HATIS: HIERARCHICAL ATOMISTIC GRAPH TRANSFER AND INTERFACE-AWARE STABLE LEARNING FOR TCR–PMHC RECOGNITION UNDER DISTRIBUTION SHIFTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Candidate screening for T-cell immunotherapy requires accurate prediction of T cell receptor (TCR) recognition of peptide–MHC (pMHC) complexes. Such screening often involves peptides and TCRs not represented in the training data, requiring models to generalize under shifts in peptide and TCR distributions. Despite recent advances, generalization across peptide distributions remains challenging, particularly when test peptides differ substantially from those used for training. A key limitation underlying this challenge is the limited coverage of TCR–pMHC interactions in training data, which restricts the learning of transferable recognition patterns. Models may also exploit spurious correlations arising from the frequent co-occurrence of molecular patterns across peptides, MHC molecules, and TCRs in training data, learning predictive shortcuts that fail under distribution shifts. We therefore propose HATIS, a framework integrating hierarchical atomistic graph transfer and interface-aware stable learning. To alleviate limited triplet supervision, we progressively transfer molecular representations and interaction knowledge across monomer, pairwise, and triadic levels to support TCR–pMHC recognition. To reduce reliance on spurious interaction correlations, we introduce interface-aware stable learning at both pairwise and triadic levels, using the molecular composition of interaction clusters to guide sample reweighting and interface-gated pooling. For systematic evaluation, we construct a benchmark covering peptide-only, TCR-only, and joint peptide–TCR distribution shifts, complemented by evaluation on an external test set. HATIS improves AUPR over the strongest baseline by 12.86–13.98 percentage points under peptide-only and joint shifts and by 4.65 points on the external test set, while remaining competitive under TCR-only shifts.

1 INTRODUCTION

T cell-mediated immunity is a central component of adaptive immunity. It is initiated by ternary TCR–peptide–MHC (TCR–pMHC) recognition in which a T cell receptor (TCR) recognizes a specific peptide presented by a major histocompatibility complex (MHC) molecule (Garboczi et al., 1996). Emerging immunotherapies, including personalized neoantigen vaccines and engineered TCR therapies, increasingly depend on identifying therapeutically relevant TCR–pMHC interactions (Ott et al., 2017; Foy et al., 2023). This creates a large-scale screening problem: given candidate peptides, patient-specific MHC alleles, and a TCR repertoire, one must identify peptide–MHC–TCR combinations likely to elicit T cell recognition. Experimental approaches such as tetramer staining, yeast display, and high-throughput T-cell assays can directly probe these interactions (Bentzen et al., 2016; Gee et al., 2018; Arnaud et al., 2022), but remain costly and difficult to scale. The combinatorial space defined by large peptide libraries, diverse MHC alleles, and highly variable TCR repertoires makes exhaustive experimental screening infeasible (Birnbaum et al., 2014). Computational models have therefore become essential for candidate prioritization and can be broadly grouped into two categories: pairwise predictors for peptide–MHC binding and TCR–peptide specificity prediction (Reynisson et al., 2020; O'Donnell et al., 2020; Montemurro et al., 2021; Que et al., 2025), and MHC-aware models that incorporate MHC context for ternary

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

TCR–peptide–MHC recognition (Springer et al., 2021; Lu et al., 2021; Feng et al., 2024; Zhang et al., 2024; Zhao et al., 2025).

Despite recent advances, candidate peptides and TCRs encountered in practical screening may be absent from the training data. Specifically, candidate peptides may arise from newly identified mutations or emerging pathogens, while patient-specific repertoires may contain previously unseen TCRs. These scenarios require models to generalize across shifts in peptide and TCR distributions, either separately or jointly, with peptide-level generalization particularly challenging when test peptides diverge substantially from those seen during training (Grazioli et al., 2022; Culka et al., 2026). Two factors further limit the model's generalization ability. First, the sparse coverage of peptide–MHC–TCR interactions in training data can restrict the learning of transferable recognition patterns (Weber et al., 2021). Second, without mechanisms to promote robust representation learning, models may rely on spurious correlations arising from repeated co-occurrence of molecular patterns, which can become unreliable under distribution shifts.

To address these challenges, we propose HATIS, a framework integrating **H**ierarchical **A**tomistic graph **T**ransfer and **I**nterface-aware **S**table learning. To alleviate limited triplet supervision, we progressively transfer molecular representations and interaction knowledge across monomer, pairwise, and triadic levels. Atomic representations are first learned through self-supervised monomer modeling, then refined by supervised pairwise interaction learning, and finally transferred to TCR–pMHC recognition. At the triadic level, the model builds on the learned pMHC interaction graph and peptide-conditioned TCR representations, rather than learning from sparse triplet data alone. Moreover, at both the pairwise and triadic levels, we design interface-aware stable learning, which leverages the molecular composition of interaction clusters to guide sample reweighting and interface-gated pooling. Together, these mechanisms reduce the reliance on spurious interaction correlations while promoting representations centered on cross-molecular interaction patterns.

To enable systematic evaluation under distribution shifts, we construct a benchmark covering peptide-only, TCR-only, and joint peptide–TCR shifts, complemented by evaluation on an external test set. Across these settings, HATIS outperforms five representative baselines under peptide-only and joint shifts, improving AUPR over the strongest baseline by 12.86–13.98 percentage points, and achieves a 4.65-point improvement on the external test set, while remaining competitive under TCR-only shifts.

Our main contributions are:

- **Hierarchical Atomistic Graph Transfer.** We develop a hierarchical framework that leverages richer lower-order supervision to transfer molecular representations and interaction knowledge across monomer, pairwise, and triadic levels for TCR–pMHC specificity prediction.
- **Interface-Aware Stable Learning.** We introduce Interface-Aware Stable Learning (IASL) for pairwise and triadic interaction learning, taking molecular composition of interaction clusters to guide sample reweighting and interface-gate pooling, thereby reducing reliance on unstable interaction correlations.
- **Systematic Evaluation under Distribution Shifts.** Across four evaluation settings, HATIS outperforms five baselines under peptide-only and joint shifts and on the external test set, while remaining competitive under TCR-only shifts. Ablations and representation analyses further demonstrate the effectiveness of hierarchical transfer and the generalization of learned representations.

2 Related Work

TCR–pMHC recognition prediction. Computational prediction of antigen recognition has evolved from pairwise interactions to joint TCR–pMHC modeling. Some predictors are developed for specific pairwise tasks. NetMHCpan and MHCflurry predict peptide–MHC binding (Reynisson et al., 2020; O'Donnell et al., 2020), while NetTCR-2.0 and TITAN predict TCR–peptide specificity (Montemurro et al., 2021; Weber et al., 2021). Subsequently, ERGO-II additionally supports MHC typing and TCR gene features (Springer et al., 2021). DeepAntigen uses graph-based interaction modeling for both pairwise tasks (Que et al., 2025). Recent three-body predictors, including EPIC-TRACE (Korpela et al., 2023), PISTE (Feng et al., 2024), and EPACT (Zhang et al., 2024),

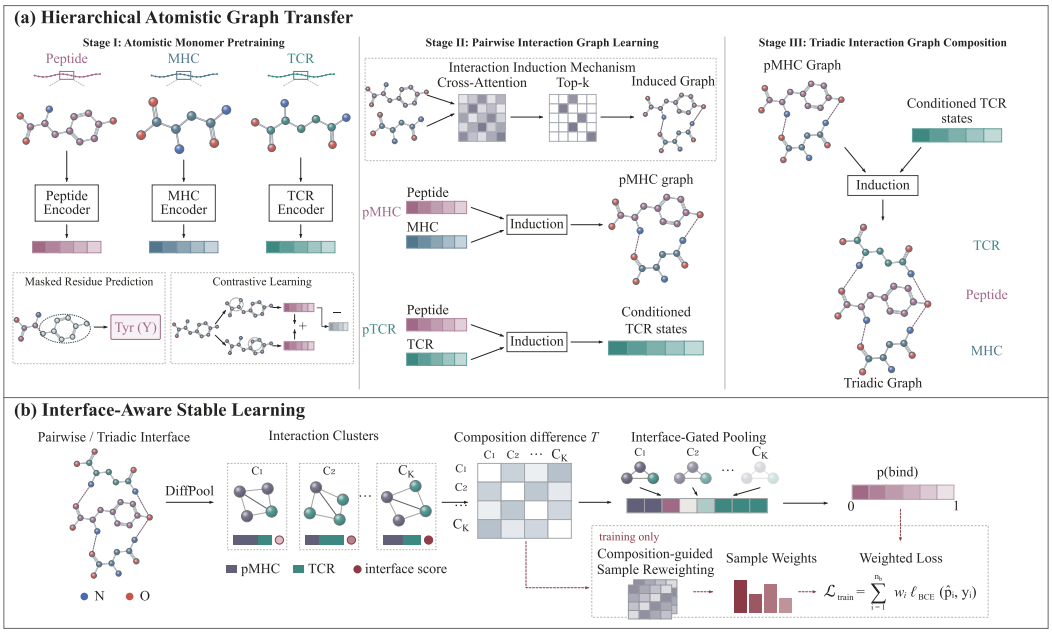


Figure 1: **Overview of HATIS.** (a) **Hierarchical Atomistic Graph Transfer:** A three-stage framework transfers atomistic representations and interaction graphs from monomer and pairwise learning to triadic recognition. (b) **Interface-Aware Stable Learning:** Cluster composition guides HSIC-based sample reweighting and interface-gated pooling for binding prediction.

jointly model peptide–MHC–TCR binding, while UniPMT unifies pairwise and three-body binding prediction (Zhao et al., 2025). Despite these advances, generalization under peptide-only and joint peptide–TCR shifts remains challenging under limited triplet supervision.

Stable learning under distribution shifts. Stable learning seeks reliable prediction when training correlations change at test time. For example, sample reweighting aims to reduce dependencies among input variables (Shen et al., 2020), and StableNet extends this principle to nonlinear dependencies in deep representations without requiring environment labels (Zhang et al., 2021). StableGNN applies reweighting to subgraph representations learned through differentiable pooling to improve graph out-of-distribution generalization (Fan et al., 2024). However, applying these approaches to molecular recognition requires accounting for the distinct molecular roles of interaction patterns when determining which dependencies to suppress.

3 METHOD

In this section, we introduce the proposed HATIS framework. First, we describe the formulation of the problem. Then, we present the Hierarchical Atomistic Graph Transfer for transferring molecular representations and pairwise interactions to triadic prediction (Section 3.2, Figure 1(a)). Finally, we show that Interface-Aware Stable Learning (IASL) operates at the pairwise and triadic levels through composition-guided sample reweighting and interface-gated pooling (Section 3.3, Figure 1(b)).

