7516	**0.8272**	**0.8119**	0.7605

9.6 Latent Space Distributions of TCR Sequence Embedding

To evaluate the performance of our models in embedding TCR sequences associated with specific epitopes, we selected four epitope types and visualized the latent space distributions of their corresponding embeddings.

Figure 2 presents the PCA projections of these embeddings, comparing the pre-trained ESM2 model with two best fine-tuned versions, 650M-C-top4 and 650M-R-lora33. The first model, fine-tuned on clean batch data with updates restricted to the top 4 layers, shows the most distinct clustering, indicating its ability to effectively separate embeddings by epitope classes. In contrast, the original pre-trained ESM2 model demonstrates significant overlap between classes, reflecting its lack of task-specific adaptation. The second fine-tuned model, utilizing LoRA across 33 layers and trained on regular batch data, retains some class separation but shows noisier clustering. This visualization highlights the importance of fine-tuning strategies in optimizing ESM2 embeddings for epitope-specific tasks.

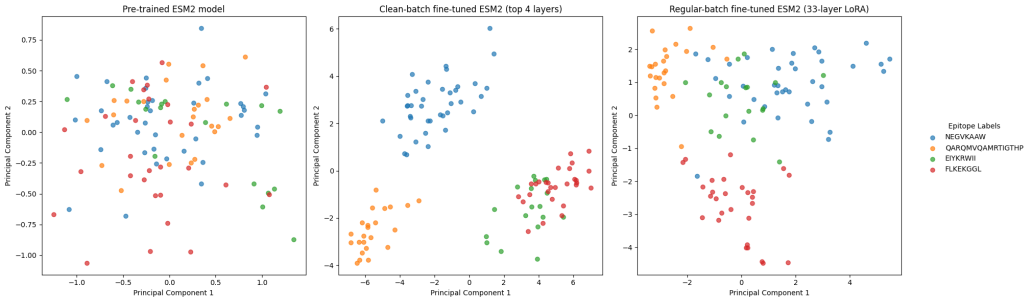

Figure 2: PCA Visualization of Latent Space Distributions of TCR Sequence Embeddings from Pre-trained and Fine-tuned ESM2 Models

9.7 SupCon Loss

The SupCon loss improves upon Triplet Margin loss by leveraging more positive and negative samples per batch, enhancing embedding separation. We adapted our previously best-performing configuration (regular batch with 33-layer LoRA) to use SupCon loss, achieving significant improvements and obtaining the best scores **across all metrics**. However, this enhancement came with substantial computational demands, extending training times to over a week per model. This limitation constrained our exploration of SupCon loss, emphasizing the need for further optimization.

Table 10: SupCon Loss

Encoder	**Predictor**	**Accuracy**	**AUROC**	**AUPR**	**F1**
650M-R-lora33	smi-CNN	0.7516	0.8272	0.8119	0.7605
650M-R-lora33-supcon	smi-CNN	**0.8052**	**0.8825**	**0.8665**	**0.8171**

10 Discussion

The results across the experiments demonstrate the efficacy of our approach in leveraging various strategies for fine-tuning and feature integration.

Our experiments show that both CLEAN-style and regular batch sampling strategies have comparable performance in most setups. CLEAN-style sampling, while effective for handling class-imbalanced datasets due to its class-based batch construction, may overlook some data. In contrast, regular batching offers greater flexibility for general applications and ensures that all data is included. The unified batch size and equalizing update counts ensured fair comparisons between these approaches.

Fine-tuning fewer top layers consistently yielded some of the best results across multiple metrics, with CLEAN-style batches slightly outperforming regular batches. This suggests that shallow fine-tuning helps maintain the generalization capabilities of the pre-trained encoder while effectively adapting to the downstream task.

Larger adapter dimensions demonstrated superior performance across metrics, indicating that additional parameters enable the model to better adapt to task-specific nuances. CLEAN-style batches proved advantageous in these setups, particularly when paired with CNN predictors, achieving the highest AUROC and AUPR scores.

Increasing the number of layers utilizing LoRA significantly improved performance, with higher configurations delivering better results across all metrics. LoRA demonstrates the potential to reduce computational overhead while maintaining high performance, especially in resource-constrained scenarios.

Incorporating chemically relevant features derived from SMILES representations further enhanced performance, particularly for AUROC, AUPR, and F1 scores, especially when combined with LoRA. This highlights the complementary nature of these features to the encoder embeddings, enabling the model to capture additional molecular-level information for improved predictions.

