GraphTCR-Mamba: heterogeneous graph learning with Mamba for TCR-epitope binding prediction

Ellison Naz[1], Adam Fan[2], Wenjun Lin[1], Ping Luo[1*]

[1]*Department of Computer Science & Mathematics, Algoma University, Brampton, Ontario L6V 1A3, Canada*

[2]*Bayview Secondary School, Richmond Hill, Ontario, L4C 2L4, Canada*

T-cell receptor (TCR) recognition of peptide epitopes is central to adaptive immunity, yet only a small fraction of potential TCR-epitope interactions can be characterized experimentally. Existing computational approaches often model candidate pairs independently, underutilizing relational information among similar sequences. We present GraphTCR-Mamba, a heterogeneous graph framework that represents TCRs and epitopes as distinct node types, integrates sequence-similarity and observed-interaction edges, and incorporates a Mamba selective state-space branch for global information propagation. Across the IEDB, McPAS-TCR, and VDJdb benchmarks, GraphTCR-Mamba achieved AUROCs of 0.938, 0.943, and 0.993 and AUPRs of 0.925, 0.947, and 0.992, respectively, outperforming the evaluated baselines. We further assessed cold-start generalization by withholding interaction labels involving designated TCRs, epitopes, or both while retaining sequence-derived graph structure. Generalization was stronger for unseen TCRs than for unseen epitopes, while prediction when both partners were unseen remained challenging. Ablation analysis further showed that graph-dependent components were particularly beneficial for unseen-epitope prediction. Overall, GraphTCR-Mamba provides an effective framework for integrating sequence and relational information for TCR-epitope prediction, while the observed cold-start performance gap highlights the importance of evaluating generalization beyond random pair-level splits.

Keywords: T cell receptor; Heterogeneous graph; TCR-epitope prediction; Mamba; Laplacian positional encoding.

[*] Corresponding author: ping.luo@algomau.ca

© 2026 The Authors. Open Access chapter published by World Scientific Publishing Company and distributed under the terms of the Creative Commons Attribution Non-Commercial (CC BY-NC) 4.0 License.

1. Introduction

T-cell receptors (TCRs) play a central role in adaptive immunity by recognizing peptide epitopes presented on the surface of infected, malignant, or otherwise abnormal cells.[1] Determining TCR-epitope specificity experimentally, however, is costly and time-consuming, and only a small fraction of the vast space of potential TCR-epitope interactions can be characterized. Computational methods offer a scalable approach for learning from known interactions and prioritizing candidate TCR–epitope pairs for experimental investigation.[2]

Recent TCR-epitope predictors address this problem using diverse representations and generalization strategies. TCR-ESM leverages protein language-model representations[3], and PanPep targets prediction with few or previously unseen peptide examples.[4] MixTCRpred develops epitope-specific predictors[5], whereas UniPMT models peptide-MHC-TCR relationships using a heterogeneous graph.[6] Collectively, these studies highlight the importance of both sequence representation and generalization beyond well-represented interaction partners. Recent benchmarking studies further demonstrate that evaluation choices, including negative-sample construction and whether epitopes are seen during training, can substantially affect reported performance.[7]

Despite this progress, important challenges remain. Many methods evaluate TCR-epitope pairs independently, overlooking relationships among sequences that may provide useful context for prediction. In particular, TCRs with similar sequence or structural properties may share binding preferences, while related epitopes may exhibit similar recognition patterns. Incorporating such relationships could therefore improve prediction, particularly when generalizing beyond interactions observed during training. Methods that explicitly model relationships among sequences, however, often rely on attention-based architectures whose computational cost increases substantially with sequence or dataset size. These limitations are especially relevant under cold-start settings, where the model must predict interactions involving TCRs, epitopes, or both that were not observed during training.

To address these challenges, we propose GraphTCR-Mamba, a framework that combines heterogeneous graph message passing with a Mamba-based selective state-space model for TCR-epitope binding prediction. Rather than treating each candidate pair independently, GraphTCR-Mamba represents TCRs and epitopes within a heterogeneous graph and propagates information among related sequences, allowing predictions to incorporate both sequence-level features and relational context. The Mamba component provides linear-complexity global node mixing after graph construction, enabling efficient propagation of long-range information across the graph.

We evaluate GraphTCR-Mamba on three benchmark datasets, IEDB[8], VDJdb[9], and McPAS-TCR[10], which differ in scale, curation strategy, and biological context. In addition to standard evaluation, we assess performance under cold-split settings designed to test generalization to previously unseen TCRs and epitopes. Across these datasets, GraphTCR-Mamba achieves strong and consistent predictive performance and retains meaningful predictive ability under the more challenging cold-split evaluation.

Our work makes three main contributions. First, we introduce a heterogeneous graph-based framework that integrates relational information with efficient sequence modeling for TCR-epitope binding prediction. Second, we demonstrate its performance across three benchmark datasets spanning nearly two orders of magnitude in scale and diverse curation settings. Third, we systematically evaluate generalization under cold-split settings, providing a more stringent assessment of the model's ability to predict interactions involving previously unseen TCRs and epitopes.

