# PIT-GCL: PROTEIN INTERACTION USING TOPOLOGICAL GRAPH CONTRASTIVE LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Protein binding prediction is central to target identification, therapeutic binder design, and large scale screening, yet remains challenging because binding depends on sequence, three dimensional geometry, and global structural organization. Recent folding models such as AlphaFold3 and Boltz-2 have substantially improved structure prediction, but their confidence outputs (pLDDT, pTM, ipTM) are not specifically designed for binary binding prediction, and dedicated structure aware predictors often require bound complex structures that are unavailable at screening scale. We introduce PIT-GCL, a dual tower structure aware framework that encodes each protein independently from its amino acid sequence, C$\alpha$ point cloud, and a global persistent homology descriptor. Each tower combines residue ESM-2 embeddings with a topological summary computed from the $H_0$ and $H_1$ persistence landscapes of a Vietoris-Rips filtration, and processes the resulting tokens with a structure aware Transformer in which pairwise C$\alpha$ distances enter as a learned attention bias. A bidirectional cross attention module then performs latent space soft docking between the two per-protein representations, and the model is trained with a combined binary cross entropy and NT-Xent contrastive objective. On three binary interaction prediction benchmarks, general PPI on PPIRef, TCR-pMHC binding on STAG, and whole chain pairs on PPB-Affinity, PIT-GCL outperforms representative sequence based, structure aware, and task specific baselines on general PPI under our evaluation, and is the only method above chance on PPB-Affinity; on TCR-pMHC it leads at a fixed decision threshold but is outranked by a task specific sequence model. Because each protein is encoded independently in the first phase, its representation can be precomputed and reused across candidate pairs, which is convenient for large scale screening.

## 1 INTRODUCTION

Predicting whether two proteins physically interact underpins target identification, therapeutic binder design and large scale screening, and is hard because binding depends on sequence, 3D geometry and global structural organization. Folding models such as AlphaFold-Multimer (Evans et al., 2022), AlphaFold3 (Abramson et al., 2024) and Boltz-2 (Passaro et al., 2025) predict complex structure rather than a calibrated binary score, and all of them co-fold the complex at inference. At minutes per pair, a screen over $10{,}000$ pairs costs on the order of $10^3$ GPU-hours, while PIT-GCL scores the same set in 17 minutes on one GPU, of which the part that scales with the number of candidates is under a minute and the rest is a one-off descriptor computation a second target reuses (Appendix G).

We propose PIT-GCL, a general purpose protein binding prediction framework that predicts binding without requiring the bound complex structure as input. PIT-GCL uses a topology aware dual tower design: each protein is encoded independently from its sequence, C$\alpha$ point cloud, and persistent homology descriptor, and pairwise interactions are modeled only after protein level representations are constructed. This formulation preserves protein specific information before interaction level reasoning and allows per-protein embeddings to be reused across candidate pairs in large scale screening.

We evaluate PIT-GCL on three binary interaction prediction tasks and they do not give the same answer, which we report as such. On general PPI (PPIRef (Bushuiev et al., 2024)) it leads every baseline we rank on all five metrics. On TCR-pMHC (STAG (Slone et al., 2025)) it leads at a fixed



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.

decision threshold but is outranked by a task-specific sequence model, so we make no state-of-the-art claim there. On PPB-Affinity (Liu et al., 2024), which applies the same label construction to whole chains rather than interface regions, it is the only method above chance, though it recovers only part of the task.

Our contributions are: a topology-aware dual-tower framework that encodes each protein independently and predicts binding through cross-protein attention, needing no bound complex; an injection of a global persistent-homology vector at the residue level before structure-aware attention, whose contribution we quantify with parameter-matched and shuffled controls; and an evaluation on three benchmarks with every baseline retrained on the benchmark it is scored on, separating where the model leads from where it does not. Code is uploaded as supplementary material.

## 2 Related Work

Methods for protein-protein binding prediction differ in the input modality they consume - sequence, individual structures, or a co-folded complex - and in what each side of a pair actually is: general PPI benchmarks such as PPIRef (Bushuiev et al., 2024) present the extracted interface region of each chain, PPB-Affinity (Liu et al., 2024) presents whole chains, and specialized settings such as TCR-pMHC fix the interaction family instead. We evaluate all three and organize prior work by input modality. PIT-GCL sits between the sequence-only and co-folding families: it takes structure from individual chains without requiring a co-folded complex, and produces per-protein representations that can be cached and reused.

**Protein encoders.** Sequence encoders range from substitution-matrix embeddings such as BLOSUM (Henikoff & Henikoff, 1992), still common in TCR-pMHC predictors, to protein language models such as ProtTrans (Elnaggar et al., 2022) and ESM-2 (Lin et al., 2023). Structure encoders operate on residue point clouds or graphs, with equivariant designs including the Geometric Vector Perceptron (Jing et al., 2021), $E(n)$-equivariant graph networks (Satorras et al., 2021) and the Equiformer (Liao & Smidt, 2023) used inside PPIFormer (Bushuiev et al., 2024). PIT-GCL pairs a frozen ESM-2 encoder with a lighter geometric strategy: a structure-aware Transformer over the C$\alpha$ point cloud plus a global topological summary of it.

**Topological descriptors for biomolecules.** Persistent homology has been applied to biomolecular property prediction since TopologyNet (Cang & Wei, 2017), which feeds persistence barcodes of protein-ligand complexes to a convolutional network, and its element-specific successor (Cang & Wei, 2018), which partitions the point cloud by atom type so that each barcode carries chemical identity. For interactions specifically, persistent spectral methods reach binding affinity through the eigenvalues of Laplacians built on the filtration rather than through barcodes alone (Wee & Xia, 2022). Those lines share a design that ours does not: the descriptor is computed on the *complex* and enriched with atom or residue attributes, which makes it interface-specific and correspondingly dependent on having the complex. PIT-GCL computes a single attribute-free landscape per protein from C$\alpha$ positions alone, which is weaker per descriptor but is what allows the two towers to be encoded and cached independently. The name is also worth separating from TopoGCL (Chen et al., 2024b), which contrasts augmented views of a graph; our contrastive term aligns two protein towers and involves no graph views.

### 2.1 Protein-Protein Interaction Prediction

**Sequence-based methods.** Sequence-based predictors learn directly from a pair of amino acid sequences. DeepDTA (Öztürk et al., 2018) stacks 1D convolutions with global max pooling and a fully connected head, and PIPR (Chen et al., 2019) uses a Siamese residual recurrent convolutional network over amino acid embeddings. Both are easy to train at scale but ignore three-dimensional geometry, which limits their ability to separate sequence-similar pairs that behave differently in 3D.

**Structure-aware methods.** D-SCRIPT (Sledzieski et al., 2021) embeds each sequence with a pretrained language model and pools a predicted inter-chain contact map into a binding probability; PPIFormer (Bushuiev et al., 2024) represents each interface as an SE(3)-equivariant point cloud, pretrains an Equiformer with masked modeling on PPIRef, and scores pairs by average log-likelihood.

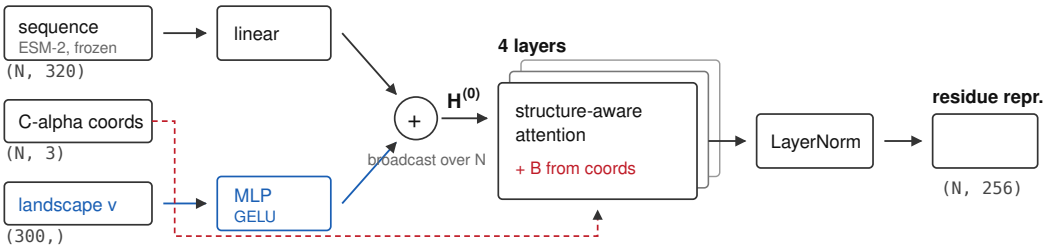



Figure 1: The per-protein encoder. A frozen ESM-2 embedding is projected to the model width, a two-layer MLP maps the persistence landscape to the same width, and the two are added so every residue token carries the protein's global topology; four structure-aware layers then attend over the residues with a bias built from the C$\alpha$ coordinates (Figure 2). How the landscape itself is built is shown in Figure 3.