11 Future Works

One key area for future work is optimizing the training process to improve efficiency, particularly for multi-GPU setups. Currently, the model struggles to handle larger batch sizes due to memory constraints, and the SupCon loss exacerbates this issue by significantly increasing the data volume. Addressing these limitations through better parallelization and memory management will be critical to enabling faster and more scalable training.

Another important direction is to better leverage epitope sequence information. Currently, our approach fully utilizes TCR information, while epitope sequences are primarily treated as class labels rather than as meaningful features. A more ideal solution would involve modifying the loss function to directly incorporate epitope sequence embeddings, enabling the model to effectively combine both TCR and epitope information. This integration could provide a more comprehensive representation of the interaction and lead to improved predictive performance.

Additionally, our current model is only capable of making predictions for unseen TCRs, while it cannot handle unseen epitopes. Methods like VIBTCR, though less effective in terms of performance, are capable of addressing unseen epitopes. Integrating such capabilities through model ensemble approaches could provide a path forward to improve the generalization of our framework and enhance its applicability.

Finally, additional experiments are needed to thoroughly test the SupCon loss. While it has shown promise, our current evaluation is not comprehensive. Conducting more trials will help us better understand its impact and refine its implementation for improved performance.

12 Conclusion

This study presents a comprehensive exploration of leveraging fine-tuned protein language models and contrastive learning for TCR-epitope binding prediction. By integrating state-of-the-art techniques such as ESM-2 embeddings, adapter layers, LoRA, and SMILES-derived chemical features, we developed a flexible and extensible framework that outperforms traditional methods like VIBTCR. Our results demonstrate the effectiveness of shallow fine-tuning strategies, the utility of CLEAN-style batch sampling for imbalanced datasets, and the complementary benefits of incorporating chemically relevant descriptors into model training.

Key findings reveal that shallow fine-tuning preserves the generalization capabilities of pre-trained encoders while effectively adapting to the TCR-epitope binding task. The introduction of LoRA and adapter layers provided parameter-efficient fine-tuning mechanisms, significantly enhancing performance across key metrics, such as AUROC and AUPR. The addition of SMILES-derived features further enriched the feature space, enabling the model to capture nuanced molecular interactions, particularly when paired with LoRA.

Despite these advancements, several challenges remain. The computational demands of large-scale training, particularly with SupCon loss, highlight the need for optimized multi-GPU training strategies. Additionally, the current framework's inability to generalize to unseen TCRs, in contrast to methods like VIBTCR, indicates a gap that could be addressed through model ensembling or enhanced loss functions. Future work should focus on improving the integration of TCR sequence embeddings into the loss function and conducting further experiments to fully evaluate the potential of SupCon loss.

Overall, this work contributes a robust and scalable framework for TCR-epitope binding prediction, demonstrating the potential of combining protein language models with contrastive learning. By advancing the predictive accuracy and generalization capabilities of these models, this study lays the groundwork for future innovations in immunotherapy, vaccine development, and precision medicine.

13 Division of Work

Yichuan designed the project architecture, built the prototype, conducted experiments, and highlighted key points for the final report. Shubham improved the code, modified the original CLEAN implementation, ran experiments, and contributed to the literature review section of the report. Isaac provided the abstract and introduction, compiled sources and precedent projects, and assisted with fleshing out the final report, formulas, and a colab friendly CLEAN implementation. Adam assisted in performance analysis and created performance graph and table.

Our project source code can be found here: **https://github.com/Yichuan0712/11785-TCR**. Although we reimplemented the CLEAN baseline in our repository, we also developed modified versions based on the original code. They are available at: **https://github.com/s-kachroo/CMU-11785-TCR-CLEAN**.

References

[1] Jamie Rossjohn, Staphanie Gras, John J Miles, Stephen J Turner, Dale I Godfrey, and James McCluskey. T-cell antigen receptor recognition of antigen-presenting molecules. *Annual Review of Immunology*, 33, 12 2014.

[2] A. Duncan F. Hooshmand C.D. Katayama, F.J. Eidelman and S.M. Hedrick. Predicted complementarity determining regions of the t-cell antigen receptor determine antigen specificity. *The EMBO Journal*, 14, 3 1995.