2. Methods

2.1. *Datasets*

Three publicly available TCR-epitope datasets were used to evaluate GraphTCR-Mamba: IEDB, VDJdb, and McPAS-TCR.[8–10] These resources differ in their curation criteria, annotation scope, and dataset composition and are not assumed to be mutually independent or exhaustive. Some TCR sequences, epitopes, or reported associations may overlap across databases. We therefore processed and evaluated each dataset separately rather than combining them.

IEDB contains experimentally validated TCR-epitope interactions supported by direct evidence of binding. Its stringent curation resulted in a smaller, high-confidence dataset comprising 8,187 TCR-epitope pairs, 7,682 unique TCRs, and 980 unique epitopes after filtering. VDJdb also contains experimentally supported TCR-epitope associations but aggregates records from a broader range of studies, yielding 71,686 pairs, 69,337 unique TCRs, and 1,352 unique epitopes after filtering. In contrast, McPAS-TCR includes TCR sequences associated with disease-related antigens and pathological contexts, such as cancer, autoimmune disease, and infection. Thus, some McPAS-TCR associations may represent antigen or disease context rather than directly validated TCR-epitope binding. After filtering, McPAS-TCR contained 996 TCR-epitope pairs, 941 unique TCRs, and 32 unique epitopes. All three resources originally contained both human and mouse TCR sequences. Consistent with the human-focused scope of this study, only human entries were retained for subsequent analyses.

2.2. *Model Architecture*

GraphTCR-Mamba formulates TCR-epitope association prediction as a link-prediction task on a heterogeneous graph comprising TCR and epitope nodes. Fig. 1 shows the architecture of the model. The framework uses domain-specific Graph-Mamba encoders that integrate edge-aware local message passing with global selective state-space modeling. TCR and epitope similarity graphs are encoded separately, while observed training associations enable information exchange between the two domains. The resulting node representations are then used by an edge classifier to predict candidate TCR-epitope interactions. This architecture extends the graph-prioritized node serialization strategy of Graph-Mamba[11] to heterogeneous TCR-epitope graphs while retaining the efficient selective state-space modeling of Mamba.[12]

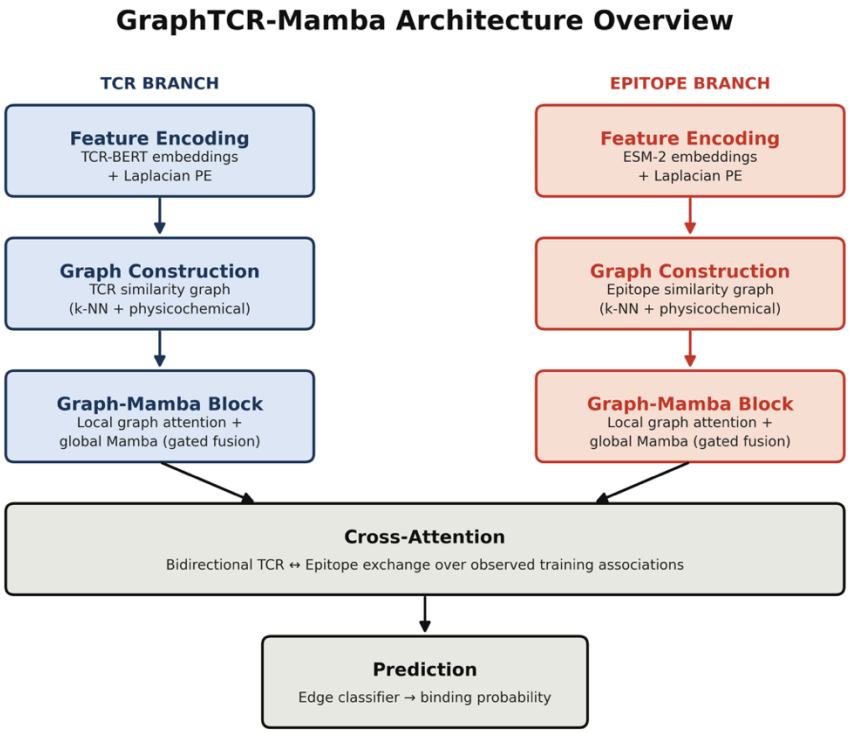


Fig. 1. Overview of the GraphTCR-Mamba architecture. TCR and epitope sequences are processed through parallel, domain-specific branches: pretrained language model embeddings (BertTCR and ESM-2) augmented with Laplacian positional encodings, followed by similarity-graph construction and stacked Graph-Mamba blocks combining local graph attention with global Mamba mixing via gated fusion. The two branches then exchange information through bidirectional cross-attention over observed training associations before a final edge classifier scores each candidate pair.