Predictors that operate on bound complexes or co-folded outputs (Gainza et al., 2020; Réau et al., 2023; Evans et al., 2022) are excluded from our screening-style evaluation, which assumes only individual structures. A practical caveat of this family is calibration: PPIFormer emits an unnormalized log-likelihood rather than a probability, so a fixed $0.5$ threshold penalises it disproportionately (Appendix D).

### 2.2 TCR-pMHC Binding Prediction

Here one partner is a T-cell receptor, characterized primarily by its CDR3$\alpha$ and CDR3$\beta$ loops, and the other a peptide presented on an MHC. Most methods are sequence-level: 1D CNNs over BLOSUM-encoded CDR3 and peptide sequences (Montemurro et al., 2021), a Random Forest with an MHC-aware variant (Pham et al., 2023), interpretable attention (Chen et al., 2024a), joint attention over the TCR and a SMILES-encoded peptide (Weber et al., 2021), and contrastive objectives (Ge et al., 2025; Shuai et al., 2025; Xu et al., 2026). None reasons directly about the 3D arrangement of the CDR loops. On the structural side, STAG (Slone et al., 2025) builds a graph from the bound complex, which captures interface geometry but requires a resolved or modelled complex for every candidate pair. PIT-GCL differs from both: it encodes each protein from its individual structure and fuses the pair only through cross-attention, removing the dependence on a pre-existing complex at inference time.

## 3 Method

PIT-GCL encodes each protein of a pair independently and models their interaction only afterwards, through cross-protein attention. The decoupled design preserves protein-specific sequence, geometric and topological information before fusion and yields per-protein representations that can be cached and reused at inference.

The encoder follows the graph-Transformer paradigm, in which each node is a token and relational structure enters as an additive bias on the attention logits rather than as message passing along edges (Ying et al., 2021; Kreuzer et al., 2021; Park et al., 2022); for protein structures that bias has been instantiated with residue distances as in the pair representation of the Evoformer (Jumper et al., 2021), and we do the same at the residue level with pairwise C$\alpha$ distances. The global descriptor comes from persistent homology, which tracks topological invariants across a filtration, is invariant under rigid motions and is provably stable under small perturbations (Cohen-Steiner et al., 2007); we vectorise it as persistence landscapes (Bubenik, 2020; 2015), with a primer in Appendix H.

**Phase 1: Per-Protein Embedding.** PIT-GCL applies the same encoder to each protein of a pair $(P_A, P_B)$ independently (Figure 1). For a protein $P$ with $N$ residues we extract three modalities. First, the C$\alpha$ coordinates $X \in \mathbb{R}^{N\times 3}$, which provide residue-level 3D geometric information. Second, we obtain residue-wise sequence embeddings from a pretrained ESM-2 (Lin et al., 2023) language model, $S \in \mathbb{R}^{N\times d_{\text{esm}}}$, which capture evolutionary and semantic information from the amino acid sequence. Third, we compute a persistent-homology-based topological descriptor $v \in \mathbb{R}^{d_{\text{topo}}}$ by constructing a Vietoris-Rips filtration (Vietoris, 1927; Edelsbrunner & Harer, 2010) on the C$\alpha$

point cloud. Since computing $H_k$ requires simplices of dimension $k+1$ (Appendix H), the complex is expanded to dimension 2. The resulting $H_0$ and $H_1$ persistence diagrams are summarized as persistence landscapes (Bubenik, 2015) and concatenated into a single vector that captures the global topological organization of the protein.

We project the residue-wise ESM-2 embeddings into the model hidden dimension via a linear map, $H_{\text{seq}} = SW_{\text{esm}} \in \mathbb{R}^{N \times d_{\text{model}}}$, and map the global topological descriptor to the same dimension using a two-layer MLP with a GELU activation, $h_{\text{topo}} = \text{MLP}_{\text{topo}}(v) \in \mathbb{R}^{d_{\text{model}}}$. We then inject the global topological signal into every residue representation by broadcasting and adding it to the residue-wise sequence embedding,

$$H^{(0)} = H_{\text{seq}} + \mathbf{1}_N \, h_{\text{topo}}^{\top} \in \mathbb{R}^{N \times d_{\text{model}}},$$

so that each residue token is conditioned on a single shared summary of the protein's topology before any structural reasoning takes place.

To incorporate 3D geometric information, $H^{(0)}$ is processed by a stack of $L$ structure-aware Transformer layers in which residues serve as tokens and pairwise C$\alpha$ distances enter as additive attention biases. For residues $i$ and $j$ we compute $d_{ij} = \|X_i - X_j\|_2$, encode the resulting distance matrix with 16 radial basis functions on the interval $[0,\ 20]$ Å, and project the RBF features through a linear layer to obtain a per-head attention bias $B_{\text{struct}} \in \mathbb{R}^{H \times N \times N}$ (Figure 2, Appendix E). The structure-aware attention operation is then defined, per head, as

$$\text{Attn}(Q, K, V) = \text{softmax}\left( \frac{QK^{\top}}{\sqrt{d_h}} + B_{\text{struct}} \right) V,$$

where $d_h = d_{\text{model}}/H$; head outputs are concatenated and combined by an output projection. We use multi-head attention with $H$ heads throughout and omit head indices for clarity. Each block follows a pre-norm architecture with residual connections and a position-wise feed-forward network with GELU activation. After $L$ layers, the encoder produces residue-level representations for the two proteins, $H_A \in \mathbb{R}^{N_A \times d_{\text{model}}}$ and $H_B \in \mathbb{R}^{N_B \times d_{\text{model}}}$.

**Phase 2: Interaction Prediction.** Given the per-protein representations $H_A$ and $H_B$, PIT-GCL models their interaction with a bidirectional cross-attention module. This step can be viewed as a latent-space soft-docking operation: rather than explicitly docking the two structures in 3D, the model allows residues in one protein to attend to residues in the other.

Query, key and value projections are protein-specific rather than tied across the two towers, $Q_A = H_A W_Q^A$, $K_A = H_A W_K^A$, $V_A = H_A W_V^A$ and likewise $W_{\cdot}^B$ for $B$, which lets each tower specialize according to its role in the interaction. For $A$ attending to $B$,

$$H_{A \leftarrow B} = \text{softmax}\left( \frac{Q_A K_B^{\top}}{\sqrt{d_h}} \right) V_B, \qquad H'_A = \text{LayerNorm}(H_A + H_{A \leftarrow B}),$$

and symmetrically for $B$ attending to $A$. Both directions use multi-head attention with an output linear projection.

To form pair-level representations, we mean-pool the updated residue embeddings, $z_A = \text{MeanPool}(H'_A)$ and $z_B = \text{MeanPool}(H'_B)$, concatenate them into $Z = [z_A \,\|\, z_B] \in \mathbb{R}^{2d_{\text{model}}}$ and pass the result through a prediction head,

$$\hat{y} = \sigma\big(\text{MLP}\big(\text{LayerNorm}(Z)\big)\big),$$

where $\hat{y} \in (0, 1)$ is the predicted probability that the two proteins interact.

**Training Objective.** We combine a binary cross-entropy term $\mathcal{L}_{\text{BCE}} = -[\, y \log \hat{y} + (1-y) \log(1-\hat{y})\,]$ on the predicted probability with a contrastive term applied directly to the pooled embeddings $z_A$ and $z_B$ produced after cross-attention, without any additional projection head. Within a mini-batch of $B$ pairs we $\ell_2$-normalize the embeddings and form a similarity matrix between the $A$- and $B$-side vectors, $\text{sim}_{ij} = \langle z_A^{(i)},\, z_B^{(j)} \rangle / \tau$ with $\tau$ a temperature. For each interacting pair (label $y_i = 1$), the matched index $i$ is the only positive on its row and column; all other indices in the mini-batch act as in-batch negatives. We use a symmetric NT-Xent objective (Sohn, 2016; van den Oord et al., 2018; Chen et al., 2020) averaged over the positive pairs of the batch in both directions, and train on $\mathcal{L} = \alpha \, \mathcal{L}_{\text{NT-Xent}} + \beta \, \mathcal{L}_{\text{BCE}}$ with mixing weights $\alpha$ and $\beta$. The contrastive term aligns each protein-$A$ embedding with its matched partner relative to in-batch alternatives, complementing the BCE term at the representation level; Appendix F writes it out.