3.1 PROBLEM DEFINITION

We formulate TCR–pMHC specificity prediction as binary classification of TCR recognition of peptide–MHC complexes. Let a labeled example be $z_i = (p_i, m_i, t_i, y_i)$, where p_i, m_i, and t_i denote a peptide, an MHC allele, and a TCR β-chain complementarity-determining region 3 (CDR3) sequence, respectively, and $y_i \in \{0, 1\}$ is the corresponding dataset interaction label. The triplet training set is $\mathcal{D}_{\mathrm{tr}} = \{z_i\}_{i=1}^{N}$, with entity sets $\mathcal{P}_{\mathrm{tr}}$, $\mathcal{M}_{\mathrm{tr}}$, and $\mathcal{T}_{\mathrm{tr}}$. The model $f_\theta(p, m, t) \in \mathbb{R}$ produces a recognition logit used to rank candidate triplets. In addition to $\mathcal{D}_{\mathrm{tr}}$, the learner may use

monomer corpora $\mathcal{D}^s_{\text{mono}}$ for $s \in \{p, m, t\}$ and labeled pairwise corpora $\mathcal{D}^{pm}_{\text{pair}}$ and $\mathcal{D}^{pt}_{\text{pair}}$, which provide lower-order supervision for peptide–MHC binding and TCR–peptide specificity, respectively. Throughout the method, binding prediction denotes these pairwise and triplet interaction-label classification tasks; pMHC affinity regression is treated separately.

3.2 Hierarchical Atomistic Graph Transfer

To leverage supervision beyond triplet binding data, we organize learning into monomer, pairwise, and triadic levels. Training proceeds sequentially, with each level supplying pretrained representations or interaction graphs to the next.

Stage I: Atomistic monomer pretraining. We define each molecule $s \in \{p, m, t\}$ as an atom graph $G^s = (\mathcal{V}^s, \mathcal{E}^s)$, whose nodes encode atom features $\boldsymbol{X}^s$ and edges encode bond attributes $\boldsymbol{e}^s_{uv}$. This representation explicitly encodes the atoms and covalent bonds within each residue. A GINE–GPS encoder E^s (Hu et al., 2020; Rampášek et al., 2022) combines bond-aware message passing and global self-attention to produce $\boldsymbol{H}^s = E^s(G^s) \in \mathbb{R}^{|\mathcal{V}^s| \times d}$, with one d-dimensional representation for each atom node.

We pretrain each encoder with masked residue prediction, which reconstructs the amino-acid identity of a masked side chain from its surrounding atomic context, and InfoNCE-based graph contrastive learning, which aligns two complementary molecular views. The monomer objective is

$$\mathcal{L}_{\text{mono}} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{NCE}}, \tag{1}$$

where $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{NCE}}$ denote the reconstruction and InfoNCE losses, respectively, with λ balancing the two objectives. The pretrained encoders E^p, E^m, and E^t initialize the pairwise models.

Interaction induction mechanism. After Stage I, the interaction induction mechanism is used at both the pairwise and triadic levels to model interactions between two atom graphs and connect them into an interaction graph based on learned interaction strengths. It takes two component graphs G^A and G^B with atom representations $\boldsymbol{H}^A$ and $\boldsymbol{H}^B$ as input; each component can be a single molecule or a learned pMHC interaction graph. Bidirectional cross-attention with parameters ψ updates each atom representation using information from the other component. We combine the attention scores in both directions as $c^{A,B}_{ij} = \sqrt{a^{A,B}_{ij} a^{B,A}_{ji}}$, where $a^{A,B}_{ij}$ is the attention score from atom i in A to atom j in B. For each atom in A, we select its top-k partners in B according to $c^{A,B}_{ij}$ and add edges in both directions, giving the cross-molecular edge set $\mathcal{E}^{A\times B}_{\psi,k}$. We preserve the existing edges in both input graphs and add the cross-component edges to form the interaction graph:

$$\begin{aligned} G^{AB}_{\psi} &= \mathcal{I}_{\psi,k}(G^A, G^B), \\ \mathcal{V}^{AB} &= \mathcal{V}^A \cup \mathcal{V}^B, \\ \mathcal{E}^{AB}_{\psi,k} &= \mathcal{E}^A \cup \mathcal{E}^B \cup \mathcal{E}^{A\times B}_{\psi,k}. \end{aligned} \tag{2}$$

The operator $\mathcal{I}_{\psi,k}$ encompasses cross-attention, interaction scoring, and edge selection. The sets $\mathcal{V}^{AB}$ and $\mathcal{E}^{AB}_{\psi,k}$ define the output graph's topology; the graph also retains the updated atom representations and cross-molecular edge weights.

Stage II: Pairwise interaction graph learning. We train separate pMHC and peptide–TCR (pTCR) models, using the pretrained monomer encoders to obtain atom representations for each pair. The interaction induction mechanism constructs the two graphs with separate parameters: $G^{pm}_{\psi_{pm}} = \mathcal{I}_{\psi_{pm},k_{pm}}(G^p, G^m)$ and $G^{pt}_{\psi_{pt}} = \mathcal{I}_{\psi_{pt},k_{pt}}(G^p, G^t)$. Each interaction graph is processed by a separate GINE–DiffPool encoder $\mathcal{P}_{\omega_{AB}}$ (Hu et al., 2020; Ying et al., 2018), producing K cluster representations $\boldsymbol{X}^{AB}$ and soft atom-to-cluster assignments $\boldsymbol{S}^{AB}$ used by IASL: $(\boldsymbol{X}^{AB}, \boldsymbol{S}^{AB}) = \mathcal{P}_{\omega_{AB}}(G^{AB}_{\psi_{AB}})$. For prediction, interface-gated pooling (Section 3.3) aggregates the resulting clusters into a graph representation $\boldsymbol{g}^{AB}$, which a pair-specific binding head h_{AB} maps to a probability:

$$\widehat{p}^{AB} = \sigma\big(h_{AB}(\boldsymbol{g}^{AB})\big), \qquad AB \in \{pm, pt\}. \tag{3}$$

Both branches are trained with binding supervision, and the pMHC branch additionally receives affinity supervision through a separate regression head on its graph representation.

Stage III: Triadic interaction graph composition. After pairwise training, we freeze the trained pairwise models and encode each triplet through their monomer encoders and pairwise interaction modules. The pMHC predictor produces the complete interaction graph $G^{pm}_{\psi^*_{pm}}$, while the pTCR predictor supplies peptide-conditioned TCR states $\boldsymbol{H}^{t|p}_{\psi^*_{pt}}$, where ψ^*_{pm} and ψ^*_{pt} denote the learned pairwise interaction parameters. Since the pTCR branch lacks MHC context, we transfer its conditioned TCR states rather than its complete interaction graph and learn their interactions with pMHC at the triadic level. These states equip the original TCR graph to form $G^{t|p} = (\mathcal{V}^t, \mathcal{E}^t; \boldsymbol{H}^{t|p}_{\psi^*_{pt}})$. The interaction induction mechanism takes $G^{pm}_{\psi^*_{pm}}$ and $G^{t|p}$ as its two inputs, with new fusion parameters ψ_{tri} and a separately configured k_{tri}, to construct the triadic graph:

$$\begin{aligned} G^{pmt} &= \mathcal{I}_{\psi_{\mathrm{tri}},k_{\mathrm{tri}}}\left(G^{pm}_{\psi^*_{pm}}, G^{t|p}\right), \\ \mathcal{V}^{pm} &\subseteq \mathcal{V}^{pmt}, \qquad \mathcal{E}^{pm}_{\psi^*_{pm}} \subseteq \mathcal{E}^{pmt}. \end{aligned} \tag{4}$$

The inclusions preserve all nodes $\mathcal{V}^{pm}$ and edges $\mathcal{E}^{pm}_{\psi^*_{pm}}$ of the transferred pMHC graph while adding connections to the TCR graph. A separate GINE–DiffPool encoder and the Stage II readout procedure yield $\boldsymbol{g}^{pmt}$, giving the recognition logit $f_\theta(p, m, t) = h_{pmt}(\boldsymbol{g}^{pmt})$. Only the fusion attention, triadic encoder, interface gate, and binding head are trained with triplet supervision.

3.3 Interface-Aware Stable Learning

Fragments from different molecular components may frequently co-occur in the training data, leading the model to memorize spurious associations that may fail under distribution shifts. The K soft clusters produced by the GINE–DiffPool encoders in Stages II and III provide a natural unit for reducing such dependencies: each cluster aggregates atom-level information from the learned interaction graph into a local interaction pattern, allowing dependencies to be examined between patterns. However, not all inter-cluster dependencies are spurious. Clusters dominated by the same component may capture useful relationships within that component, while clusters containing atoms from both components may capture related interface patterns. IASL therefore uses cluster composition to guide sample reweighting, applying stronger dependence penalties to cluster pairs with different compositions and weaker penalties to those with similar compositions. It further uses interface-gated pooling to emphasize candidate interface patterns within each complex.

Molecular roles of interaction clusters. Each interaction cluster contains atoms from one or both molecular components. We describe each cluster by the proportion of its assignment mass contributed by each component, distinguishing clusters concentrated within one molecule from those spanning the interaction interface. For triadic graphs, the two components are pMHC and TCR; for pairwise graphs, they are peptide and MHC or peptide and TCR. For sample i, let $\mathcal{V}^c_i$ be the atoms of component c and S_{iuk} the soft assignment weight of atom u to cluster k. The proportion contributed by each component is

$$r_{ikc} = \frac{\sum_{u \in \mathcal{V}^c_i} S_{iuk}}{\sum_u S_{iuk}}, \qquad c \in \{1, 2\}. \tag{5}$$

Together, these proportions form the composition vector $\boldsymbol{r}_{ik} = (r_{ik1}, r_{ik2})^\top$.

For a mini-batch of n_b examples, we measure the difference between the molecular roles of clusters k and ℓ by their average cosine distance:

$$T_{k\ell} = \frac{1}{n_b} \sum_{i=1}^{n_b} \big[1 - \cos(\boldsymbol{r}_{ik}, \boldsymbol{r}_{i\ell})\big], \qquad T_{kk} = 0, \tag{6}$$

where $\cos(\cdot, \cdot)$ denotes cosine similarity. The resulting type matrix $\boldsymbol{T}$ assigns larger values to cluster pairs with more distinct compositions. For example, clusters concentrated on opposite molecular components have high $T_{k\ell}$, whereas clusters concentrated on the same component or sharing similar mixed compositions have low $T_{k\ell}$.

Composition-guided sample reweighting. We learn sample weights to reduce dependencies between clusters, with stronger penalties on cluster pairs whose molecular compositions differ more. These weights are then used in the binding loss to adjust each training example's contribution.