2.2.1. *Node features and graph structure*

For each TCR with paired CDR3α and CDR3β sequences, 768-dimensional mean-pooled BertTCR[13] embeddings of the CDR3α and CDR3β sequences were concatenated to obtain the TCR representation. Epitopes were represented by 1,280-dimensional mean-pooled embeddings from the ESM-2 model.[14] Type-specific linear layers, followed by layer normalization and ReLU activation, projected the TCR and epitope representations into a shared hidden space.

Two undirected intra-type graphs were constructed from precomputed sequence similarities. Each TCR was connected to its k_T nearest TCR neighbors, and each epitope was connected to up to k_e nearest epitope neighbors, with the pairwise similarity w_{ij} retained as an edge feature. Similarity was defined as an equally weighted combination of standardized k-mer cosine similarity and AAindex-

derived physicochemical cosine similarity.[15] Known TCR-epitope binding pairs formed the cross-type edge set. Within each cross-validation fold, only positive associations from the training partition were included in message passing, ensuring that held-out interactions were not exposed to the graph encoder.

To encode position within each intra-type graph, the model additionally used the eight lowest non-trivial eigenvectors of the unweighted normalized graph Laplacian, a standard spectral positional encoding for graph networks.[16] For adjacency matrix A and degree matrix D,

$$L = I - D^{-1/2}\,\mathrm{A}\,D^{-1/2}, \quad p_i = [u_2(\mathrm{i}), \ldots, u_9(\mathrm{i})] \tag{1}$$

where u_e is the eigenvector associated with the eth-smallest eigenvalue of L. The positional vector was layer-normalized, linearly projected, and added to the type-specific feature projection. Thus, for node i of type $r \in \{TCR, Epitope\}$:

$$x_i^{(0)} = \mathrm{ReLU}(\mathrm{LN}(W_r\, e_i + b_r)) + W_p\,\mathrm{LN}(p_i) \tag{2}$$

2.2.2. *Intra-domain Graph-Mamba encoder*

Each node type was processed by its own stack of two Graph-Mamba blocks. A block first applies two-head, edge-aware graph attention over the sparse similarity graph. For destination node i, source neighbor j, and attention head m, the unnormalized score and normalized attention coefficient are:

$$s_{ij}^m = \frac{((W_Q^m\, x_i)^T (W_K^m\, x_j))}{\sqrt{d_h}} + (W_e\, w_{ij})_m, \quad \alpha_{ij}^m = \frac{\exp(s_{ij}^m)}{\sum_{\mathrm{k}\in\mathrm{N(i)}} \exp(s_{ik}^m)} \tag{3}$$

where dh = d/2 = 128. Values from neighboring nodes are aggregated separately for each head, concatenated, linearly transformed, and combined with a residual connection:

$$h_i = \mathrm{LN}(x_i + \mathrm{Dropout}(W_O\ \mathrm{Concat}_m[\textstyle\sum_{\mathrm{j}\in\mathrm{N(i)}} \alpha_{ij}^m\, W_V^m\, x_j])) \tag{4}$$

The local representation H = [h₁, …, hₙ] is then serialized for global Mamba processing. Because a graph has no canonical node order, a graph-prioritized permutation is obtained by sorting nodes by degree perturbed by independent uniform noise, following the Hybrid-Degree-Noise strategy of Graph-Mamba.[11] If P_π denotes this permutation, the global output is:

$$\pi = \mathrm{argsort}(\deg(\mathrm{v}) + \varepsilon_v),\ \varepsilon_v \sim \mathrm{U}(0,1); \quad \mathrm{G} = P_\pi^{-1}\,\mathrm{Mamba}(P_\pi\,\mathrm{H}) \tag{5}$$

Mamba applies an input-dependent selective state-space recurrence along the permuted node sequence. In its discretized form, the recurrent state and output at position t can be summarized as:

$$z_t = \bar{A}_t\, z_{\mathrm{t}-1} + \bar{B}_t\, h_{\pi(\mathrm{t})}, \quad g_{\pi(\mathrm{t})} = C_t\, z_t + \mathrm{D}\, h_{\pi(\mathrm{t})} \tag{6}$$

where the discretized state-space parameters depend on the current input, allowing selective propagation or suppression of context.7 In the implementation, the selective state-space dimension is 16 (controlling how much per-position context the selective scan retains), the local convolution width is 4 (capturing short-range dependencies before the global scan), and the Mamba expansion factor is 2 (the internal channel multiplier applied within each block). One noisy permutation is sampled on each

training forward pass. At evaluation, outputs from five independently perturbed degree-based permutations are averaged after mapping them back to the original node order, reducing sensitivity to an arbitrary serialization.