**Inference and Scalability.** Phase 1 depends on one protein only, so its residue representations can be precomputed and cached and only Phase 2 is recomputed per candidate pair. With individual structures rather than complexes as input, this is what makes the design usable at screening scale (Appendix G).

# 4 EXPERIMENTS

## 4.1 EVALUATION PROTOCOL

**Datasets.** We evaluate on three binary interaction tasks, each under its own benchmark's protocol rather than a partition of our choosing. For general PPI we use PPIRef (Bushuiev et al., 2024), taking the authors' filtered and iDist-clustered fold - their near-duplicate threshold of $0.04$ reduces $349{,}685$ filtered interfaces to $51{,}755$ cluster representatives, and this is the fold their release prescribes - which we partition $80/10/10$ per seed, with their re-pairing protocol for negatives. For TCR-pMHC we use the STAG benchmark's own train/test split (Slone et al., 2025) unchanged, carving a validation set from its training half because the benchmark provides none. We additionally evaluate on PPB-Affinity (Liu et al., 2024), a curated set of complexes with measured binding affinities. Both PPIRef and PPB-Affinity supply positive pairs only, so on both we build negatives the same way, by re-pairing a chain with one drawn from a different complex; the measured affinities are not thresholded and do not enter the label. What differs is the input: a PPIRef row is the $6$ Å interface region of each chain, a median of $24$ residues per tower, while a PPB-Affinity row is the whole chain, a median of $199$. PPB-Affinity therefore runs the same architecture and the same label construction on a roughly eight-fold larger input. It uses an $80/10/10$ partition of $10{,}484$ balanced pairs ($8{,}414/1{,}036/1{,}034$). Split statistics and the provenance of the coordinates - which are the chains of resolved complexes on all three benchmarks - are in Appendix A, and the screening throughput the design is motivated by is measured in Appendix G.

**Baselines.** On PPIRef we compare against DeepDTA (Öztürk et al., 2018), PIPR (Chen et al., 2019), D-SCRIPT (Sledzieski et al., 2021), PPIFormer (Bushuiev et al., 2024) and GIGN (Yang et al., 2023); on STAG against STAG (Slone et al., 2025), TEPCAM (Chen et al., 2024a), TRAP (Ge et al., 2025), ConTCR (Shuai et al., 2025), epiTCR and epiTCR-MHC (Pham et al., 2023), NetTCR-2.0 (Montemurro et al., 2021) and TITAN (Weber et al., 2021); on PPB-Affinity against GraphDTA (Nguyen et al., 2021), the four TCR-pMHC predictors that expose a retrainable interface, and DeepDTA, PIPR and D-SCRIPT carried over from the PPIRef comparison. Every baseline was retrained from scratch on the training split of the benchmark it is scored on, so each is fitted to the data PIT-GCL sees; PPIFormer is the single exception, used as its released PPIRef-pretrained checkpoint. Architectures, input representations and the full training protocol are in Appendix J.

**Metrics.** We report AUC, accuracy, F1, PR-AUC, and RMSE between predicted probabilities and labels, thresholding probabilities at $0.5$ for the binary metrics. Baselines that output raw scores rather than calibrated probabilities (e.g. PPIFormer, which returns log-likelihoods) are min-max normalized first. Unless a caption says otherwise, every entry in Tables 1, 2 and 11 is a mean $\pm$ standard deviation over five runs, seeds $42$–$46$; what a seed varies differs by benchmark: on PPIRef it redraws the whole $80/10/10$ partition, so the spread covers split, initialisation and negative sampling; on STAG the test split is fixed by the benchmark and the seed redraws the train/validation partition; on PPB-Affinity all three splits are fixed by the benchmark and the seed varies initialisation and batch order only. Throughout, the best result in a column is in **bold** and the runner-up is underlined. Split sizes are in Table 4 and the full hyperparameter configuration in Appendix F.

## 4.2 RESULT ANALYSIS

**PPI** Table 1 reports PPIRef results against five baselines. PIT-GCL attains the highest score on all five metrics among the methods we rank; GIGN is excluded for the reason given below. Over the strongest baseline, PIPR, it gains $0.0462$ AUC and between $0.0409$ and $0.0588$ on the other four. Both methods are averaged over the same five seeds, so these margins can be read against the seed-to-seed spread of the comparison itself: each is between $6.4$ and $9.6$ times the combined standard deviation of the two methods. That margin has two parts, and only the second is architectural. DeepDTA and PIPR learn their residue embeddings from scratch while PIT-GCL starts from a frozen

| Model Name | AUC↑ | ACC↑ | F1↑ | PR-AUC↑ | RMSE↓ |
|---|---|---|---|---|---|
| DeepDTA | $0.8691_{\pm 0.0059}$ | $0.7785_{\pm 0.0070}$ | $0.7791_{\pm 0.0042}$ | $0.8765_{\pm 0.0038}$ | $0.3865_{\pm 0.0055}$ |
| PIPR | $\underline{0.8840}_{\pm 0.0033}$ | $\underline{0.7944}_{\pm 0.0024}$ | $\underline{0.7825}_{\pm 0.0042}$ | $\underline{0.8948}_{\pm 0.0026}$ | $\underline{0.3723}_{\pm 0.0030}$ |
| D-SCRIPT | $0.6517_{\pm 0.0584}$ | $0.5409_{\pm 0.0243}$ | $0.2060_{\pm 0.1802}$ | $0.6760_{\pm 0.0553}$ | $0.5273_{\pm 0.0222}$ |
| PPIFormer | $0.8334_{\pm 0.0058}$ | $0.6546_{\pm 0.0608}$ | $0.4692_{\pm 0.1417}$ | $0.8692_{\pm 0.0051}$ | $0.4636_{\pm 0.0085}$ |
| GIGN† | $0.9986_{\pm 0.0009}$ | $0.7781_{\pm 0.1596}$ | $0.6754_{\pm 0.2718}$ | $0.9954_{\pm 0.0036}$ | $0.3426_{\pm 0.1300}$ |
| PIT-GCL (ours) | $\mathbf{0.9302}_{\pm \mathbf{0.0037}}$ | $\mathbf{0.8410}_{\pm \mathbf{0.0057}}$ | $\mathbf{0.8413}_{\pm \mathbf{0.0052}}$ | $\mathbf{0.9365}_{\pm \mathbf{0.0035}}$ | $\mathbf{0.3314}_{\pm \mathbf{0.0057}}$ |

Table 1: Performance on the PPIRef test set. A logistic regression on five chain-length features alone reaches AUC $0.8138$ on this split, so the absolute numbers include a benchmark cue no method has to learn; Appendix B measures it and Section 4.3 reports what survives when it is removed. †GIGN is reported for completeness but is not a binding-prediction result and is excluded from the ranking (Appendix B). D-SCRIPT is the eight-member ensemble its authors' pipeline produces (Appendix C).

language model, so we also report a control that keeps the language model and discards everything else: mean-pool the same cached ESM-2 embeddings, concatenate, and fit an MLP (C1 in Table 3). It reaches $0.9126 \pm 0.0032$. Of the $0.0462$ over PIPR, $0.0286$ is therefore the pretrained encoder and $0.0176$ is the architecture on top of it, the latter still $3.6$ times the combined standard deviation but a much smaller number than the headline margin.

Structure aware approaches show two distinct failure modes. PPIFormer (Bushuiev et al., 2024) stays competitive on the ranking metrics but drops sharply on the threshold-dependent ones, a calibration effect rather than poor discrimination: its F1 ranges from $0.3635$ to $0.6994$ across seeds while its AUC moves by $\pm 0.0058$. Selecting each method's threshold on that seed's validation predictions cuts our F1 margin over it from $0.3721$ to $0.1764$, while the margins over DeepDTA and PIPR barely move (Appendix D). GIGN (Yang et al., 2023) is the outlier and we flag it rather than rank by it: its AUC is the highest in the table while its accuracy and F1 vary across seeds by $\pm 0.1596$ and $\pm 0.2718$, two orders of magnitude more. A near-perfect and reproducible ordering whose operating point is close to arbitrary is not binding prediction by degree, and the likely cause is that GIGN consumes the complex interface directly and so can read the re-pairing artefact, which requires no notion of binding. We therefore exclude GIGN from the comparisons below; Appendix B gives the evidence and what it implies for the other rows, ours included.