Concretely, let $\boldsymbol{X}_k = [\boldsymbol{x}_{1k}, \ldots, \boldsymbol{x}_{n_b k}]^\top \in \mathbb{R}^{n_b \times D}$ collect the D-dimensional representation of cluster k across the current mini-batch. The weight vector lies on the probability simplex $\Delta^{n_b-1} = \{\boldsymbol{w} \in \mathbb{R}_+^{n_b} : \sum_i w_i = 1\}$. We measure dependence with the Hilbert–Schmidt independence criterion (HSIC). We compute its empirical weighted form, denoted by $\mathrm{HSIC}(\boldsymbol{X}_k, \boldsymbol{X}_\ell, \boldsymbol{w})$, using radial basis function (RBF) kernels on $\mathrm{diag}(\boldsymbol{w})\boldsymbol{X}_k$ and $\mathrm{diag}(\boldsymbol{w})\boldsymbol{X}_\ell$; multiplication by $\mathrm{diag}(\boldsymbol{w})$ scales each row by its sample weight.

We use $T_{k\ell}$ to weight the dependence penalty for each cluster pair and optimize the sample weights as follows:

$$\begin{aligned} \boldsymbol{w}^* &= \arg\min_{\boldsymbol{w} \in \Delta^{n_b-1}} \left[\sum_{k<\ell} \max((T_{k\ell})^\alpha, \delta)\ \mathrm{HSIC}(\boldsymbol{X}_k, \boldsymbol{X}_\ell, \boldsymbol{w}) + \lambda_w \|\boldsymbol{w}\|_2^2 \right], \\ \mathcal{L}_{\mathrm{bind}}^{\mathrm{IASL}} &= \sum_{i=1}^{n_b} w_i^*\ \mathrm{BCE}(f_\theta(p_i, m_i, t_i), y_i)\,. \end{aligned} \tag{7}$$

Here, BCE denotes binary cross-entropy applied to logits. The exponent $\alpha > 0$ controls how composition differences translate into penalty weights, while δ sets their minimum value. The regularizer $\lambda_w \|\boldsymbol{w}\|_2^2$ discourages excessive sample-weight concentration.

At each training iteration, we first fix the current cluster representations and optimize one weight per example to reduce type-weighted dependence. We then fix these weights and update the predictor with the weighted binding loss. The sample weights are updated as the learned interaction representations change. Sample reweighting applies only to the training loss; at inference, the model uses interface-gated pooling and the binding head without HSIC optimization or sample weights.

Interface-gated pooling. Sample reweighting operates across examples; the interface gate instead controls the contribution of clusters within each complex. To emphasize candidate interface patterns, we assign greater pooling weights to clusters with more balanced molecular compositions, thereby limiting the influence of changes in lower-scoring clusters. We define the normalized interface score $\phi_{ik} = 2(1 - \max_c r_{ikc}) \in [0, 1]$: it is zero for a cluster drawn entirely from one component and reaches one when both components contribute equally. Using the logistic function σ with learned slope $a > 0$ and bias b, we compute

$$g_{ik} = \sigma(a\phi_{ik} + b), \quad a > 0, \qquad \boldsymbol{g}_i = \frac{\sum_{k=1}^K g_{ik} \boldsymbol{x}_{ik}}{\sum_{k=1}^K g_{ik}}, \tag{8}$$

The pooled representation $\boldsymbol{g}_i$ is passed to the binding head, yielding the probability $\hat{p}_i = \sigma(f_\theta(p_i, m_i, t_i))$. Since a is positive, clusters with more balanced contributions from the two molecular components receive larger pooling weights. The gate depends only on each complex's interface scores and remains active at inference. With scores fixed, it reduces the total pooling weight of lower-scoring clusters, limiting the worst-case effect of feature perturbations confined to those clusters relative to uniform mean pooling (Appendix B.2).

Theoretical analysis. We analyze how changes in inter-cluster dependence affect risk under a composition-guided recombination model. This reference distribution combines clusters from two independent source draws using a shared threshold on their molecular compositions, preserving each cluster's marginal distribution while preferentially separating clusters with distinct compositions. It provides an analytical model of selective dependence changes; it is not assumed to coincide with the target distribution.

Theorem 1 (composition-guided risk discrepancy). Let P_w be a reweighted distribution of cluster representations and Q_{comp} its composition-guided recombination distribution. For the same conditional expected loss m_f, denote $R_\nu(f) = \mathbb{E}_\nu m_f(Z)$. Assume m_f is a sum of single-cluster terms, pairwise terms $h_{k\ell} \in \mathcal{H}_k \otimes \mathcal{H}_\ell$, and a remainder bounded by ϵ_{app} on both supports. Here, $\mathcal{H}_k$ is the reproducing kernel Hilbert space associated with the kernel on cluster k. Define $B^2 = \sum_{k<\ell} \|h_{k\ell}\|^2$ and

$$J_T(P_w) = \sum_{k<\ell} T_{k\ell}\, \mathrm{HSIC}_{P_w}(Z_k, Z_\ell), \qquad J_{\mathrm{unif}}(P_w) = \sum_{k<\ell} \mathrm{HSIC}_{P_w}(Z_k, Z_\ell).$$

Then

$$\left|R_{Q_{\mathrm{comp}}}(f) - R_{P_w}(f)\right| \le B\sqrt{J_T(P_w)} + 2\epsilon_{\mathrm{app}} \le B\sqrt{J_{\mathrm{unif}}(P_w)} + 2\epsilon_{\mathrm{app}}. \tag{9}$$

The bound depends on the aggregate pairwise loss norm B and the remainder bound ϵ_{app}, which accounts for higher-order effects and approximation error. For $B > 0$, the composition-guided upper bound is strictly smaller than the uniform bound whenever a pair with $T_{k\ell} < 1$ has positive HSIC. Dependencies among similarly composed clusters contribute less to this bound because they are less likely to be separated during recombination. This motivates composition-guided dependence penalties under the specified shift model; the comparison is between bounds for the same predictor and distributions, not between the prediction risks of separately trained algorithms. The assumptions and proof are provided in Appendix B.1.

4 Experiments

4.1 Datasets and evaluation splits

Existing benchmarks evaluate complementary generalization settings (Grazioli et al., 2022; Weber et al., 2021; Zhang et al., 2024; Feng et al., 2024), but differ in held-out entities, similarity controls, and data sources. Our benchmark jointly evaluates peptide-only, TCR-only, and joint shifts using cluster-level hold-outs on both molecular axes, complemented by an external test set.

Data and partitions. We aggregate public monomer, pairwise, and triplet data (Feng et al., 2024; Zhang et al., 2024; Shugay et al., 2018), yielding a triplet training pool of 90,169 positive examples. Peptide and CDR3β entities are clustered separately using MMseqs2 (Steinegger & Söding, 2017) with minimum sequence identity 0.4; holding out peptide clusters, TCR clusters, or both defines the three cluster-based settings. For external evaluation, we use the public TCRCube pan-peptide test set released by Culka et al. (2026). We retain its supplied labels and exclude its peptide and TCR entities from triplet training. Details are provided in Appendix C.

4.2 Experimental setup

Baselines and implementation. We compare HATIS with five representative TCR–peptide or TCR–pMHC predictors: NetTCR-2.0 (Montemurro et al., 2021), ERGO-II (Springer et al., 2021), EPIC-TRACE (Korpela et al., 2023), PISTE (Feng et al., 2024), and EPACT (Zhang et al., 2024). Baselines are trained on the benchmark training pool using method-specific preprocessing and training procedures. We report AUROC and AUPR, using AUPR as the primary metric for the imbalanced binding prediction task. Descriptions of the baselines and implementation settings are provided in Appendices D and E, respectively.

4.3 Main results

Table 1 summarizes the results across four generalization settings. We have the following observations: (1) HATIS achieves the best AUROC and AUPR under peptide-only and joint shifts, demonstrating its effectiveness when recognition must transfer to unseen peptides. (2) Its advantage extends to the external test set, supporting generalization beyond the benchmark's cluster-based partitions. (3) Under TCR-only shifts, HATIS remains competitive with the baselines. The stronger performance of most methods in this setting is consistent with shared recognition motifs among TCRs targeting the same pMHC (Dash et al., 2017; Glanville et al., 2017).

4.4 Ablation and analysis

Effect of hierarchical transfer. To assess hierarchical transfer, we remove monomer pretraining, pMHC transfer, pTCR transfer, or all pretraining. As shown in Table 2(A), Full substantially outperforms training from scratch across all settings, demonstrating the effectiveness of hierarchical pretraining. At the monomer level, removing monomer pretraining degrades most metrics, especially on the external test set, highlighting the importance of learning transferable molecular representations through self-supervision. At the pairwise level, removing pMHC transfer produces the largest performance drops among individual-stage ablations, underscoring its critical role in transferring

Table 1: **Main results** (%). AUPR is the primary metric; bold and underline denote the best and second-best result per row. The external test set is peptide–TCR entity-disjoint from triplet training.

Test set	Metric	NetTCR-2.0	ERGO-II	EPIC-TRACE	PISTE	EPACT	**HATIS**
TCR-only shift	AUROC	62.43	92.67	91.21	<u>92.71</u>	**93.14**	92.34
	AUPR	25.71	<u>69.50</u>	61.92	68.26	**71.32**	66.24
Peptide-only shift	AUROC	53.38	65.80	64.47	<u>67.20</u>	66.39	**79.53**
	AUPR	19.16	27.76	24.72	30.54	<u>33.21</u>	**46.07**
Joint shift	AUROC	52.10	62.91	58.98	<u>64.73</u>	64.35	**79.51**
	AUPR	19.27	25.51	21.31	28.49	<u>30.56</u>	**44.54**
External test	AUROC	<u>57.00</u>	39.71	49.48	51.08	52.05	**62.51**
	AUPR	<u>21.74</u>	13.98	16.39	16.70	19.99	**26.39**

Table 2: **Ablation study** (%). Parentheses give percentage-point differences from **Full (ours)**. (A) disables transfer from earlier stages; (B) removes IASL at Stage III only or Stages II+III.