A learned sigmoid gate controls how much global state-space context is injected into each locally aggregated node representation:

$$\gamma_i = \sigma(W_g[h_i \parallel g_i] + b_g), \quad \tilde{h}_i = \mathrm{LN}(h_i + \mathrm{Dropout}(q_i \odot g_i)) \tag{7}$$

The block concludes with a position-wise feed-forward network of width 2d, ReLU, dropout, a residual connection, and layer normalization. The local branch costs O(|E|d) on the sparse similarity graph, and the Mamba scan costs O(|V|d) with respect to node count, avoiding the O(|V|²d) global self-attention term after the similarity graph has been constructed.

2.2.3. *Cross-domain exchange and link prediction*

After the two intra-domain stacks, a pair of sparse cross-attention blocks propagates information over known training associations. TCR nodes first query their associated epitope nodes, and epitope nodes then query the updated TCR representations over the reversed association edges. For query node i and its cross-domain neighbors $\mathcal{N}_\times(i)$, each attention head computes:

$$\beta_{ij}^{m} = \mathrm{softmax}_{\mathrm{j}\in\mathcal{N}_\times(\mathrm{i})}\left(\frac{((W_Q^{\times,\mathrm{m}}\, q_i)^T (W_K^{\times,\mathrm{m}}\, k_j))}{\sqrt{d_h}}\right), \quad c_i = W_O^{\times}\, \mathrm{Concat}_m[\textstyle\sum_j \beta_{ij}^{m}\, W_V^{\times,\mathrm{m}}\, k_j] \tag{8}$$

followed by residual layer normalization and a two-layer feed-forward network. This operation incorporates context from associations observed in the training graph. The held-out candidate edge itself is not added to cross-attention.

For a candidate pair (i, j), the final TCR and epitope embeddings are concatenated and passed through a two-layer multilayer perceptron with hidden width 256, ReLU, dropout, and a scalar output. The binding probability is:

$$\ell_{ij} = w_2{}^T\, \mathrm{ReLU}(W_1[h_i^{\mathrm{TCR}} \parallel h_j^{\mathrm{EPI}}] + b_1) + b_2, \quad \hat{p}_{ij} = \sigma(\ell_{ij}) \tag{9}$$

2.2.4. *Training configuration*

Experimentally observed TCR-epitope associations were treated as positive pairs. Because experimentally validated non-binding pairs were not available, unobserved TCR-epitope pairs (i.e., zero-valued entries in the association matrix) were uniformly sampled as negative examples. These sampled pairs therefore represent unknown interactions rather than confirmed non-binding pairs. For each random seed, we sampled an equal number of positive and negative pairs to obtain a balanced dataset, and the two sets were independently shuffled and partitioned into ten folds. During each fold, only positive associations in the training partition were added as TCR-epitope edges to the graph encoder for message passing; sampled negative pairs were not added to the graph. Both positive and

sampled negative pairs were subsequently provided to the link classifier as candidate TCR-epitope pairs for predicting association probabilities.

The model was optimized using binary cross-entropy with logits and Adam with a learning rate of 0.0005 and weight decay of 0.0001. Residual and classifier dropout rates were set to 0.3, and training was limited to 150 epochs. Validation was evaluated every 10 epochs, and early stopping was applied after seven consecutive validation evaluations without improvement. Experiments were repeated using five random seeds. Hyperparameters, including the hidden dimension, number of attention heads, number of Graph-Mamba layers, dropout rate, learning rate, and weight decay, were selected through preliminary experiments on IEDB and subsequently fixed across all datasets and cold-split protocols to maintain a consistent evaluation setting.

Experiments were conducted on clusters of the Digital Research Alliance of Canada using the mamba_ssm implementation.[12] The IEDB, VDJdb, and McPAS-TCR datasets are publicly available from their respective sources, and the implementation is available from GitHub at https://github.com/GenTensor/GraphTCR-Mamba.

2.3. *Evaluation Protocol*

Model performance was evaluated using stratified 10-fold cross-validation repeated across five random seeds to assess robustness to data partitioning. Performance was assessed using the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPR). AUROC measures discrimination between positive and negative pairs across classification thresholds, whereas AUPR emphasizes performance on the positive class and is particularly informative when evaluating imbalanced interaction data.

To assess generalization to unseen sequences, we evaluated three cold-split settings: TCR-cold, where test TCRs were absent from training; epitope-cold, where test epitopes were absent; and both-cold, where neither member of a test pair appeared in training. Unlike random pair-level splits, these settings prevent the corresponding sequence identities from being shared across training and test partitions. We also evaluated a sequence-only ablation that retained the pretrained embeddings, type-specific projections, and pair classifier while removing graph processing, positional encodings, Mamba, and cross-attention. All other evaluation settings were kept unchanged for a controlled comparison.