**What survives a harder negative.** All three benchmarks supply positive pairs only, so negatives must be constructed, and re-pairing chains across complexes is the protocol PPIRef defines. That choice leaves a cue: chain lengths alone separate the resulting pairs to AUC $0.8138$ (Appendix B), because the two sides of a real interface contribute comparable numbers of residues and a uniformly re-paired negative does not. Lacking a source of labelled non-binders we made the re-pairing harder instead, pairing each chain with one of *similar* length, which collapses the cue to $0.5456$. Against those negatives the published seed-42 model scores $0.8863$, a loss of $0.0495$ where the cue lost $0.2686$, and still above the strongest baseline on the easy split. Since the model never saw this distribution in training that figure is a lower bound, and length matching bounds one cue rather than all of them. Against the strongest structure aware baseline (PPIFormer), PIT-GCL improves AUC by $0.0968$ and F1 by $0.3721$ at the fixed cut-off ($0.1764$ under calibrated thresholds, as above); against D-SCRIPT it improves AUC by $0.2785$ and F1 by $0.6353$. Section 4.3 separates the two components behind these gains, and the outcome is only partly what the design would suggest: deleting the topological descriptor costs $0.0157$ AUC, while deleting the structure-aware attention bias changes it by an amount we cannot distinguish from zero (A6z, $+0.0014$, $95\%$ CI $[-0.0011, +0.0038]$). We therefore leave the remaining gap to structure-aware baselines unattributed rather than assign it to a component we have not measured contributing.

**TCR-pMHC** We also evaluate on the STAG benchmark against eight TCR-pMHC predictors, all retrained on its training split. PIT-GCL leads on the threshold-dependent metrics - accuracy $+0.0594$ and F1 $+0.0386$ over the strongest baseline, epiTCR - and trails it on the ranking ones, by $-0.0187$ AUC and $-0.0550$ PR-AUC. We therefore make no state-of-the-art claim on TCR-pMHC: a task-specific sequence model orders these pairs better, while PIT-GCL converts its scores into decisions more effectively under the split's $76.8\%$ class imbalance. The table, the per-baseline reading and the

| Model Name | AUC↑ | ACC↑ | F1↑ | PR-AUC↑ | RMSE↓ |
|---|---|---|---|---|---|
| ConTCR | $0.5001_{\pm 0.0017}$ | $0.5004_{\pm 0.0016}$ | $0.3104_{\pm 0.2898}$ | $0.5000_{\pm 0.0026}$ | $0.5000_{\pm 0.0000}$ |
| epiTCR | $0.5314_{\pm 0.0188}$ | $0.5080_{\pm 0.0016}$ | $0.0500_{\pm 0.0075}$ | $0.5348_{\pm 0.0149}$ | $0.5121_{\pm 0.0018}$ |
| epiTCR-MHC | $\underline{0.5342_{\pm 0.0107}}$ | $\underline{0.5097_{\pm 0.0023}}$ | $0.0920_{\pm 0.0058}$ | $\underline{0.5434_{\pm 0.0088}}$ | $0.5051_{\pm 0.0009}$ |
| NetTCR-2.0 | $0.5048_{\pm 0.0278}$ | $0.5028_{\pm 0.0225}$ | $0.3341_{\pm 0.1814}$ | $0.5024_{\pm 0.0199}$ | $0.5003_{\pm 0.0016}$ |
| GraphDTA | $0.4972_{\pm 0.0165}$ | $0.4983_{\pm 0.0057}$ | $\underline{0.4642_{\pm 0.1988}}$ | $0.5030_{\pm 0.0129}$ | $0.5118_{\pm 0.0241}$ |
| DeepDTA* | $0.5054_{\pm 0.0098}$ | $0.5050_{\pm 0.0067}$ | $0.3005_{\pm 0.2644}$ | $0.5091_{\pm 0.0059}$ | $0.5816_{\pm 0.0784}$ |
| PIPR* | $0.5011_{\pm 0.0030}$ | $0.5000_{\pm 0.0000}$ | $0.2667_{\pm 0.3652}$ | $0.4984_{\pm 0.0021}$ | $0.5000_{\pm 0.0000}$ |
| D-SCRIPT* | $0.5135_{\pm 0.0074}$ | $0.5004_{\pm 0.0080}$ | $0.0665_{\pm 0.0845}$ | $0.5179_{\pm 0.0139}$ | $0.6268_{\pm 0.0078}$ |
| PIT-GCL (ours) | $\mathbf{0.6612_{\pm 0.0196}}$ | $\mathbf{0.6023_{\pm 0.0166}}$ | $\mathbf{0.5366_{\pm 0.0113}}$ | $\mathbf{0.6431_{\pm 0.0234}}$ | $0.5446_{\pm 0.0279}$ |

Table 2: Performance on PPB-Affinity. Negatives are re-paired chains from different complexes, as on PPIRef; the measured affinities are not thresholded and do not enter the label. *marks the general-purpose PPI baselines of Table 1, retrained on this benchmark's training split; the unmarked rows are models built for other interaction settings. Every baseline sits at chance either way. No RMSE is bolded: the column's lowest values belong to arms that emit a constant $0.5$, which scores $0.5000$ on a balanced split without ordering anything.

protocol are in Appendix I. The analysis below concerns PPIRef, the setting our topological claim is about.

**PPB-Affinity** Table 2 reports PPB-Affinity, which shares PPIRef's label construction and differs from it in input size. Every baseline sits at chance on the ranking metrics, with accuracy within a point and a half of $0.50$; the arms that reach a non-trivial F1 do so only by collapsing towards one class in some runs, which is what F1 standard deviations between $0.18$ and $0.37$ record against means below $0.47$. The unmarked rows are models built for other interaction settings and on their own would establish only that the benchmark resists transfer. The rows marked * remove that reading: DeepDTA, PIPR and D-SCRIPT, the general-purpose PPI baselines of Table 1, land at chance after training on this split itself, two of them by collapsing to a constant class rather than by ranking badly. PIT-GCL is the only method in the table that separates the classes at all, though far below its own PPIRef performance. The limit is therefore not transfer, and we do not attribute it to the affinity provenance of these complexes either, because the binary label does not use it. What remains is the input: interface regions against whole chains. The chain-length control that separates PPIRef's pairs to $0.8138$ reaches $0.4940$ here (Appendix B), so the cue that makes PPIRef easy is absent, which is the more parsimonious reading of the gap. We read PPB-Affinity as a harder setting rather than a solved one.

Taken together, PIT-GCL is the strongest method we evaluate on general PPI by a margin far outside seed variation, and is the strongest on threshold-dependent metrics but not on ranking metrics on TCR-pMHC, where a task-specific sequence model ranks better; on the whole-chain setting of PPB-Affinity it is the only method that leaves chance at all, on a benchmark where three sequence-based models trained on the split itself do not. The rest of this section concerns PPIRef, where the topological claim is made and tested.

### 4.3 Component Ablation

We ablate on PPIRef, the setting our topological claim concerns; STAG is a single-family task used to test transfer, not to attribute modalities. Every arm is a single edit to the published code path. A1–A5 edit the two terms of the fusion equation $H^{(0)} = H_{\text{seq}} + \mathbf{1}_N h_{\text{topo}}^{\top}$, A5 degenerating both at once as a leakage floor; A6 and A6z edit the structural attention bias; A8 sweeps the contrastive weight; A10 removes the topological descriptor and the bias together; the labels index a larger internal grid and are therefore not consecutive. Arms run at three seeds (42–44) and differences are tested with a paired bootstrap over $10{,}000$ resamples on the seed-$42$ predictions, paired on identical test rows. Table 3 gives every arm and every metric; Appendix E adds the per-arm reading.