	TCR-only shift		Peptide-only shift		Joint shift		External test	
Variant	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR	AUROC	AUPR
A. Hierarchical transfer								
No pretraining	89.83 (↓2.51)	58.34 (↓7.90)	65.12 (↓14.41)	26.88 (↓19.19)	64.15 (↓15.36)	26.11 (↓18.43)	52.98 (↓9.53)	17.15 (↓9.24)
w/o monomer pretraining	92.30 (↓0.04)	**66.86** (↑0.62)	78.51 (↓1.02)	45.01 (↓1.06)	78.82 (↓0.69)	44.59 (↑0.05)	56.19 (↓6.32)	19.28 (↓7.11)
w/o pMHC transfer	90.19 (↓2.15)	60.52 (↓5.72)	62.17 (↓17.36)	26.79 (↓19.28)	61.65 (↓17.86)	28.39 (↓16.15)	47.87 (↓14.64)	16.33 (↓10.06)
w/o pTCR transfer	92.24 (↓0.10)	66.80 (↑0.56)	**79.82** (↑0.29)	**46.99** (↑0.92)	**80.27** (↑0.76)	**46.53** (↑1.99)	61.04 (↓1.47)	24.51 (↓1.88)
Full model (ours)	**92.34**	66.24	79.53	46.07	79.51	44.54	**62.51**	**26.39**
B. IASL								
w/o IASL (III)	92.32 (↓0.02)	66.04 (↓0.20)	79.36 (↓0.17)	45.84 (↓0.24)	79.11 (↓0.40)	43.66 (↓0.88)	61.82 (↓0.69)	**26.53** (↑0.14)
w/o IASL (II+III)	**92.37** (↑0.03)	**66.79** (↑0.55)	78.83 (↓0.71)	45.65 (↓0.42)	78.99 (↓0.52)	42.46 (↓2.09)	55.36 (↓7.15)	22.10 (↓4.29)
Full IASL (ours)	92.34	66.24	**79.53**	**46.07**	**79.51**	**44.54**	**62.51**	26.39

peptide–MHC interaction knowledge to triadic recognition. In contrast, pTCR transfer has smaller, mixed effects and reduces performance under peptide-only and joint shifts, possibly because the absence of MHC context during pTCR pretraining limits its transferability to triadic recognition.

Effect of IASL. To assess IASL, we remove it from triadic training alone (w/o IASL (III)) or from both pairwise and triadic training (w/o IASL (II+III)). As shown in Table 2(B), removing triadic IASL generally reduces performance under peptide-only and joint shifts and on the external test set. Removing IASL from both stages leads to larger declines accordingly. These results highlight the importance of IASL in both pairwise and triadic training under peptide distribution shifts, with greater benefits when applied across both stages. Furthermore, the larger overall gains under joint shifts and on the external test set suggest that IASL becomes increasingly effective as distribution shifts intensify, highlighting its importance for generalization in more challenging settings.

Representation analysis. To examine representation transfer, we compare HATIS with PISTE and EPACT using pre-classifier embeddings. Figure 2(A) shows that peptide shifts displace test examples farther from their training neighborhoods than TCR-only shifts. Despite this displace-

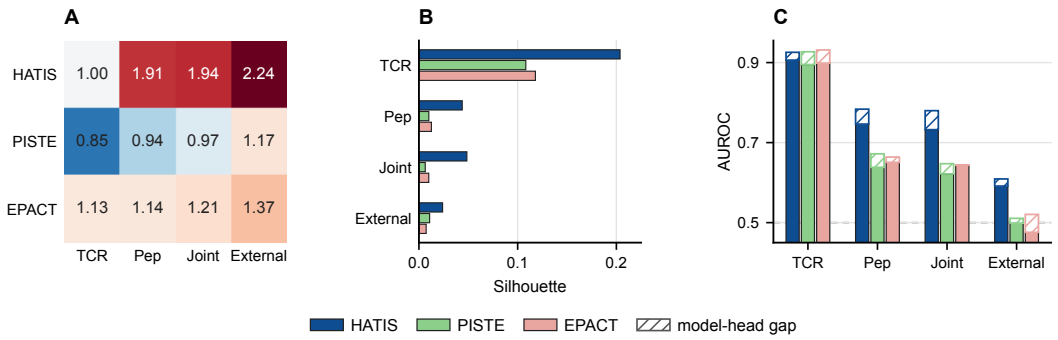


Figure 2: Representation geometry under distribution shifts. (A) Label-free neighborhood Gap, compared across splits within each model. (B) Binding-label silhouette coefficient. (C) Centroid-based transfer AUROC; hatched caps show the gap to learned-head AUROC. TCR, Pep, Joint, and External denote TCR-only, peptide-only, joint shifts, and the external test set, respectively.

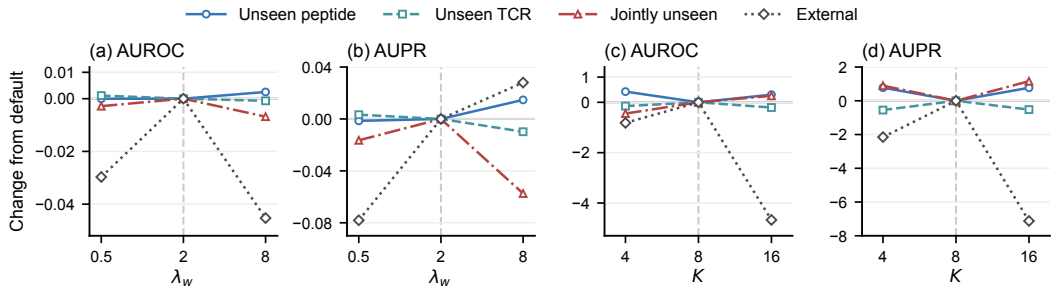


Figure 3: Parameter sensitivity of the triadic stage. (a,b) Varying λ_w with $K = 8$; (c,d) varying K with $\lambda_w = 2$. Changes are in percentage points relative to the shared default run.

ment, HATIS exhibits clearer separation between binding and non-binding examples, as measured by the silhouette coefficient (B). It also achieves higher centroid-based transfer AUROC across the three settings involving peptide shifts (C). Together, these results indicate that HATIS preserves more transferable binding-class structure under peptide shifts. Probe definitions are provided in Appendix E.4.

Parameter study. We vary the triadic sample-weight regularization coefficient $\lambda_w \in \{0.5, 2, 8\}$ and final DiffPool cluster count $K \in \{4, 8, 16\}$, changing one parameter at a time while fixing the pairwise stages. As shown in Figure 3, performance changes little across the tested λ_w values. Cluster count has a stronger, setting-dependent effect: compared with $K = 8$, both $K = 4$ and $K = 16$ improve peptide-only and joint-shift AUPR but reduce performance on the external test set. This suggests that finer interface partitioning does not necessarily yield more transferable representations.

5 Conclusion

In this paper, we investigate the limited generalization of TCR–pMHC predictors to unseen peptides and peptide–TCR combinations. We establish a unified benchmark that controls molecular identity and sequence similarity to distinguish peptide, TCR, and joint shifts, complemented by evaluation on an external test set. To improve generalization, we propose HATIS, which transfers molecular representations and interaction knowledge from monomer and pairwise supervision to triadic recognition and incorporates IASL to reduce reliance on spurious interaction correlations. Experiments show that HATIS outperforms representative baselines on unseen peptides, jointly unseen pairs, and the external test set, while remaining competitive when only TCRs are unseen. Together, these results highlight the importance of leveraging hierarchical interaction knowledge and stable representation learning for robust TCR–pMHC prediction under realistic distribution shifts.

AI USE STATEMENT

We used large language models to assist with manuscript writing and revision, including improving clarity, refining sentence structure, reducing redundancy, and maintaining consistent terminology. The authors directed the revisions and retain full responsibility for the scientific claims, interpretation of results, and final content of the paper.

ETHICS STATEMENT

This work uses existing public molecular sequence and binding datasets and does not involve new human-subject recruitment or animal experiments. The proposed model is intended for research; its predictions require experimental validation before biomedical use. Biases in the source datasets may limit generalization across populations and molecular contexts.

REPRODUCIBILITY STATEMENT

We provide anonymized code and the benchmark in the supplementary material submitted through OpenReview. Dataset sources and benchmark construction are described in Appendix C, and implementation details, hyperparameters, and the software and hardware environment are provided in Appendix E. The assumptions and proofs supporting our theoretical analysis are given in Appendix B.

REFERENCES

Marion Arnaud, Johanna Chiffelle, Raphael Genolet, et al. Sensitive identification of neoantigens and cognate TCRs in human solid tumors. *Nature Biotechnology*, 40(5):656–660, 2022. doi: 10.1038/s41587-021-01072-6.

Amalie Kai Bentzen, Andrea Marion Marquard, Rikke Lyngaa, et al. Large-scale detection of antigen-specific T cells using peptide–MHC-I multimers labeled with DNA barcodes. *Nature Biotechnology*, 34(10):1037–1045, 2016. doi: 10.1038/nbt.3662.

Michael E. Birnbaum, Juan L. Mendoza, Dhruv K. Sethi, et al. Deconstructing the Peptide-MHC Specificity of T Cell Recognition. *Cell*, 157(5):1073–1087, 2014. doi: 10.1016/j.cell.2014.03.047.

Martin Culka, Nicolas W. Lounsbury, William Thrift, et al. Predicting specificity of TCR-pMHC interactions using machine-learning and biophysical models. *Cell Systems*, 17(9):101700, 2026. doi: 10.1016/j.cels.2026.101700.

Pradyot Dash, Andrew J. Fiore-Gartland, Tomer Hertz, et al. Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature*, 547(7661):89–93, 2017. doi: 10.1038/nature22383.

Shaohua Fan, Xiao Wang, Chuan Shi, et al. Generalizing graph neural networks on out-of-distribution graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):322–337, 2024. doi: 10.1109/TPAMI.2023.3321097. URL https://arxiv.org/abs/2111.10657.

Ziyan Feng, Jingyang Chen, Youlong Hai, et al. Sliding-attention transformer neural architecture for predicting T cell receptor–antigen–human leucocyte antigen binding. *Nature Machine Intelligence*, 6(10):1216–1230, 2024. doi: 10.1038/s42256-024-00901-y.

Susan P. Foy, Kyle Jacoby, Daniela A. Bota, et al. Non-viral precision T cell receptor replacement for personalized cell therapy. *Nature*, 615(7953):687–696, 2023. doi: 10.1038/s41586-022-05531-1.

David N. Garboczi, Partho Ghosh, Ursula Utz, et al. Structure of the complex between human T-cell receptor, viral peptide and HLA-A2. *Nature*, 384(6605):134–141, 1996. doi: 10.1038/384134a0.

Marvin H. Gee, Arnold Han, Shane M. Lofgren, et al. Antigen identification for orphan T cell receptors expressed on tumor-infiltrating lymphocytes. *Cell*, 172(3):549–563.e16, 2018. doi: 10.1016/j.cell.2017.11.043.

Jacob Glanville, Huang Huang, Allison Nau, et al. Identifying specificity groups in the T cell receptor repertoire. *Nature*, 547(7661):94–98, 2017. doi: 10.1038/nature22976.

Filippo Grazioli, Anja Mösch, Pierre Machart, et al. On TCR binding predictors failing to generalize to unseen peptides. *Frontiers in Immunology*, 13:1014256, 2022. doi: 10.3389/fimmu.2022.1014256.