2.4. *Baseline Models*

Four published methods were included as baselines. NetTCR-2.0 is a sequence-based deep learning model that predicts TCR-peptide interactions from paired TCRα and TCRβ CDR3 sequences.[17] HeteroTCR models TCR-peptide interactions using a heterogeneous graph that integrates sequence-derived similarities and known associations through graph neural networks.[18] GRAPE combines protein language-model representations with graph-based learning to exploit relationships among TCRs and epitopes and address interaction-data sparsity.[19] GTE similarly formulates TCR-epitope prediction as a graph-learning problem, primarily using TCRβ-chain sequence information to capture interaction-

related structure.[20] These graph-based baselines provide particularly relevant comparisons with GraphTCR-Mamba, which separately models TCR-TCR and epitope-epitope similarity structure, enables cross-domain information exchange, and incorporates Mamba-based global node mixing. All baselines were evaluated using the same cross-validation partitions and negative-sampling protocol as GraphTCR-Mamba.

3. Results

3.1. *Performance across benchmark datasets*

Figs. 2-4 compare GraphTCR-Mamba with the baseline methods on the IEDB, VDJdb, and McPAS-TCR datasets. The largest improvement was observed on IEDB, where GraphTCR-Mamba achieved an AUROC of 0.938 and an AUPR of 0.925, exceeding the strongest baseline by 0.066 and 0.032, respectively. On McPAS-TCR, the improvements were more modest, with gains of 0.011 in AUROC and 0.012 in AUPR over the best-performing baseline. Performance on VDJdb approached the ceiling across several methods; nevertheless, GraphTCR-Mamba achieved the highest AUROC (0.993) and tied NetTCR-2.0 for the highest AUPR (0.992). Overall, GraphTCR-Mamba achieved consistently strong performance across all three datasets, with the greatest advantage observed on the more challenging IEDB dataset.

We additionally evaluated the computational cost of GraphTCR-Mamba across datasets (Table 1). Mean runtime per fold increased from 3.58 s for McPAS-TCR (973 nodes) to 81.78 s for VDJdb (70,689 nodes), while peak GPU memory allocation increased from 0.43 to 26.06 GiB. These results demonstrate the practical feasibility of the framework on datasets of varying graph sizes, while also highlighting the increased memory requirements for larger interaction graphs.

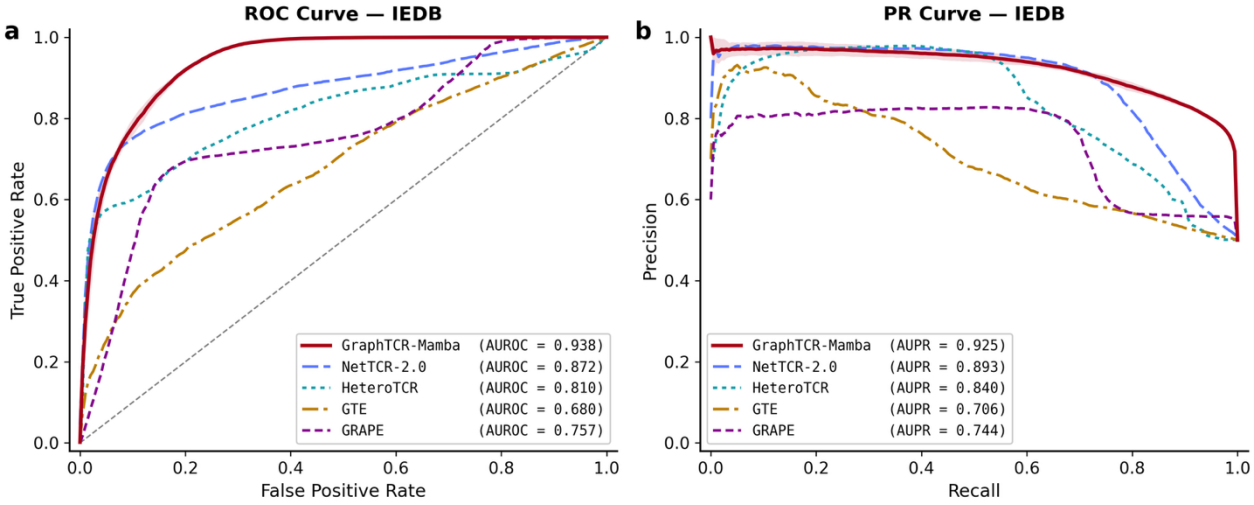


Fig. 2. Performance comparison on the IEDB dataset.