**The topological descriptor carries protein-specific signal, not capacity.** Deleting the topology branch (A1) costs $0.0157$ AUC, but on its own that number is ambiguous: the deletion also removes $142{,}848$ parameters, so the drop is equally consistent with the model simply being short of capacity. Two controls separate the two explanations. A1z keeps the topology MLP and trains it but feeds it

| Arm | | AUC↑ | ACC↑ | F1↑ | PR-AUC↑ | RMSE↓ |
|---|---|---|---|---|---|---|
| main | full model | $\mathbf{0.9305}_{\pm 0.0048}$ | $\mathbf{0.8414}_{\pm 0.0077}$ | $\mathbf{0.8422}_{\pm 0.0071}$ | $\mathbf{0.9370}_{\pm 0.0042}$ | $\mathbf{0.3295}_{\pm 0.0067}$ |
| A1 | − topology | $0.9149_{\pm 0.0039}$ | $0.8246_{\pm 0.0061}$ | $0.8208_{\pm 0.0025}$ | $0.9248_{\pm 0.0041}$ | $0.3516_{\pm 0.0035}$ |
| A1z | topo. zeroed | $0.9165_{\pm 0.0046}$ | $0.8276_{\pm 0.0049}$ | $0.8263_{\pm 0.0046}$ | $0.9259_{\pm 0.0037}$ | $0.3460_{\pm 0.0038}$ |
| A1s | topo. shuffled | $0.9157_{\pm 0.0043}$ | $0.8266_{\pm 0.0056}$ | $0.8240_{\pm 0.0065}$ | $0.9255_{\pm 0.0040}$ | $0.3473_{\pm 0.0029}$ |
| A6 | − bias | $0.9302_{\pm 0.0040}$ | $0.8430_{\pm 0.0046}$ | $0.8427_{\pm 0.0039}$ | $0.9360_{\pm 0.0032}$ | $0.3309_{\pm 0.0034}$ |
| A6z | bias zeroed | $0.9305_{\pm 0.0060}$ | $0.8427_{\pm 0.0072}$ | $0.8392_{\pm 0.0068}$ | $0.9364_{\pm 0.0050}$ | $0.3322_{\pm 0.0076}$ |
| *Sequence arms* | | | | | | |
| A2 | seq. zeroed‡ | $0.8253_{\pm 0.0092}$ | $0.7267_{\pm 0.0157}$ | $0.7452_{\pm 0.0098}$ | $0.8406_{\pm 0.0095}$ | $0.4184_{\pm 0.0079}$ |
| A3 | seq. → one-hot | $0.9058_{\pm 0.0114}$ | $0.8178_{\pm 0.0133}$ | $0.8128_{\pm 0.0130}$ | $0.9179_{\pm 0.0088}$ | $0.3578_{\pm 0.0096}$ |
| A4 | seq. → ESM-35M | $0.9289_{\pm 0.0028}$ | $0.8396_{\pm 0.0045}$ | $0.8398_{\pm 0.0014}$ | $0.9353_{\pm 0.0022}$ | $0.3339_{\pm 0.0030}$ |
| A5 | floor§ | $0.5401_{\pm 0.0040}$ | $0.5000_{\pm 0.0000}$ | $0.6667_{\pm 0.0000}$ | $0.5234_{\pm 0.0038}$ | $0.6153_{\pm 0.0659}$ |
| A8a | − NT-Xent | $0.9267_{\pm 0.0032}$ | $0.8376_{\pm 0.0039}$ | $0.8367_{\pm 0.0062}$ | $0.9319_{\pm 0.0025}$ | $0.3344_{\pm 0.0048}$ |
| A8b | $\alpha$=0.25 | $0.9299_{\pm 0.0039}$ | $0.8435_{\pm 0.0048}$ | $0.8423_{\pm 0.0063}$ | $0.9354_{\pm 0.0039}$ | $0.3352_{\pm 0.0067}$ |
| A10 | − topo, − bias | $0.9152_{\pm 0.0050}$ | $0.8245_{\pm 0.0080}$ | $0.8231_{\pm 0.0080}$ | $0.9253_{\pm 0.0039}$ | $0.3501_{\pm 0.0018}$ |
| C1† | ESM mean-pool | $0.9126_{\pm 0.0032}$ | $0.8241_{\pm 0.0040}$ | $0.8202_{\pm 0.0049}$ | $0.9228_{\pm 0.0030}$ | $0.3476_{\pm 0.0036}$ |

Table 3: Ablation on the PPIRef test set, seeds 42–44, on the $H_0$+$H_1$ configuration of Table 1. Bold marks the full model as the reference, not a winner. Each A arm is one flag on the released code path. †C1 is not: the same cached embeddings mean-pooled into an MLP, untuned, over five seeds. ‡A2 and §A5 are not read at face value. Appendix E explains all three.

a zero vector, so the branch reduces to a single learned bias added to every residue of every protein: capacity intact, protein-specific information gone. A1s changes only the data served to that branch, giving each protein another protein's landscape, so parameter count, input distribution, descriptor statistics and the gradient path are unchanged and only the correspondence between a protein and its topology is destroyed. A1z recovers $0.0016$ of the loss, so capacity accounts for about a tenth of A1's total and protein-specific topological signal for the remaining $0.0140$. A1s agrees, the two controls landing within $0.0013$ of each other on the seed-42 partition (both $p < 10^{-4}$): a wrong descriptor is worth about what a constant one is worth, which is what a branch read for protein-specific content predicts.

**The sequence arms.** Replacing the embedding with a parameter-free sinusoidal encoding of residue index ($\mathrm{A2}_{H_0}$) removes amino acid identity while keeping residues distinguishable, and the model still reaches $0.9057$ AUC, above every baseline we rank. Read against control C1 below, which keeps only the sequence pathway and reaches $0.9126$, this says that either pathway alone gets most of the way here, not that the structural one is privileged. One-hot identity (A3) costs a paired $-0.0216$ ($p < 10^{-4}$), so residue identity does not substitute for the pretrained representation. Scaling the encoder four-fold slightly hurts instead of helping, a paired $-0.0037$ ($p = 0.025$), so the result does not come from the size of the language model and we keep the smallest checkpoint. A2 zeroes the embedding rather than substituting for it, which as the table's ‡ note records also closes the coordinate pathway, so it is a topology-only arm and the useful complement to A1: topology without sequence (paired $-0.0999$) and sequence without topology (paired $-0.0166$) both fall short of the full model. The floor A5 collapses to a constant, so a model with both input channels degenerate cannot separate the pairs at all, which rules out leakage through the splits, the caches or the metric code. Appendix E reads each arm, reports the $H_0$-only versions and gives the evidence behind both notes.

**The structural attention bias does not measurably contribute.** Without these arms the gap over structure-aware baselines could only be credited to the attention bias by elimination, since the topological branch was the only component measured. A6 (bias deleted with its parameters) and A6z (branch kept and trained, its output discarded) test that directly, and neither moves AUC: $-0.0011$ ($p = 0.44$) and $+0.0014$ ($p = 0.27$), both 95% intervals straddling zero. We therefore report the gap as unattributed. What these arms remove is larger than "attention bias" suggests: the C$\alpha$ coordinates enter the residue encoder at exactly one point, so deleting it leaves the persistence landscape - computed from the same point cloud - as the only channel through which 3D structure reaches the model. That direct view of the coordinates costs nothing we can measure while the

descriptor costs $0.0157$ (A1), so the geometry this model uses is what the landscape summarises, not what the dense pairwise-distance bias supplies.

**The contrastive term contributes, but far less than the descriptor.** Arms A8a and A8b sweep $\alpha$ in $\mathcal{L} = \alpha\,\mathcal{L}_{\text{NT-Xent}} + \beta\,\mathcal{L}_{\text{BCE}}$ with everything else fixed; $\alpha = 0.5$ is the full-model row. Halving it is indistinguishable from the published setting, and removing the term costs a paired $-0.0054$ ($[-0.0079, -0.0028]$, $p < 10^{-4}$) - measurable, but about a third of what the descriptor does (A1, $0.0157$). The multi-objective loss is not what carries the result.

**What the architecture is worth.** A10 keeps the dual-tower encoder, cross-attention and the contrastive term but removes both structural inputs; control C1 removes the encoder as well, mean-pooling the same cached embeddings into an MLP. Their three rows split the $0.0179$ over C1 into two very unequal parts. The topological descriptor and the distance bias account for $0.0153$, essentially all of it and essentially all of A1's $0.0157$ on its own. The entire attention stack on top of that - four structure-aware layers, cross-attention soft docking and the NT-Xent term - accounts for a further $0.0026$, which is $0.4$ times the combined standard deviation of the two arms and not a difference we can measure. On this benchmark the descriptor is doing the work the architecture is credited with, and A6, A6z and A8 say the same thing component by component. C1 is untuned, so $0.0179$ bounds the architectural gain from above rather than estimating it.