Arthur Gretton, Olivier Bousquet, Alex Smola, et al. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory*, volume 3734 of *Lecture Notes in Computer Science*, pp. 63–77, 2005. doi: 10.1007/11564089_7.

Weihua Hu, Bowen Liu, Joseph Gomes, et al. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJlWWJSFDH.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Kendall_Multi-Task_Learning_Using_CVPR_2018_paper.html.

Dani Korpela, Emmi Jokinen, Alexandru Dumitrescu, et al. EPIC-TRACE: Predicting TCR binding to unseen epitopes using attention and contextualized embeddings. *Bioinformatics*, 39(12):btad743, 2023. doi: 10.1093/bioinformatics/btad743.

Tianshi Lu, Ze Zhang, James Zhu, et al. Deep learning-based prediction of the T cell receptor–antigen binding specificity. *Nature Machine Intelligence*, 3(10):864–875, 2021. doi: 10.1038/s42256-021-00383-2.

Alessandro Montemurro, Viktoria Schuster, Helle Rus Povlsen, et al. NetTCR-2.0 enables accurate prediction of TCR-peptide binding by using paired TCRα and β sequence data. *Communications Biology*, 4(1):1060, 2021. doi: 10.1038/s42003-021-02610-3.

Timothy J. O'Donnell, Alex Rubinsteyn, and Uri Laserson. MHCflurry 2.0: Improved pan-allele prediction of MHC class I-presented peptides by incorporating antigen processing. *Cell Systems*, 11(1):42–48.e7, 2020. doi: 10.1016/j.cels.2020.06.010.

Patrick A. Ott, Zhuting Hu, Derin B. Keskin, et al. An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature*, 547(7662):217–221, 2017. doi: 10.1038/nature22991.

Jinhao Que, Guangfu Xue, Tao Wang, et al. Identifying T cell antigen at the atomic level with graph convolutional network. *Nature Communications*, 16(1):5171, 2025. doi: 10.1038/s41467-025-60461-6.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, et al. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14501–14515, 2022. doi: 10.52202/068431-1054. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/5d4834a159f1547b267a05a4e2b7cf5e-Abstract-Conference.html.

Birkir Reynisson, Bruno Alvarez, Sinu Paul, et al. NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. *Nucleic Acids Research*, 48(W1):W449–W454, 2020. doi: 10.1093/nar/gkaa379.

Zheyan Shen, Peng Cui, Tong Zhang, et al. Stable learning via sample reweighting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5692–5699, 2020. doi: 10.1609/aaai.v34i04.6024. URL https://ojs.aaai.org/index.php/AAAI/article/view/6024.

Mikhail Shugay, Dmitriy V Bagaev, Ivan V Zvyagin, et al. VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Research*, 46(D1):D419–D427, 2018. doi: 10.1093/nar/gkx760.

Ido Springer, Nili Tickotsky, and Yoram Louzoun. Contribution of T cell receptor alpha and beta CDR3, MHC typing, V and J genes to peptide binding prediction. *Frontiers in Immunology*, 12: 664514, 2021. doi: 10.3389/fimmu.2021.664514.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017. doi: 10.1038/nbt.3988.

Anna Weber, Jannis Born, and María Rodriguez Martínez. TITAN: T-cell receptor specificity prediction with bimodal attention networks. *Bioinformatics*, 37(Supplement_1):i237–i244, 2021. doi: 10.1093/bioinformatics/btab294.

Zhitao Ying, Jiaxuan You, Christopher Morris, et al. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/e77dbaf6759253c7c6d0efc5690369c7-Abstract.html.

Xingxuan Zhang, Peng Cui, Renzhe Xu, et al. Deep stable learning for out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5372–5382, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Zhang_Deep_Stable_Learning_for_Out-of-Distribution_Generalization_CVPR_2021_paper.html.

Yumeng Zhang, Zhikang Wang, Yunzhe Jiang, et al. Epitope-anchored contrastive transfer learning for paired CD8+ T cell receptor–antigen recognition. *Nature Machine Intelligence*, 6(11):1344–1358, 2024. doi: 10.1038/s42256-024-00913-8.

Yunxiang Zhao, Jijun Yu, Yixin Su, et al. A unified deep framework for peptide–major histocompatibility complex–T cell receptor binding prediction. *Nature Machine Intelligence*, 7(4):650–660, 2025. doi: 10.1038/s42256-025-01002-0.

A PRELIMINARIES

Differentiable Pooling (DiffPool). DiffPool learns hierarchical graph representations by softly assigning nodes to clusters (Ying et al., 2018). Given node features $\boldsymbol{H} \in \mathbb{R}^{n\times d}$ and an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, a pooling layer computes

$$\begin{aligned} \boldsymbol{Z} &= \mathrm{GNN}_{\mathrm{emb}}(\boldsymbol{A}, \boldsymbol{H}), \qquad \boldsymbol{S} = \mathrm{softmax}_{\mathrm{row}}(\mathrm{GNN}_{\mathrm{pool}}(\boldsymbol{A}, \boldsymbol{H}))\,, \\ \boldsymbol{X} &= \boldsymbol{S}^\top \boldsymbol{Z}, \qquad\qquad\qquad \boldsymbol{A}' = \boldsymbol{S}^\top \boldsymbol{A}\boldsymbol{S}. \end{aligned} \tag{10}$$

Here, $\boldsymbol{Z} \in \mathbb{R}^{n\times D}$ contains node embeddings, $\boldsymbol{S} \in \mathbb{R}^{n\times K}$ contains soft assignments to K clusters, and $\boldsymbol{X} \in \mathbb{R}^{K\times D}$ contains the resulting cluster representations. The pooled features $\boldsymbol{X}$ and adjacency matrix $\boldsymbol{A}'$ define a coarsened graph for subsequent layers. For multiple pooling layers, multiplying their assignment matrices yields the assignment from original nodes to final clusters. For graph i, S_{iuk} denotes the assignment of node u to final cluster k, and $\boldsymbol{x}_{ik} \in \mathbb{R}^D$ denotes the representation of that cluster.

Stable Learning. Sample-reweighting approaches to stable learning aim to reduce reliance on spurious feature dependencies under distribution shifts (Shen et al., 2020; Zhang et al., 2021). They alternate between learning sample weights that reduce representation dependence and updating the predictive model with a weighted empirical loss. StableGNN extends this principle to graph prediction by treating learned subgraph representations as the variables to be decorrelated (Fan et al., 2024). Specifically, differentiable pooling maps each graph to K cluster representations, and each cluster is treated as a multidimensional variable rather than decorrelating individual feature coordinates. For a mini-batch of n_b graphs, let $\boldsymbol{X}_k = [\boldsymbol{x}_{1k}, \ldots, \boldsymbol{x}_{n_b k}]^\top \in \mathbb{R}^{n_b\times D}$ collect the representations of cluster k across samples. The two alternating optimization problems are

$$\begin{aligned} \boldsymbol{w}^{(t+1)} &\in \arg\min_{\boldsymbol{w}\in\Delta^{n_b-1}} \sum_{k<\ell} \mathrm{HSIC}\Big(\boldsymbol{X}_k^{(t)}, \boldsymbol{X}_\ell^{(t)}, \boldsymbol{w}\Big)\,, \\ \theta^{(t+1)} &\in \arg\min_{\theta} \sum_{i=1}^{n_b} w_i^{(t+1)} \mathcal{L}(f_\theta(G_i), y_i)\,, \end{aligned} \tag{11}$$

where $\Delta^{n_b-1} = \{\boldsymbol{w} \in \mathbb{R}_+^{n_b} : \sum_i w_i = 1\}$, $\mathrm{HSIC}(\boldsymbol{X}_k, \boldsymbol{X}_\ell, \boldsymbol{w})$ denotes the sample-weighted HSIC criterion, and θ includes both representation and prediction parameters. The inner weight optimization fixes the current representations $\boldsymbol{X}_k^{(t)}$ and learns one shared weight per graph to reduce dependence across cluster pairs. The outer model optimization fixes these weights and updates the predictor using the weighted loss $\mathcal{L}$. In practice, these objectives are optimized by alternating gradient updates. Sample-weight optimization is used only during training.

B PROOFS

B.1 PROOF OF THEOREM 1

Consider a fixed predictor, representation map, and composition table $\{\boldsymbol{r}_{ik}\}$, where $\boldsymbol{r}_{ik} \in \mathbb{R}_+^2$ and its entries sum to one. For n_b composition profiles, define

$$T_{k\ell} = \frac{1}{n_b} \sum_{i=1}^{n_b} [1 - \cos(\boldsymbol{r}_{ik}, \boldsymbol{r}_{i\ell})]\,. \tag{12}$$

Let P_w be the reweighted source law of consistently indexed clusters $Z = (Z_1, \ldots, Z_K)$. For bounded measurable kernels with separable RKHSs $\mathcal{H}_k$ and measurable feature maps φ_k, assume the same conditional expected loss on the source and recombination supports admits the integrable decomposition

$$m_f(z) = c + \sum_k h_k(z_k) + \sum_{k<\ell} h_{k\ell}(z_k, z_\ell) + r(z), \qquad |r(z)| \le \epsilon_{\mathrm{app}}, \tag{13}$$

where $h_{k\ell} \in \mathcal{H}_k \otimes \mathcal{H}_\ell$. Write $b_{k\ell} = \|h_{k\ell}\|$, $B^2 = \sum_{k<\ell} b_{k\ell}^2$, and $R_\nu(f) = \mathbb{E}_\nu m_f(Z)$.