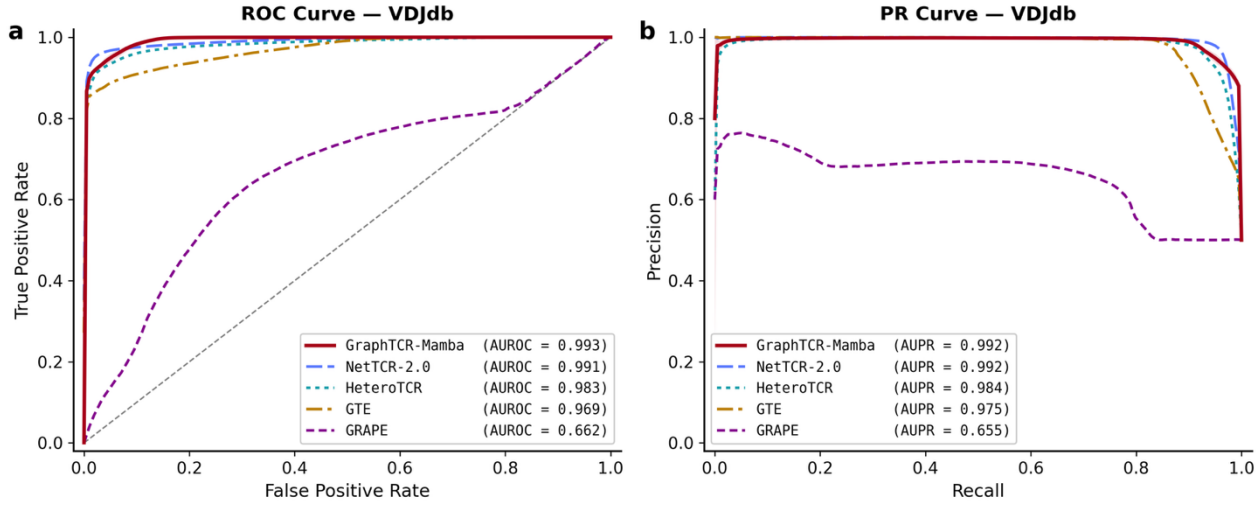


Fig. 3. Performance comparison on the VDJdb dataset.

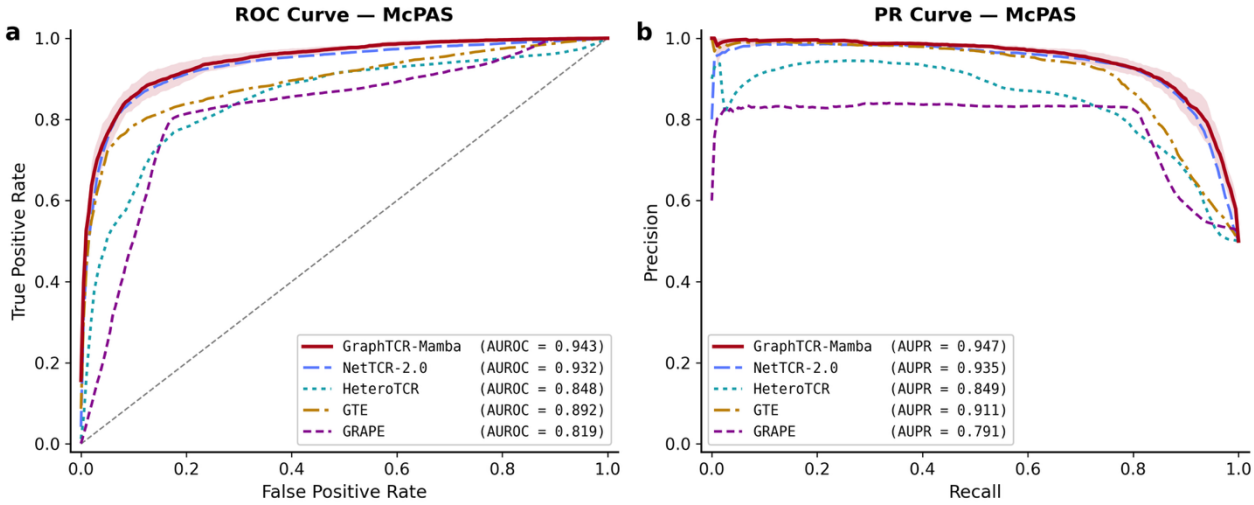


Fig. 4. Performance comparison on the McPAS-TCR dataset.

Table 1. Runtime and peak GPU memory usage of GraphTCR-Mamba. Runtime is reported as mean ± standard deviation per cross-validation fold, and memory as peak GPU allocation. All measurements were obtained using a single NVIDIA A100 GPU (40 GB) per job.

Dataset	Nodes	Time/fold (s)	Peak GPU memory (GiB)
IEDB	8,662	8.63 ± 0.21	3.27
McPAS-TCR	973	3.58 ± 0.33	0.43
VDJdb	70,689	81.78 ± 0.69	26.06

3.2. *Generalization under cold-start splits*

To assess whether GraphTCR-Mamba captures transferable interaction patterns beyond those observed during training, we evaluated the model under three cold-split settings for each dataset: TCR-cold, epitope-cold, and both-cold. Table 2 summarizes the results.

Across all three datasets, generalization depended strongly on which interaction partner was unseen. The model retained high discriminative performance under the TCR-cold protocol, suggesting that representations learned for observed epitopes transfer well to novel TCR sequences. Performance declined consistently when epitopes were held out, showing that generalization to new antigenic targets is substantially more difficult than generalization to new receptors.