## 5 Conclusion

We presented PIT-GCL, a dual-tower framework that encodes each protein independently from its sequence, C$\alpha$ point cloud and persistent-homology descriptor, and models the interaction through bidirectional cross-attention rather than joint encoding of a bound complex. It leads every baseline we evaluate on general PPI, leads at a fixed threshold on TCR-pMHC while being outranked by a task-specific sequence model, and is the only method above chance on PPB-Affinity; the dual-tower design makes per-protein representations reusable across candidate pairs. Two results qualify the picture: the component analysis is run on general PPI only, and the attention machinery around the descriptor does not contribute measurably, so the margin over structure-aware baselines is left unattributed.

## 6 Limitations

**Unbound structures as a binding prior.** PIT-GCL consumes each protein's individual structure, so the geometry it sees is an imperfect proxy for the bound interface in strongly induced-fit systems. The choice is deliberate, since at screening scale the bound complex is exactly what is unavailable, but the expected cost is degradation on flexible complexes relative to near-rigid ones, and we do not stratify by conformational change.

**The architecture beyond the descriptor.** Section 4.3 leaves the attention stack unsupported as a contribution on PPIRef. The version those arms point to is a simplified model - frozen embeddings, the topological descriptor, a small head - which we have not trained, and whether the same holds on TCR-pMHC and on the whole-chain pairs of PPB-Affinity, where the descriptor behaves differently, is open. The contrastive term is also tested in a narrow regime: only interacting pairs serve as anchors, so the labelled negatives never enter it and batches of $8$ to $16$ leave a handful of in-batch negatives per anchor.

**Scope and descriptor design.** Because the descriptor summarizes the whole point cloud rather than the interface, how much of the benefit is interface-specific rather than fold-level is unresolved. It is also fixed, non-differentiable and tied to particular choices (Vietoris-Rips, dimensions $0$ and $1$, landscape vectorization), so it cannot adapt to the task. Antibody-antigen and protein-peptide settings would help characterize where the dual-tower formulation generalizes.

## AI USE STATEMENT

Generative AI tools were used in preparing this work, and we disclose their role as follows. Coding assistance: an AI assistant was used to write and debug experiment infrastructure - feature-cache builders, seed drivers, aggregation and significance-testing scripts - and to implement the ablation switches. All such code was reviewed by the authors, and the numerical equivalence of the batched forward path against the per-pair reference implementation is verified by a test included in the released code. Writing assistance: an AI assistant was used to draft and revise portions of the experimental sections. Every number reported in this paper was produced by the runs described in Section 4 and checked by the authors against the run artifacts; no result, table entry or citation was generated by an AI tool. The research questions, experimental design, and the interpretation of the results are the authors' own. We take responsibility for the final content of this work.

## ETHICS STATEMENT

PIT-GCL predicts whether two proteins bind. The intended use is to prioritize candidates for experimental validation in target identification and binder design, where cheap in-silico triage reduces wet-lab cost. The same capability is dual-use: a method that ranks candidate binders cheaply could in principle be used to search for binders to targets of concern, and the screening throughput that motivates this work is precisely what makes such a search less costly. We consider the marginal risk small relative to already available tools - PIT-GCL predicts a binary interaction label, not a structure, a sequence, or a designed binder, and it has no generative component - but it is not zero, and it grows if such predictors are placed inside a design loop. We therefore report the method and evaluation protocol rather than releasing tuned checkpoints for sensitive target families, and we encourage downstream users who couple binding predictors to generative binder design to adopt the screening practices developed for that setting. Beyond misuse, the practical risk is over-reliance: our model is trained on the structural and interaction distributions of PPIRef, STAG and PPB-Affinity, its predictions are not calibrated probabilities of physical binding, and it should inform experimental prioritization rather than substitute for it.

## REPRODUCIBILITY STATEMENT

The code is uploaded as supplementary material. It is PIT-GCL alone: the architecture, the three dataset loaders, the training and evaluation entry points, the parallel feature-cache builders for all three benchmarks, the per-seed drivers, and the scripts that aggregate those seeds into our row of Tables 1, 2 and 11. Baseline code, the ablation switches of Section 4.3, the paired-bootstrap script and the threshold analysis of Appendix D are kept out of that package so that it runs one configuration and no flags; all are available on request. Section 4 gives the splits, seeds and protocol for each benchmark; Appendix A gives the per-split pair counts and Appendix F the full hyper-parameter configuration.

Four details are easy to get wrong when re-running from source. Three are ours and are documented in the code: the persistence descriptor requires the Rips complex to be expanded one dimension above the homology dimension being computed; the per-protein feature cache key carries no topological parameters, so a cache built for one descriptor configuration must not be reused for another; and ESM-2 accepts at most $1{,}022$ residues, so we truncate both the embedding and the C$\alpha$ coordinates there to keep the two aligned, which on PPB-Affinity affects $144$ of $10{,}484$ pairs and 2 of the $1{,}034$ test pairs (Appendix F). The fourth is in a baseline: D-SCRIPT's contact module allocates a positional-weight buffer for sequences of at most $2{,}000$ residues and then silently truncates, which fails on PPB-Affinity chains of up to $3{,}302$ residues, so reproducing its row in Table 2 requires enlarging that buffer. The change leaves the computation identical for any chain within the original limit, so it does not affect D-SCRIPT's PPIRef numbers.

## REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvu-

nakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630:493–500, 2024. doi: 10.1038/s41586-024-07487-w. URL https://doi.org/10.1038/s41586-024-07487-w.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(1):77–102, 2015.

Peter Bubenik. *The Persistence Landscape and Some of Its Properties*, pp. 97–117. Springer International Publishing, 2020. ISBN 9783030434083. doi: 10.1007/978-3-030-43408-3_4. URL http://dx.doi.org/10.1007/978-3-030-43408-3_4.

Anton Bushuiev, Roman Bushuiev, Petr Kouba, Anatolii Filkin, Marketa Gabrielova, Michal Gabriel, Jiri Sedlar, Tomas Pluskal, Jiri Damborsky, Stanislav Mazurenko, and Josef Sivic. Learning to design protein-protein interactions with enhanced generalization. In *The Twelfth International Conference on Learning Representations*, 2024.

Zixuan Cang and Guo-Wei Wei. TopologyNet: Topology based deep convolutional and multitask neural networks for biomolecular property predictions. *PLoS Computational Biology*, 13(7): e1005690, 2017. doi: 10.1371/journal.pcbi.1005690.

Zixuan Cang and Guo-Wei Wei. Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. *International Journal for Numerical Methods in Biomedical Engineering*, 34(2):e2914, 2018. doi: 10.1002/cnm.2914.

J. Chen, B. Zhao, S. Lin, H. Sun, X. Mao, M. Wang, Y. Chu, L. Hong, D. Q. Wei, M. Li, and Y. Xiong. Tepcam: Prediction of t-cell receptor-epitope binding specificity via interpretable deep learning. *Protein Science*, 33(1):e4841, 2024a. doi: 10.1002/pro.4841. URL https://doi.org/10.1002/pro.4841.

Muhao Chen, Chelsea J T Ju, Guangyu Zhou, Xuelu Chen, Tianran Zhang, Kai-Wei Chang, Carlo Zaniolo, and Wei Wang. Multifaceted protein–protein interaction prediction based on siamese residual rcnn. *Bioinformatics*, 35(14):i305–i314, 07 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz328. URL https://doi.org/10.1093/bioinformatics/btz328.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1597–1607, 2020.

Yuzhou Chen, Jose Frias, and Yulia R. Gel. TopoGCL: Topological graph contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11453–11461, 2024b. doi: 10.1609/aaai.v38i10.29026.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007. doi: 10.1007/s00454-006-1276-5.

Herbert Edelsbrunner and John L. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010. ISBN 978-0-8218-4925-5.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7112–7127, 2022. doi: 10.1109/TPAMI.2021.3095381.

Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew W. Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, Olaf Ronneberger, Sebastian Bodenstein, Michal Zielinski, Alex Bridgland, Anna Potapenko, Andrew Cowie, Kathryn Tunyasuvunakool, Rishub Jain, Ellen Clancy, Pushmeet Kohli, John Jumper, and Demis Hassabis. Protein complex prediction with AlphaFold-Multimer. *bioRxiv*, 2022. doi: 10.1101/2021.10.04.463034.