To construct Q_{comp}, write $\boldsymbol{r}_{ik}/\|\boldsymbol{r}_{ik}\|_2 = (\cos\theta_{ik}, \sin\theta_{ik})^\top$ with $\theta_{ik} \in [0, \pi/2]$. Draw $I \sim \mathrm{Unif}\{1, \dots, n_b\}$, $U \sim \mathrm{Unif}[0, \pi/2]$, and $Z^{(0)}, Z^{(1)} \sim P_w$, all mutually independently, and set

$$\widetilde{Z}_k = \begin{cases} Z_k^{(0)}, & \theta_{Ik} \le U, \\ Z_k^{(1)}, & \theta_{Ik} > U, \end{cases} \qquad Q_{\mathrm{comp}} = \mathrm{Law}(\widetilde{Z}). \tag{14}$$

Since the choice of source draw is independent of its values, each $\widetilde{Z}_k$ has marginal law $P_{w,k}$. For a pair (k, ℓ), the two coordinates come from different source draws exactly when U lies between θ_{Ik} and $\theta_{I\ell}$. Writing $d_{ik\ell} = |\theta_{ik} - \theta_{i\ell}|$, the probability of this event is

$$q_{k\ell} = \frac{1}{n_b} \sum_{i=1}^{n_b} \frac{2 d_{ik\ell}}{\pi}. \tag{15}$$

To relate this probability to $T_{k\ell}$, concavity of sine on $[0, \pi/4]$ gives

$$\sin(d/2) \ge \frac{\sqrt{2} d}{\pi}, \qquad \left(\frac{2d}{\pi}\right)^2 \le 2\sin^2(d/2) = 1 - \cos d, \qquad 0 \le d \le \pi/2. \tag{16}$$

Applying Jensen's inequality to the average defining $q_{k\ell}$, we obtain

$$q_{k\ell}^2 \le \frac{1}{n_b} \sum_i \left(\frac{2 d_{ik\ell}}{\pi}\right)^2 \le \frac{1}{n_b} \sum_i (1 - \cos d_{ik\ell}) = T_{k\ell} \le 1. \tag{17}$$

When separated, the two coordinates come from independent source draws; otherwise they retain their source joint law. Thus

$$Q_{\mathrm{comp},k\ell} = (1 - q_{k\ell}) P_{w,k\ell} + q_{k\ell} (P_{w,k} \otimes P_{w,\ell}). \tag{18}$$

Let $\mu_k = \mathbb{E}_{P_w} \varphi_k(Z_k)$ and $C_{k\ell} = \mathbb{E}_{P_w}[\varphi_k(Z_k) \otimes \varphi_\ell(Z_\ell)] - \mu_k \otimes \mu_\ell$. The population HSIC satisfies $H_{k\ell} := \mathrm{HSIC}_{P_w}(Z_k, Z_\ell) = \|C_{k\ell}\|^2$ (Gretton et al., 2005). By the reproducing property in $\mathcal{H}_k \otimes \mathcal{H}_\ell$,

$$\mathbb{E}_{P_{w,k\ell}} h_{k\ell} - \mathbb{E}_{P_{w,k} \otimes P_{w,\ell}} h_{k\ell} = \langle h_{k\ell}, C_{k\ell} \rangle. \tag{19}$$

Substituting the pairwise mixture law and applying Cauchy–Schwarz therefore gives

$$\begin{aligned} \left|\mathbb{E}_{Q_{\mathrm{comp}}} h_{k\ell} - \mathbb{E}_{P_w} h_{k\ell}\right| &= q_{k\ell} \left|\langle h_{k\ell}, C_{k\ell} \rangle\right| \\ &\le q_{k\ell} b_{k\ell} \sqrt{H_{k\ell}}. \end{aligned} \tag{20}$$

Since the single-cluster marginals are unchanged, the constant and single-cluster terms cancel in the risk difference:

$$\begin{aligned} R_{Q_{\mathrm{comp}}}(f) - R_{P_w}(f) &= \sum_{k<\ell} \left(\mathbb{E}_{Q_{\mathrm{comp}}} h_{k\ell} - \mathbb{E}_{P_w} h_{k\ell}\right) + \mathbb{E}_{Q_{\mathrm{comp}}} r - \mathbb{E}_{P_w} r, \\ \left|\mathbb{E}_{Q_{\mathrm{comp}}} r - \mathbb{E}_{P_w} r\right| &\le \mathbb{E}_{Q_{\mathrm{comp}}} |r| + \mathbb{E}_{P_w} |r| \le 2\epsilon_{\mathrm{app}}. \end{aligned} \tag{21}$$

Define $J_T(P_w) = \sum_{k<\ell} T_{k\ell} H_{k\ell}$ and $J_{\mathrm{unif}}(P_w) = \sum_{k<\ell} H_{k\ell}$. Combining the pairwise bounds with $q_{k\ell}^2 \le T_{k\ell}$ and applying Cauchy–Schwarz again, we obtain

$$\begin{aligned} |R_{Q_{\mathrm{comp}}}(f) - R_{P_w}(f)| &\le \sum_{k<\ell} q_{k\ell} b_{k\ell} \sqrt{H_{k\ell}} + 2\epsilon_{\mathrm{app}} \\ &\le \sum_{k<\ell} b_{k\ell} \sqrt{T_{k\ell} H_{k\ell}} + 2\epsilon_{\mathrm{app}} \\ &\le \left(\sum_{k<\ell} b_{k\ell}^2\right)^{1/2} \left(\sum_{k<\ell} T_{k\ell} H_{k\ell}\right)^{1/2} + 2\epsilon_{\mathrm{app}} \\ &= B \sqrt{J_T(P_w)} + 2\epsilon_{\mathrm{app}} \\ &\le B \sqrt{J_{\mathrm{unif}}(P_w)} + 2\epsilon_{\mathrm{app}}. \end{aligned} \tag{22}$$

The last inequality follows from $0 \le T_{k\ell} \le 1$ and $H_{k\ell} \ge 0$. Moreover,

$$J_{\mathrm{unif}}(P_w) - J_T(P_w) = \sum_{k<\ell} (1 - T_{k\ell}) H_{k\ell}. \tag{23}$$

For $B > 0$, the composition-guided bound is therefore strictly smaller whenever some pair has $T_{k\ell} < 1$ and $H_{k\ell} > 0$. This completes the proof. □

The analysis holds for fixed representations, composition profiles, kernels, and predictor, under a common conditional loss on both supports. The recombination distribution is an analytical construction and need not coincide with a target environment. The bound concerns population HSIC under P_w; the implemented training criterion instead applies RBF kernels to sample-scaled features. Thus, the result motivates composition-guided dependence penalties.

B.2 Proposition and Proof: Sensitivity of Interface-Gated Pooling

Proposition 1 (fixed-score sensitivity of interface-gated pooling). Fix the composition scores and gate parameters $a > 0$ and b, and consider a set $\mathcal{N}$ of m clusters with scores at most ϕ_{low}, while the other $K - m$ scores are at least $\phi_{\text{high}} \geq \phi_{\text{low}}$, with $0 < m < K$. If only features in $\mathcal{N}$ change, each by norm at most ρ, then the interface-gated pooled representation satisfies

$$\|\boldsymbol{g}' - \boldsymbol{g}\| \leq \rho \, \frac{m\sigma(a\phi_{\text{low}} + b)}{m\sigma(a\phi_{\text{low}} + b) + (K - m)\sigma(a\phi_{\text{high}} + b)} \leq \rho \frac{m}{K}. \tag{24}$$

The final term is the worst-case sensitivity of uniform mean pooling under the same perturbations. By reducing the total pooling weight of low-interface clusters, the gate yields a strictly tighter bound when $\rho > 0$ and $\phi_{\text{low}} < \phi_{\text{high}}$. The comparison holds with composition scores and gate parameters fixed.

Proof. Suppress the sample index and write $\phi_k = \phi_{ik}$, $g_k = \sigma(a\phi_k + b)$, and $\pi_k = g_k / \sum_j g_j$. For fixed scores and finite gate parameters with $a > 0$, the pooling weights are positive and fixed, and $\boldsymbol{g} = \sum_k \pi_k \boldsymbol{x}_k$. Let $g_{\text{low}} = \sigma(a\phi_{\text{low}} + b)$ and $g_{\text{high}} = \sigma(a\phi_{\text{high}} + b)$. Monotonicity of the sigmoid gives $g_k \leq g_{\text{low}}$ for $k \in \mathcal{N}$ and $g_k \geq g_{\text{high}}$ otherwise. The total weight of the m lower-scoring clusters therefore satisfies

$$\sum_{k \in \mathcal{N}} \pi_k = \frac{\sum_{k \in \mathcal{N}} g_k}{\sum_{k \in \mathcal{N}} g_k + \sum_{k \notin \mathcal{N}} g_k} \leq \frac{m g_{\text{low}}}{m g_{\text{low}} + (K - m) g_{\text{high}}} \leq \frac{m}{K}, \tag{25}$$

where the last inequality uses $g_{\text{low}} \leq g_{\text{high}}$. For perturbations $\Delta_k = \boldsymbol{x}'_k - \boldsymbol{x}_k$ confined to $\mathcal{N}$, with $\|\Delta_k\| \leq \rho$, the triangle inequality gives

$$\|\boldsymbol{g}' - \boldsymbol{g}\| = \left\| \sum_{k \in \mathcal{N}} \pi_k \Delta_k \right\| \leq \rho \sum_{k \in \mathcal{N}} \pi_k \leq \rho \frac{m g_{\text{low}}}{m g_{\text{low}} + (K - m) g_{\text{high}}} \leq \rho \frac{m}{K}. \tag{26}$$

Uniform mean pooling has worst-case sensitivity $\rho m / K$, attained when all m perturbations equal ρ times the same unit vector. If $\rho > 0$ and $\phi_{\text{low}} < \phi_{\text{high}}$, then $g_{\text{low}} < g_{\text{high}}$, making the last inequality strict and proving Equation 24. □

C Datasets and Benchmark Construction

Evaluation rationale. Existing benchmarks examine complementary aspects of generalization. TChard evaluates unseen-peptide prediction through peptide-level hold-outs (Grazioli et al., 2022), while TITAN considers unseen TCRs and jointly unseen peptide–TCR pairs (Weber et al., 2021). EPACT uses epitope-clustered cross-validation and evaluates transfer across TCR background populations (Zhang et al., 2024); PISTE evaluates on independent external collections of TCR–peptide–HLA interactions (Feng et al., 2024). These protocols differ in their held-out entities, similarity controls, and data sources, motivating a common evaluation that examines these dimensions together.

Building on these settings, our benchmark combines peptide-only, TCR-only, and joint shifts within one triplet prediction task. This separates generalization to new peptides from generalization to new receptors and tests the additional challenge when both change simultaneously. We apply cluster-level hold-outs to both molecular axes, extending similarity control beyond peptide-only clustering and exact entity exclusion. We further assess generalization on a separately constructed external test set.

Data sources. Monomer corpora combine peptide and TCR sequences from EPACT (Zhang et al., 2024), deepAntigen (Que et al., 2025), and the recognition collections below, together with class-I/class-II MHC sequence libraries and PISTE's HLA resource (Feng et al., 2024). Pairwise supervision uses deepAntigen and EPACT eluted-ligand positives for pMHC binding, NetMHCpan affinity records distributed with EPACT for regression, and deepAntigen positives for pTCR binding. Triplet positives are pooled from PISTE, EPACT, human VDJdb (Shugay et al., 2018), and the public TCRCube pan-peptide training collection (Culka et al., 2026). The public TCRCube test collection is reserved for external evaluation.