Table 2. TCR-Epitope binding prediction performance under cold-split Protocols. Across all datasets, performance is higher for unseen TCRs than for unseen epitopes, while generalization when both partners are unseen is the most challenging.

Dataset	Protocol	Mean AUROC	Mean AUPR
IEDB	TCR-cold	0.871	0.895
IEDB	Epitope-cold	0.717	0.638
IEDB	Both-cold	0.503	0.533
McPAS-TCR	TCR-cold	0.920	0.901
McPAS-TCR	Epitope-cold	0.677	0.638
McPAS-TCR	Both-cold	0.568	0.629
VDJdb	TCR-cold	0.987	0.989
VDJdb	Epitope-cold	0.856	0.761
VDJdb	Both-cold	0.691	0.647

The both-cold protocol was the most challenging, with performance declined and showed greater fold-to-fold variability. This instability may partly reflect the smaller evaluation sets, particularly for McPAS-TCR, where three folds were excluded. Nevertheless, the consistent ordering across datasets suggests that the model relies partly on information associated with interaction partners observed during training. The model is therefore better suited to screening new TCRs against known epitopes than to predicting interactions in which both sequences are novel.

3.3. *Ablation study*

We evaluated LPE, the Mamba branch, and the sequence-only variant under the same cold-split protocols, using identical folds, negative samples, hyperparameters, and checkpoint selection (Table 3).

LPE removal had little effect under TCR-cold evaluation but consistently reduced performance when epitopes were unseen, with epitope-cold AUROC decreasing by 0.063–0.130 and AUPR by 0.072–0.100 across datasets. The contribution of Mamba was more dataset dependent: it improved unseen-epitope prediction on McPAS-TCR and VDJdb but provided little or no benefit in other settings. The sequence-only model performed similarly to the full model under TCR-cold evaluation but

substantially worse under epitope-cold evaluation, with AUROC reductions of 0.149–0.254 and AUPR reductions of 0.099–0.128. Both-cold performance was also generally lower. Overall, the ablations indicate that graph-dependent components are most beneficial for generalization to unseen epitopes, with LPE providing the most consistent contribution.

Table 3. Cold-split ablation analysis. Graph-dependent components provide the greatest benefit for unseen-epitope prediction, with LPE showing a consistent contribution across datasets, whereas the effect of the Mamba branch is dataset dependent. Values are mean AUROC / mean AUPR across folds.

Dataset	Protocol	Full model	Without Mamba	Without LPE	Sequence only
IEDB	TCR-cold	0.871 / 0.895	0.871 / 0.893	0.865 / 0.891	0.884 / 0.905
IEDB	Epitope-cold	0.717 / 0.638	0.735 / 0.663	0.622 / 0.558	0.463 / 0.510
IEDB	Both-cold	0.503 / 0.533	0.509 / 0.545	0.473 / 0.497	0.478 / 0.508
McPAS-TCR	TCR-cold	0.920 / 0.901	0.930 / 0.923	0.919 / 0.900	0.935 / 0.925
McPAS-TCR	Epitope-cold	0.677 / 0.638	0.634 / 0.608	0.547 / 0.538	0.528 / 0.538
McPAS-TCR	Both-cold	0.568 / 0.629	0.532 / 0.597	0.341 / 0.540	0.490 / 0.596
VDJdb	TCR-cold	0.987 / 0.989	0.987 / 0.989	0.984 / 0.987	0.988 / 0.990
VDJdb	Epitope-cold	0.856 / 0.761	0.798 / 0.736	0.793 / 0.689	0.688 / 0.637
VDJdb	Both-cold	0.691 / 0.647	0.676 / 0.640	0.635 / 0.584	0.643 / 0.593

4. Discussion

GraphTCR-Mamba achieved strong performance across the three datasets, while the cold-start experiments revealed important differences in generalization. The model generalized more effectively to unseen TCRs than to unseen epitopes, with both-cold prediction remaining the most challenging. The sequence-only ablation remained competitive under TCR-cold evaluation but performed substantially worse when epitopes were unseen, indicating that graph-derived information is particularly important for epitope generalization. The heterogeneous design preserves distinct TCR and epitope similarity structures while enabling cross-domain information exchange, and Mamba provides global node mixing without relying solely on local graph propagation. LPE provided the most consistent benefit for unseen epitopes, whereas the contribution of Mamba was dataset dependent. Thus, the architecture provides complementary sequence, graph, and global context for candidate prioritization, although the benefit of individual components varies across datasets and evaluation settings.

These findings have potential biological and translational relevance because TCR–epitope prediction can help prioritize candidate immune interactions from the large space of possible pairs for downstream investigation. Generalization to previously unseen receptors or epitopes is particularly relevant to such applications, where newly observed sequences may not be represented in existing training data. However, several limitations should be considered. Unobserved TCR–epitope pairs were sampled as negatives and therefore represent unknown rather than experimentally confirmed non-binding interactions. The framework also does not explicitly incorporate MHC information, and

sequence or interaction redundancy within and across public databases may affect apparent generalization. In addition, the cold-start evaluation can still leverage sequence-derived graph structure for unseen entities and therefore does not represent fully independent inductive prediction.