P. Gainza, F. Sverrisson, F. Monti, E. Rodolà, D. Boscaini, M. M. Bronstein, and B. E. Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17:184–192, 2020. doi: 10.1038/s41592-019-0666-6.

J. Ge, J. Wang, Q. Ye, L. Pan, Y. Kang, C. Shen, Y. Deng, C. Y. Hsieh, and T. Hou. Trap: a contrastive learning-enhanced framework for robust tcr-pmhc binding prediction with improved generalizability. *Chemical Science*, 16(22):9881–9894, 2025. doi: 10.1039/d4sc08141b. URL https://doi.org/10.1039/d4sc08141b.

Steven Henikoff and Jorja G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992. doi: 10.1073/pnas.89.22.10915.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael J. L. Townshend, and Ron O. Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations (ICLR)*, 2021.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021. doi: 10.1038/s41586-021-03819-3.

Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21618–21629, 2021.

Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3D atomistic graphs. In *International Conference on Learning Representations (ICLR)*, 2023.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/abs/10.1126/science.ade2574.

Huaqing Liu, Peiyi Chen, Xiaochen Zhai, et al. PPB-Affinity: Protein–protein binding affinity dataset for AI-based protein drug discovery. *Scientific Data*, 11(1):1316, 2024. doi: 10.1038/s41597-024-03997-4. URL https://www.nature.com/articles/s41597-024-03997-4.

A. Montemurro, V. Schuster, H. R. Povlsen, A. K. Bentzen, V. I. Jurtz, W. D. Chronister, S. R. Hadrup, M. Nielsen, and P. Marcatili. Nettcr-2.0 enables accurate prediction of tcr-peptide binding by using paired tcr$\alpha$ and $\beta$ sequence data. *Communications Biology*, 4:1060, 2021. doi: 10.1038/s42003-021-02610-3. URL https://doi.org/10.1038/s42003-021-02610-3.

Thin Nguyen, Hang Le, Thomas P. Quinn, Tri Nguyen, Thuc Duy Le, and Svetha Venkatesh. GraphDTA: predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, 37(8):1140–1147, 2021.

Wonpyo Park, Woonggi Chang, Donggeon Lee, Juntae Kim, and Seungwon Hwang. GRPE: Relative positional encoding for graph transformer. In *ICLR 2022 Machine Learning for Drug Discovery Workshop*, 2022.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, David Kwabi-Addo, Dominique Beaini, Tommi Jaakkola, and Regina Barzilay. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, 2025. doi: 10.1101/2025.06.14.659707. URL https://www.biorxiv.org/content/early/2025/06/18/2025.06.14.659707.

My-Diem Nguyen Pham, Thanh-Nhan Nguyen, Le Son Tran, Que-Tran Bui Nguyen, Thien-Phuc Hoang Nguyen, Thi Mong Quynh Pham, Hoai-Nghia Nguyen, Hoa Giang, Minh-Duy Phan, and Vy Nguyen. epitcr: a highly sensitive predictor for tcr–peptide binding. *Bioinformatics*, 39(5):btad284, 05 2023. ISSN 1367-4811. doi: 10.1093/bioinformatics/btad284. URL https://doi.org/10.1093/bioinformatics/btad284.

Manon Réau, Nicolas Renaud, Li C. Xue, and Alexandre M. J. J. Bonvin. Deeprank-gnn: a graph neural network framework to learn patterns in protein–protein interfaces. *Bioinformatics*, 39(1): btac759, 2023. doi: 10.1093/bioinformatics/btac759.

Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *International Conference on Machine Learning (ICML)*, 2021.

Yi Shuai, Pengcheng Shen, and Xianrui Zhang. Multi-positive contrastive learning-based cross-attention model for t cell receptor–antigen binding prediction. *Computer Methods and Programs in Biomedicine*, 268:108797, 2025. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2025.108797. URL https://www.sciencedirect.com/science/article/pii/S0169260725002147.

Samuel Sledzieski, Rohit Singh, Lenore Cowen, and Bonnie Berger. D-script translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein-protein interactions. *Cell Systems*, 12(10):969–982.e6, 2021. ISSN 2405-4712. doi: https://doi.org/10.1016/j.cels.2021.08.010. URL https://www.sciencedirect.com/science/article/pii/S2405471221003331.

J. K. Slone, A. Conev, M. M. Rigo, A. Reuben, and L. E. Kavraki. Tcr-pmhc binding specificity prediction from structure using graph neural networks. *IEEE Transactions on Computational Biology and Bioinformatics*, 22(1):171–179, 2025. doi: 10.1109/TCBBIO.2024.3504235. URL https://doi.org/10.1109/TCBBIO.2024.3504235.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Leopold Vietoris. Über den höheren zusammenhang kompakter räume und eine klasse von zusammenhangstreuen abbildungen. *Mathematische Annalen*, 97(1):454–472, 1927. doi: 10.1007/BF01447877.

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

JunJie Wee and Kelin Xia. Persistent spectral based ensemble learning (PerSpect-EL) for protein-protein binding affinity prediction. *Briefings in Bioinformatics*, 23(2):bbac024, 2022. doi: 10.1093/bib/bbac024.

Minrui Xu, Manman Lu, Peng Liu, Siwen Zhang, Lanming Chen, Qi Liu, Yong Lin, and Lu Xie. Protcr: a protein language model-driven framework for decoding tcr-antigen recognition toward precision immunotherapies. *Briefings in Bioinformatics*, 27(1):bbaf716, 01 2026. ISSN 1477-4054. doi: 10.1093/bib/bbaf716. URL https://doi.org/10.1093/bib/bbaf716.

Ziduo Yang, Weihe Zhong, Qiujie Lv, Tiejun Dong, and Calvin Yu-Chian Chen. Geometric interaction graph neural network for predicting protein-ligand binding affinities from 3d structures (gign). *The Journal of Physical Chemistry Letters*, 14(8):2020–2033, 2023. doi: 10.1021/acs.jpclett.2c03906.

| Dataset | Train | | Val | | Test | |
|---|---|---|---|---|---|---|
| | Pos | Neg | Pos | Neg | Pos | Neg |
| STAG (Slone et al., 2025) | 5,906 | 19,497 | - | - | 1,476 | 4,874 |
| PPIRef (Bushuiev et al., 2024) | 40,868 | 40,868 | 5,099 | 5,099 | 5,121 | 5,121 |
| PPB-Affinity (Liu et al., 2024) | 4,207 | 4,207 | 518 | 518 | 517 | 517 |

Table 4: Dataset statistics for each task. STAG is released as a two-way split, so its validation fold is the $10\%$ of the train column that each seed holds out rather than a separate fold; PPIRef and PPB-Affinity both use an $80/10/10$ partition with held-out validation and test folds. The PPIRef counts are the interfaces actually used: the $80/10/10$ partition of the $51{,}755$ deduplicated keys yields $41{,}404/5{,}175/5{,}176$ entries, of which $536/76/55$ are dropped because the corresponding structure file is absent from the released `ppi_6A` archive. The PPIRef test set therefore contains $10{,}242$ pairs. PPB-Affinity is partitioned $80/10/10$ over $10{,}484$ balanced pairs and, like PPIRef, carries an equal number of positives and negatives in every fold.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, volume 34, pp. 28877–28888, 2021.

Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 09 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty593. URL `https://doi.org/10.1093/bioinformatics/bty593`.

## A Dataset Details

PPIRef is a general PPI dataset that is not restricted to a specific interaction family. The STAG dataset is purpose-built for TCR-pMHC interaction. All models compared on PPB-Affinity are trained and tested on that dataset under the same $80/10/10$ protocol used for PPIRef, so no method sees a different partition.