Dataset construction. We standardize source fields and MHC identifiers, remove records without supported MHC mappings, and deduplicate sequences or interaction tuples. For the triplet benchmark, peptides and TCR CDR3 sequences are clustered separately with MMseqs2 (Steinegger & Söding, 2017), using minimum identity 0.4 and coverage 0.7 (coverage mode 1). Holding out peptide clusters, TCR clusters, or both defines peptide-only, TCR-only, and joint shifts; no separate MHC hold-out is imposed. Each axis targets 6% of source positives, preferentially selecting low-frequency clusters. The single-axis names identify the deliberately held-out axis, without requiring every entity on the other axis to occur in training.

Triplet training excludes the held-out peptide/TCR entities and those in the external test set. The final triplet training pool contains 90,169 positives. The external test set is a separately constructed evaluation collection.

For internal binding data, five negatives per positive are constructed by recombining partners within each partition, rejecting retained positive interactions and duplicate negatives. Triplet sampling draws peptide, MHC, and TCR independently, so it does not preserve pMHC pairing. Negatives represent unobserved combinations rather than experimentally confirmed non-binders. The external test set retains its supplied 205 positive and 1,002 negative labels. We provide an 80:20 label-stratified training–validation partition of the labeled triplet training pool.

Table 3 summarizes the learning corpora and evaluation partitions; triplet training counts precede the training–validation division.

Table 3: Dataset statistics. Binding counts are positive examples; monomer and affinity counts are entries and regression records, respectively. Entity columns report distinct peptides, MHC identifiers, and TCRs.

Stage	Corpus / partition	Examples	Peptides	MHCs	TCRs
Monomer	Peptide	1,350,396	1,350,396	–	–
	MHC allele–sequence	18,125	–	18,125	–
	TCR CDR3β	358,602	–	–	358,602
Pairwise	pMHC binding	341,610	235,331	144	–
	pMHC affinity	170,250	39,214	111	–
	pTCR binding	417,800	676	–	124,296
Triplet	Training pool	90,169	321	33	84,645
	Peptide-only shift	6,871	1,797	73	5,009
	TCR-only shift	6,852	208	30	6,852
	Joint shift	511	380	30	511
	External test	205	62	14	174

D BASELINES

We compare with five representative baselines, described below according to whether their original architectures support explicit MHC inputs.

TCR–peptide predictors. These methods predict recognition from peptide and TCR sequences without explicitly incorporating MHC information.

- **NetTCR-2.0** (Montemurro et al., 2021) encodes peptide and TCR sequences with BLOSUM50 and applies convolutional filters and max pooling to extract sequence features for

binding prediction. The model supports single-chain and paired TCR α/β CDR3 inputs, with the original study emphasizing the value of paired-chain information.

MHC-aware TCR–pMHC predictors. These methods support MHC information as categorical allele features or sequence representations alongside peptide and TCR inputs.

- **ERGO-II** (Springer et al., 2021) encodes TCR CDR3 sequences using LSTMs or pretrained autoencoders and encodes peptides using an LSTM, followed by an MLP binding classifier. Its modular architecture optionally incorporates TCR α, V/J genes, and MHC typing; MHC alleles are represented by learned categorical embeddings.
- **EPIC-TRACE** (Korpela et al., 2023) combines ProtBERT embeddings and one-hot sequence encodings with convolution and multi-head attention for TCR binding prediction. It supports TCR α/β chains and incorporates V/J genes and MHC alleles as categorical features, with a focus on generalization to unseen epitopes.
- **PISTE** (Feng et al., 2024) uses a physics-inspired sliding-attention Transformer to model TCR–peptide–HLA binding. It first models peptide–HLA interactions and then TCR–pHLA interactions, followed by alignment-based pooling onto the HLA pseudo-sequence for binding prediction.
- **EPACT** (Zhang et al., 2024) targets paired TCR–pMHC recognition by combining pretrained TCR and pMHC representations through epitope-anchored contrastive transfer learning. It uses pMHC representations as contrastive anchors to bring binding TCRs closer and separate non-binding TCRs in a shared embedding space.

E Experimental Details

E.1 Software and Hardware

We implement HATIS using PyTorch and PyTorch Geometric (PyG), with RDKit for molecular graph construction and Accelerate for distributed training. We use Python 3.10.18, PyTorch 2.4.1 with CUDA 12.1, PyG 2.6.1, RDKit 2025.03.6, and Accelerate 1.10.1. The GPU server is equipped with four NVIDIA Tesla V100 GPUs, each with 32 GB of memory.

E.2 Implementation Details

Input features and graph construction. For $s \in \{p, m, t\}$, RDKit converts each sequence into $G^s = (\mathcal{V}^s, \mathcal{E}^s)$ with heavy-atom nodes and bidirectional covalent edges; MHC alleles are first mapped to amino-acid sequences. The encoder inputs are

$$\begin{aligned} \boldsymbol{x}_u^s &= \mathrm{AtomFeatures}(u) \in \mathbb{R}^{45}, \qquad \boldsymbol{e}_{uv}^s = \mathrm{BondFeatures}(u, v) \in \mathbb{R}^{13}, \\ \mathbf{h}_u^{s,(0)} &= \left[\mathrm{Linear}_{45\to 248}(\boldsymbol{x}_u^s) \| \, \mathrm{Linear}_{32\to 8}(\mathrm{BN}(\mathbf{r}_u^s))\right] \in \mathbb{R}^{256}. \end{aligned} \tag{27}$$

Here, $\|$ denotes concatenation, BN denotes batch normalization, and $\mathbf{r}_u^s$ contains the return probabilities of 1–32-step random walks. Atom features encode element, charge, hybridization, aromaticity, degree, hydrogen count, ring information, backbone roles, and donor/acceptor status. Bond features encode type, ring membership, conjugation, stereochemistry, and amide/backbone status. Categorical attributes are one-hot encoded and concatenated with scalar and binary attributes; atom-to-residue indices are retained for pretraining.

Monomer pretraining. *Masked residue prediction.* For each molecule, we sample two residue sets, disjoint when enough residues are available, to construct complementary masked views. At each selected residue, we delete its side-chain atoms and incident bonds while retaining the backbone. This removes the side-chain pattern that would otherwise allow the encoder to identify the amino acid directly. To further reduce identity leakage, we replace degree and hydrogen-count features at the masked C_α and its adjacent backbone N/C atoms with unknown categories, and recompute random-walk positional encodings on the masked graph. A classification head predicts the original amino-acid identity from each masked C_α representation; reconstruction cross-entropy is averaged over masked sites within each view and then over the two views. The original sequence supplies targets and is not fed to the encoder.

Graph contrastive learning. Both views pass through the same encoder, followed by mean pooling over atom representations and a shared two-layer projection head. The projected vectors are ℓ_2-normalized. The two views of the same molecule form a positive pair, while other molecules in the minibatch provide cross-view negatives. We compute cosine-similarity logits with temperature 0.2 and average InfoNCE in both view-to-view directions. This encourages graph representations to remain consistent under different side-chain masks. The contrastive term has weight 0.2 relative to masked residue prediction.

Pairwise training losses. The pMHC and peptide–TCR objectives are

$$\begin{aligned} \mathcal{L}_{pm} &= \mathcal{U}(\mathcal{L}^{pm}_{\text{bind}}, \mathcal{L}^{pm}_{\text{aff}}) + \lambda_{\text{pool}}\mathcal{L}^{pm}_{\text{pool}}, \\ \mathcal{L}_{pt} &= \mathcal{L}^{pt}_{\text{bind}} + \lambda_{\text{pool}}\mathcal{L}^{pt}_{\text{pool}}. \end{aligned} \tag{28}$$

Here, $\mathcal{L}^{AB}_{\text{bind}}$ is the IASL-weighted binary cross-entropy from equation 7, applied to the corresponding pairwise predictor, and $\mathcal{L}^{pm}_{\text{aff}} = n_a^{-1}\sum_i(\widehat{a}_i - a_i)^2$ is the affinity regression loss over n_a labeled pairs, with target a_i and prediction $\widehat{a}_i$. The two pMHC heads share the encoder and use minibatches from their respective labeled corpora. The operator $\mathcal{U}$ adaptively balances the two tasks using learnable uncertainty weights and their regularization terms, following the uncertainty-based multi-task weighting approach of Kendall et al. (2018). $\mathcal{L}_{\text{pool}}$ averages DiffPool link-reconstruction and assignment-entropy terms, with $\lambda_{\text{pool}} = 0.5$.

Triadic binding loss. The triadic binding objective is

$$\mathcal{L}^{pmt}_{\text{bind}} = \sum_{i=1}^{n_b} w_i^* \,\text{BCE}(f_\theta(p_i, m_i, t_i), y_i)\,. \tag{29}$$

The pairwise models are frozen while the fusion module and triadic predictor are updated. IASL alternates between optimizing sample weights with fixed representations and updating the predictor with fixed weights, using uniform weights during warm-up.

Hyperparameter settings. The architecture and optimization parameters are set as follows:

- Encoder hidden dimension: 256; output dimension: 128.
- Encoder depth: 12 layers for peptide and eight layers for MHC and TCR.
- Interaction-module depth: eight layers; attention heads: eight.
- Feed-forward expansion factor: four; dropout: 0.1.
- Cross-edge selection: $k_{pm} = k_{pt} = k_{\text{tri}} = 4$.
- DiffPool cluster counts: $64 \to 8$; hidden dimension: 64.
- Pairwise DiffPool loss coefficient: $\lambda_{\text{pool}} = 0.5$.
- Monomer masking ratio: 0.2 for peptide and 0.3 for TCR.
- Monomer contrastive loss coefficient: $\lambda = 0.2$; temperature: 0.2.
- Optimizer: AdamW for monomer and pairwise training; Adam for triadic training.
- Learning rate: 10^{-5} for monomer, 2×10^{-5} for pMHC, 10^{-5} for peptide–TCR, and 10^{-5} for triadic training.
- Weight decay: 10^{-2} for monomer, 2×10^{-3} for both pairwise and triadic training.
- Configured batch size: 2,000 for peptide/TCR monomer, 512 for pMHC, 1,024 for peptide–TCR, and 192 globally for triadic training (48 per GPU across four GPUs). Triadic evaluation uses batch size 256.
- Gradient accumulation: two steps for peptide–TCR training.
- Training budget: up to 50 epochs for peptide/TCR monomer, 20 for pMHC, 50 for peptide–TCR, and 12 for triadic training.
- Triadic schedule: cosine decay from 10^{-5} to 10^{-7} over five scheduler steps, taken at epoch ends; the rate reaches 10^{-7} at the end of epoch 5 and remains fixed for epochs 6–12.

Table 4: Training data used by the baselines and HATIS. A checkmark indicates use of the corresponding dataset.