Future work should evaluate more stringent inductive settings and benchmark alternative methods under matched cold-split protocols, while incorporating biological context such as MHC information and assessing generalization across independently curated datasets. Pair-level analysis of interactions uniquely recovered or missed by GraphTCR-Mamba may further clarify the sequence and graph contexts in which the framework is most beneficial.

5. Conclusion

We introduced GraphTCR-Mamba, a heterogeneous graph framework that integrates sequence-derived relationships, relational message passing, and global node modeling for TCR-epitope prediction. The framework showed strong overall performance and encouraging generalization under cold-start settings, while prediction involving unseen epitopes and fully unseen pairs remains challenging. GraphTCR-Mamba can serve as a computational tool for prioritizing candidate TCR-epitope interactions. Further biological context, broader evaluation, and deeper analysis of individual predictions will be needed to establish its translational utility.

Acknowledgements

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference number RGPIN-2025-06236], and the Algoma University Research Fund. Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), [numéro de référence RGPIN-2025-06236].

References

1. Attaf, M., Legut, M., Cole, D. K. & Sewell, A. K. The T cell antigen receptor: the Swiss army knife of the immune system. *Clin. Exp. Immunol.* **181**, 1–18 (2015).
2. Springer, I., Besser, H., Tickotsky-Moskovitz, N., Dvorkin, S. & Louzoun, Y. Prediction of Specific TCR-Peptide Binding From Large Dictionaries of TCR-Peptide Pairs. *Front. Immunol.* **11**, (2020).
3. Yadav, S., Vora, D. S., Sundar, D. & Dhanjal, J. K. TCR-ESM: Employing protein language embeddings to predict TCR-peptide-MHC binding. *Comput. Struct. Biotechnol. J.* **23**, 165–173 (2023).
4. Gao, Y. *et al.* Pan-Peptide Meta Learning for T-cell receptor–antigen binding recognition. *Nat. Mach. Intell.* **5**, 236–249 (2023).
5. Croce, G. *et al.* Deep learning predictions of TCR-epitope interactions reveal epitope-specific chains in dual alpha T cells. *Nat. Commun.* **15**, 3211 (2024).

6. Zhao, Y. *et al.* A unified deep framework for peptide–major histocompatibility complex–T cell receptor binding prediction. *Nat. Mach. Intell.* **7**, 650–660 (2025).
7. Lu, Y. *et al.* Assessment of computational methods in predicting TCR–epitope binding recognition. *Nat. Methods* **23**, 248–259 (2026).
8. Vita, R. *et al.* The Immune Epitope Database (IEDB): 2024 update. *Nucleic Acids Res.* **53**, D436–D443 (2025).
9. Shugay, M. *et al.* VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Res.* **46**, D419–D427 (2018).
10. Tickotsky, N., Sagiv, T., Prilusky, J., Shifrut, E. & Friedman, N. McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences. *Bioinformatics* **33**, 2924–2929 (2017).
11. Wang, C., Tsepa, O., Ma, J. & Wang, B. Graph-Mamba: Towards Long-Range Graph Sequence Modeling with Selective State Spaces. *arXiv.org* https://arxiv.org/abs/2402.00789v1 (2024).
12. Gu, A. & Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. Preprint at https://doi.org/10.48550/arXiv.2312.00752 (2024).
13. Zhang, M. *et al.* BertTCR: a Bert-based deep learning framework for predicting cancer-related immune status based on T cell receptor repertoire. *Brief. Bioinform.* **25**, bbae420 (2024).
14. Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
15. Kawashima, S. & Kanehisa, M. AAindex: Amino Acid index database. *Nucleic Acids Res.* **28**, 374 (2000).
16. Maskey, S. *et al.* Generalized Laplacian Positional Encoding for Graph Representation Learning. Preprint at https://doi.org/10.48550/arXiv.2210.15956 (2022).
17. Montemurro, A. *et al.* NetTCR-2.0 enables accurate prediction of TCR-peptide binding by using paired TCRα and β sequence data. *Commun. Biol.* **4**, 1060 (2021).
18. Yu, Z., Jiang, M. & Lan, X. HeteroTCR: A heterogeneous graph neural network-based method for predicting peptide-TCR interaction. *Commun. Biol.* **7**, 684 (2024).
19. Fu, X. *et al.* GRAPE: graph-regularized protein language modeling unlocks TCR-epitope binding specificity. *Brief. Bioinform.* **26**, bbaf522 (2025).
20. Jiang, F. *et al.* GTE: a graph learning framework for prediction of T-cell receptors and epitopes binding specificity. *Brief. Bioinform.* **25**, bbae343 (2024).