[3] T Dong K Harlos K Di Gleria L Dorrell DC Douek PA van der Merwe EY Jones JK Lee, G Stewart-Jones and AJ McMichael. T cell cross-reactivity and conformational changes during tcr engagement. *Journal of Experimental Medicine*, 11:1455–1466, 12 2004.

[4] Ton N Schumacher Robert D Schreiber. Neoantigens in cancer immunotherapy. *Science*, 4 2015.

[5] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[6] William Chronister, Austin Crinklaw, Swapnil Mahajan, Randi Vita, Zeynep Kosaloglu, Zhen Yan, Jason Greenbaum, Leon Jessen, Morten Nielsen, Scott Christley, Lindsay Cowell, Alessandro Sette, and Bjoern Peters. Tcrmatch: Predicting t-cell receptor specificity based on sequence similarity to previously characterized receptors. *Frontiers in Immunology*, 12:640725, 03 2021.

[7] Vanessa Isabell Jurtz, Leon Eyrich Jessen, Amalie Kai Bentzen, Martin Closter Jespersen, Swapnil Mahajan, Randi Vita, Kamilla Kjærgaard Jensen, Paolo Marcatili, Sine Reker Hadrup, Bjoern Peters, and Morten Nielsen. Nettcr: sequence-based prediction of tcr binding to peptide-mhc complexes using convolutional neural networks. *bioRxiv*, 2018.

[8] Tristan Bepler and Bonnie Berger. Learning the protein language: Evolution, structure, and function. *Cell systems*, 12(6):654–669, 2021.

[9] Duolin Wang, Usman L Abbas, Qing Shao, Jin Chen, and Dong Xu. S-plm: Structure-aware protein language model via contrastive learning between sequence and structure. *bioRxiv*, 2023.

[10] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[11] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[13] Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. *Science*, 379(6639):1358–1363, 2023.

[14] Yiming Fang, Xuejun Liu, and Hui Liu. Attention-aware contrastive learning for predicting T cell receptor–antigen binding specificity. *Briefings in Bioinformatics*, 23(6):bbac378, 09 2022.

[15] Margarita Pertseva, Oceane Follonier, Daniele Scarcella, and Sai T Reddy. TCR clustering by contrastive learning on antigen specificity. *Briefings in Bioinformatics*, 25(5):bbae375, 08 2024.

[16] Yuta Nagano, Andrew Pyo, Martina Milighetti, James Henderson, John Shawe-Taylor, Benny Chain, and Andreas Tiffeau-Mayer. Contrastive learning of t cell receptor representations, 2024.

[17] Anna Weber, Jannis Born, and María Rodríguez-Martínez. Titan: T-cell receptor specificity prediction with bimodal attention networks. *Bioinformatics*, 37(23):3221–3244, 10 2021.

[18] Dmitry Bagayev, Evgeny Egorov, Andrey Bagaev, Natalia Vostrikov, Valtteri Scholz, Ilya Zvyagin, Benjamin Chain, Viktor Egorov, and Olga Britanova. Vdjdb in 2019: database extension, new analysis infrastructure, and a t-cell receptor motif compendium. *Nucleic Acids Research*, 48(D1):D1057–D1062, 01 2020.

[19] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[20] Jennifer N. Dines, Kyle P. Jensen, Nanna P. Jørgensen, Ramin S. Herati, Angela C. Huang, Georgia Oliveira, Carl M. Goedhart, Jason C. Chase, Martha A. Alexander-Miller, and Stephen P. Schoenberger. The immunodominant sars-cov-2 t-cell epitope hla-a*02:01 glctlvaml: evidence for cd8+ t-cell activation in covid-19 patients. *medRxiv*, 07 2020.

[21] Giancarlo Croce, Sara Bobisse, Dana Léa Moreno, Julien Schmidt, Philippe Guillame, Alexandre Harari, and David Gfeller. Deep learning predictions of tcr-epitope interactions reveal epitope-specific chains in dual alpha t cells. *Nature Communications*, 15(3211), 09 2024.

[22] Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning (supplementary materials). *Science*, 379(6639):2, 2023.

[23] Filippo Grazioli, Pierre Machart, Anja Mösch, Kai Li, Leonardo V. Castorina, Nico Pfeifer, and Martin Renqiang Min. Attentive variational information bottleneck for tcr–peptide interaction prediction. *Bioinformatics*, 39(1):btac820, 01 2023.