Model	Peptide monomer	MHC monomer	TCR monomer	pMHC affinity	pMHC binding	pTCR binding	Triplet binding
NetTCR-2.0							✓
ERGO-II							✓
EPIC-TRACE							✓
PISTE							✓
EPACT	✓		✓	✓			✓
HATIS	✓	✓	✓	✓	✓	✓	✓

- IASL sample-weight regularization coefficient: $\lambda_w = 2 \times 10^{-3}$ for both pairwise and triadic training. The implementation sets `lambdap`=2 in $1000J/\texttt{lambdap}+\|\mathbf{w}\|_2^2$, which is equivalent up to positive scaling to $J + \lambda_w\|\mathbf{w}\|_2^2$.
- IASL composition weighting: $\alpha = 1$ at both stages; $\delta = 0$ at both stages.
- RBF bandwidth: an adaptive median heuristic is applied separately to each cluster. The squared bandwidth is the median squared Euclidean distance between distinct sample indices in the sample-scaled cluster features, with bandwidth multiplier one and a numerical floor of 10^{-8} on the median squared distance.
- IASL inner steps: 50 for both pairwise and triadic training.
- IASL inner learning rate: 0.02 for both pairwise and triadic training.
- IASL warm-up: ten epochs for peptide–TCR; zero for pMHC and triadic training.
- Triadic IASL uses the local 48-example minibatch, with the global history buffer disabled.
- Triadic interface gate: normalized interface scores, positive slope initialized to 0.05, and bias initialized to zero.
- Triadic gradient clipping threshold: 5; mixed precision: FP16; random seed: 42; early stopping disabled. Structure supervision is disabled.

E.3 Baseline Implementation

We build the baseline experiments on the implementations released by their authors, with data interfaces adapted to our benchmark. The implementation sources are listed below:

- **NetTCR-2.0:** `https://github.com/mnielLab/NetTCR-2.0`.
- **ERGO-II:** `https://github.com/IdoSpringer/ERGO-II`.
- **EPIC-TRACE:** `https://github.com/DaniTheOrange/EPIC-TRACE`.
- **PISTE:** `https://github.com/Armilius/PISTE`; the released Code Ocean package is available at `https://doi.org/10.24433/CO.3216167.v2`.
- **EPACT:** `https://github.com/zhangyumeng1sjtu/EPACT`.

Table 4 summarizes the datasets used by the baselines and HATIS.

The main baseline settings are summarized below:

- **NetTCR-2.0:** convolutional kernel sizes $\{1, 3, 5, 7, 9\}$; 16 filters per kernel size; dense hidden dimension 32; Adam learning rate 10^{-3}; batch size 64; at most 100 epochs; early stopping after 10 epochs without improvement in validation binary cross-entropy.
- **ERGO-II:** LSTM hidden dimension 500; amino-acid embedding dimension 10; categorical embedding dimension 50; learning rate 10^{-3}; batch size 512; dropout 0.1.
- **EPIC-TRACE:** learning rate 10^{-3}; batch size 1,024; convolutional dropout 0.2; attention dropout 0.2.
- **PISTE:** embedding dimension 64; feed-forward dimension 512; nine attention heads; three encoder layers; Adam learning rate 10^{-3}; batch size 1,024; 200 training epochs.

- **EPACT:** triplet learning rate 2.5×10^{-4}; batch size 100; weight decay 10^{-2}; contrastive temperature 0.5; contrastive loss coefficient 0.3.

E.4 Representation geometry and transfer probes

These probes use the graph representations before the binding head, rather than atom-to-cluster assignments inside DiffPool. The models share a reference subset drawn from the triplet training pool by label-stratified sampling. For each model, each embedding coordinate is standardized using its mean and standard deviation on this reference subset; the same transformation is applied to test embeddings. Distances are Euclidean in the standardized, original-dimensional representation space; a two-dimensional visualization is not used to compute the metrics. Let $\mathcal{R}$ contain n standardized training reference vectors, and write $\mathcal{N}_k(z; \mathcal{R})$ for the indices of the $k = 10$ nearest reference vectors to a test representation z. Let $\mathcal{D}_{\text{test}}$ denote the set of test representations.

Neighborhood displacement (Gap). Define $d_k(z, \mathcal{R}) = k^{-1} \sum_{j \in \mathcal{N}_k(z;\mathcal{R})} \|z - z_j\|_2$ and

$$\text{Gap} = \frac{\text{median}_{z \in \mathcal{D}_{\text{test}}} d_k(z, \mathcal{R})}{n^{-1} \sum_{i=1}^{n} d_k(z_i, \mathcal{R} \setminus \{z_i\})}. \tag{30}$$

The denominator excludes each training point's self-match, and is bounded below by 10^{-8} numerically. Gap is compared across splits within each model.

Binding-label silhouette. Within each test split, let n_+ and n_- denote the numbers of positive and negative examples. We sample $\min(n_+, n_-)$ examples from each class without replacement, giving $n_{\text{bal}} = 2 \min(n_+, n_-)$ balanced examples. For sampled point i, let a_i be its mean distance to other points of the same class and b_i its mean distance to points of the other class. The reported coefficient is

$$\text{Sil} = \frac{1}{n_{\text{bal}}} \sum_i \frac{b_i - a_i}{\max(a_i, b_i)}. \tag{31}$$

The classes are the observed binding labels, not labels obtained by fitting a clustering algorithm.

Centroid transfer. Let μ_+ and μ_- be the positive and negative centroids of $\mathcal{R}$. The centroid score is

$$s_C(z) = \|z - \mu_-\|_2 - \|z - \mu_+\|_2. \tag{32}$$

Its test AUROC defines C_{AUROC}. The reference centroids use training examples only; test labels are used to evaluate rankings.

F Additional Experiments

F.1 Comparison under Shared Auxiliary Supervision

We evaluate HATIS under shared auxiliary supervision by augmenting EPACT with the pMHC binding dataset used by HATIS. We choose EPACT because its hierarchical training pipeline most closely resembles ours; the other baselines would require additional training stages to incorporate these data. Binding classification and affinity regression jointly train EPACT's pMHC encoder before triplet training. Since EPACT has no pTCR pretraining branch, we use the existing HATIS w/o pTCR transfer variant as the primary comparison.

Table 5: Data access in the four-model comparison. A checkmark indicates use of the corresponding dataset. The two primary comparison methods are highlighted in bold.

Model	Peptide monomer	MHC monomer	TCR monomer	pMHC affinity	pMHC binding	pTCR binding	Triplet binding
EPACT	✓		✓	✓			✓
EPACT + binding	✓		✓	✓	✓		✓
HATIS w/o pTCR transfer	✓	✓	✓	✓	✓		✓
HATIS	✓	✓	✓	✓	✓	✓	✓

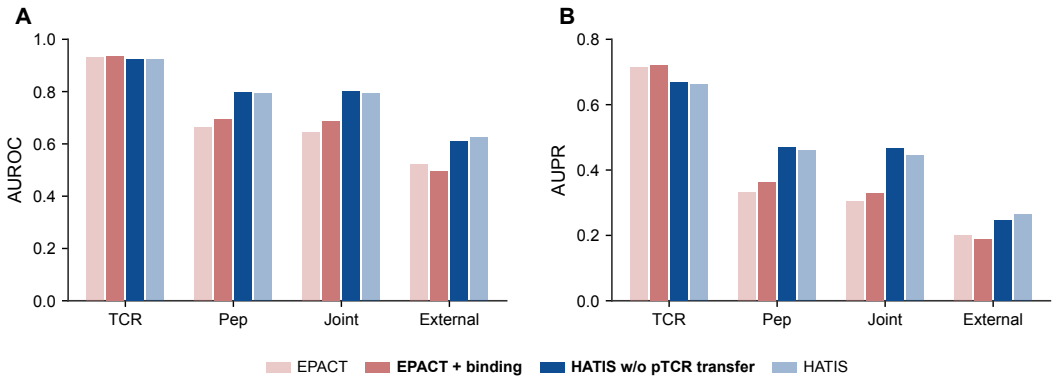


Figure 4: Comparison under shared auxiliary supervision. Panels A and B report AUROC and AUPR, respectively. TCR, Pep, Joint, and External denote TCR-only shift, peptide-only shift, joint shift, and the external test set. Opaque bars highlight EPACT + binding and HATIS w/o pTCR transfer; translucent bars show the original EPACT and full HATIS for reference.

As shown in Table 5, **EPACT + binding and HATIS w/o pTCR transfer use the same auxiliary datasets except for MHC monomer pretraining**. EPACT has no standalone MHC pretraining stage, so this corpus is not included in its training; its MHC encoder is instead learned through the shared pMHC affinity and binding tasks. Figure 4 compares these variants alongside EPACT and full HATIS under the same triplet training, validation, and test partitions.

HATIS delivers substantially stronger peptide-shift generalization under shared auxiliary supervision. Without pTCR transfer, HATIS surpasses EPACT + binding by 10.82 and 13.67 AUPR percentage points under peptide-only and joint shifts, respectively, and by 5.61 points on the external test set. These gains are accompanied by higher AUROC in all three settings, demonstrating the strength of HATIS's hierarchical atomistic modeling in learning transferable interactions from shared supervision. In Figure 4, full HATIS achieves an external AUPR of 26.39%, compared with 24.51% without pTCR transfer. EPACT + binding performs best under TCR-only shifts, while HATIS excels in the peptide-shift settings targeted by our framework.

F.2 Nearest-Neighbor Representation Transfer

We further assess whether local neighborhoods in the learned representation space preserve binding information across distribution shifts. Complementing the centroid-based probe, we evaluate nearest-neighbor transfer using the same pre-classifier embeddings of HATIS, PISTE, and EPACT. We use the shared reference subset and standardization procedure described in Appendix E.4, retrieving the $k = 10$ nearest training neighbors by Euclidean distance. For a test representation z, the transfer score is

$$s_T(z) = \frac{1}{k} \sum_{j \in \mathcal{N}_k(z;\mathcal{R})} y_j, \tag{33}$$

where $\mathcal{R}$ is the training reference set and $y_j \in \{0, 1\}$ is the binding label of neighbor j. The test AUROC of s_T defines T_{AUROC}. This probe uses training labels directly, without fitting an additional prediction head; test labels are used only for evaluation.

Table 6: Nearest-neighbor transfer AUROC (T_{AUROC}, %). The best result in each column is highlighted in bold.

Model	TCR-only	Peptide-only	Joint	External test
EPACT	**92.03**	65.96	63.75	52.62
PISTE	91.07	67.28	64.03	58.25
HATIS	91.57	**74.35**	**74.12**	**58.84**

HATIS learns more transferable local binding neighborhoods under peptide shifts. As shown in Table 6, HATIS exceeds the strongest baseline by 7.07 and 10.09 percentage points under peptide-only and joint shifts, respectively. It also achieves the highest score on the external test set, while EPACT leads under TCR-only shifts. Together with the centroid-based analysis, these results show that HATIS's transfer advantage is reflected in both global class organization and local neighborhood structure.