**Input provenance.** On all three benchmarks the C$\alpha$ coordinates come from experimentally resolved structures, and on all three they are the chains of a resolved complex rather than independently determined or predicted monomers, so the conformation is the bound one. The granularity differs. A PPIRef row is the 6 Å interface region extracted from the complex, a median of $24$ residues per tower and a $90$th percentile of $46$; a PPB-Affinity row is the whole chain, median $199$ and $90$th percentile $438$; STAG supplies the chains of the TCR-pMHC complex. This bears on the screening argument of Section 1: an interface region cannot be identified without the complex that defines it, so the PPIRef numbers are obtained on an input a screening pipeline could not construct for itself. The PPB-Affinity and STAG inputs do not have that problem, but their conformations still come from the bound state. We state this rather than qualify the claim away. The architecture never encodes an assembled complex, which is the design claim; but our evaluation does not establish that it works from unbound or predicted monomer structures, and that is the experiment the claim needs.

We evaluate PIT-GCL on three binary binding prediction benchmarks. Each pair $(P_A, P_B)$ is labeled as $1$ if the two proteins are reported to interact and $0$ otherwise. STAG supplies labelled negatives with its own split and we use them unchanged. PPIRef and PPB-Affinity supply positive pairs only, so on both we construct negatives by pairing each positive with a chain drawn from a different complex, following the re-pairing used for PPIRef in Bushuiev et al. (2024). On PPB-Affinity the constructed pairs are written once to fixed train/val/test CSVs and reused by every method and every seed. Table 4 summarizes the resulting positive/negative counts in each split.

## B On GIGN's Near-Perfect Scores

Retrained on the PPIRef split, GIGN (Yang et al., 2023) reports an AUC of $0.9986 \pm 0.0009$ and a PR-AUC of $0.9954 \pm 0.0036$. We do not treat this as a binding-prediction result, for three reasons that are visible in the numbers themselves.

First, the ranking metrics and the threshold metrics disagree by two orders of magnitude in stability. Across five seeds GIGN's AUC varies by $\pm 0.0009$ while its accuracy varies by $\pm 0.1596$ and its F1 by $\pm 0.2718$, spanning $0.30$ to $0.98$. A model that has learned a graded notion of binding affinity produces scores whose ordering *and* whose scale are reproducible; a model reading a near-deterministic cue produces a perfect ordering whose absolute placement is arbitrary.

Second, the ceiling is implausible for the task. PPIRef pairs include interactions whose strength varies over orders of magnitude and interfaces whose geometry the bound structure does not fully determine; no method in the literature, including those trained on far more PPI data, approaches $0.999$. The other four baselines on this split span $0.65$–$0.88$.

Third, and most directly, the negatives invite exactly this shortcut. Following PPIFormer's protocol, a negative pair is produced by re-pairing two chains at random. The result is a complex that was never assembled, and assembling it mechanically leaves geometric traces - backbone clashes, voids, spacing no folded interface would show. GIGN takes the complex interface as its input and is, of the baselines, the one best equipped to notice this. Distinguishing a real interface from a mechanically re-paired one requires no model of binding.

This has a consequence beyond GIGN's own row, which we state plainly: every number on PPIRef, ours included, is measured against these negatives, so some share of every model's score may reflect decoy detection rather than binding prediction. What bounds our own exposure is architectural rather than empirical. PIT-GCL encodes each protein separately and never sees the assembled complex, so the inter-chain geometry that carries the re-pairing artefact - backbone clashes and implausible spacing *between* the two chains - is not part of its input at all; the only cross-chain computation is attention between two independently built residue representations. That argument limits the channel but does not measure the residue, and none of our ablation arms measures it either: the arm that removes both input terms (A5) also collapses the coordinate pathway, so its being at chance says nothing about whether the geometric artefact is readable. We therefore do not claim a bound on how much of our PPIRef score is decoy detection, and regard a cluster-disjoint split with harder negatives as the more informative evaluation and the main limitation of this benchmark.

**How much of PPIRef is available without a model.** A trivial-feature control puts a number on the question. A logistic regression on five chain-length features - the two lengths, their absolute difference, their product, and the ratio of the smaller to the larger - reaches AUC $0.8138$ on the seed-$42$ PPIRef test split, and one on the two persistence descriptors, combined as an elementwise product and an absolute difference, reaches $0.7905$. Neither contains any model of binding. The mechanism is that a $6$ Å interface region is mutually defined: the two sides of a real interface contribute comparable numbers of residues, and a re-paired negative breaks that correlation. The same length control on PPB-Affinity, whose towers see whole chains, reaches $0.4940$.

We therefore read Table 1 as follows. Of PIT-GCL's $0.9302$, $0.8138$ is reachable from chain lengths alone; our margin over that control is $0.1164$ and our margin over the strongest baseline is $0.0462$. The ranking in Table 1 is unaffected, since every method is scored on the same pairs, but the absolute numbers are not a measure of binding prediction alone, and the same caution applies to the ablation deltas of Section 4.3, which are differences on top of that floor. One detail is worth recording for anyone repeating this: a logistic regression on the two feature sets *concatenated* rather than combined scores exactly $0.5000$, because the negatives re-pair the same chains and the class marginals are identical by construction. Only interaction terms expose the cue, so a concatenation-style control would wrongly suggest there is none.

## C D-SCRIPT Configurations and Reproducibility

D-SCRIPT is the one baseline whose reported number depends heavily on how it is run, so Table 5 reports both configurations rather than only the favourable one.

Three observations follow. First, a single D-SCRIPT model does not learn this task: across five seeds it sits at $0.5158 \pm 0.0276$ AUC, indistinguishable from chance, and its F1 standard deviation ($0.2577$) is larger than its mean. Second, ensembling eight such models recovers real signal ($0.6517 \pm 0.0584$), so the ensemble is not merely averaging away noise but combining weak and unstable predictors into a usable one. Third, and the reason we report a range rather than a point, the ensemble itself

| Config. | AUC↑ | ACC↑ | F1↑ | PR-AUC↑ | RMSE↓ |
|---|---|---|---|---|---|
| Ensemble of 8 | $0.6517_{\pm 0.0584}$ | $0.5409_{\pm 0.0243}$ | $0.2060_{\pm 0.1802}$ | $0.6760_{\pm 0.0553}$ | $0.5273_{\pm 0.0222}$ |
| Single model | $0.5158_{\pm 0.0276}$ | $0.5130_{\pm 0.0223}$ | $0.3345_{\pm 0.2577}$ | $0.5571_{\pm 0.0374}$ | $0.6125_{\pm 0.0491}$ |

Table 5: D-SCRIPT on the PPIRef test set, means over five seeds. Table 1 carries the ensemble, the stronger of the two configurations.

| Model | Dataset | Threshold | Acc | F1 | Predicted 1 |
|---|---|---|---|---|---|
| PIT-GCL | PPIRef | 0.1 | 0.7598 | 0.8026 | 7343 |
| PIT-GCL | PPIRef | 0.3 | 0.8335 | 0.8466 | 5994 |
| PIT-GCL | PPIRef | 0.5 | **0.8501** | **0.8500** | 5114 |
| PIT-GCL | PPIRef | 0.7 | 0.8409 | 0.8268 | 4289 |
| PIT-GCL | PPIRef | 0.9 | 0.7976 | 0.7520 | 3238 |
| PIT-GCL | STAG | 0.1 | 0.8619 | **0.7203** | 1659 |
| PIT-GCL | STAG | 0.3 | 0.8671 | 0.7166 | 1502 |
| PIT-GCL | STAG | 0.5 | 0.8706 | 0.7146 | 1404 |
| PIT-GCL | STAG | 0.7 | 0.8732 | 0.7074 | 1275 |
| PIT-GCL | STAG | 0.9 | **0.8827** | 0.7125 | 1115 |

Table 6: PIT-GCL under different classification thresholds on PPIRef and STAG, on the seed-42 partition of each; the single-seed accuracy at $0.5$ is therefore $0.8501$ rather than the five-seed $0.8410$ of Table 1. Best value per column block in **bold**.

is unstable across seeds: individual ensembles span $0.5954$ to $0.7233$. On PPB-Affinity we report a three-member ensemble per seed, and all $15$ members fall in AUROC $0.4852$–$0.5535$.

Every D-SCRIPT figure we quote is a five-seed mean rather than a best run, as is every other number in this paper; given the spread above, a best run would credit the baseline with a value no re-run recovers.

## D Threshold Sensitivity and Calibration

Table 6 sweeps PIT-GCL's decision threshold on PPIRef and STAG. Both peak at or near the default $0.5$: on PPIRef, F1 spans $0.7520$–$0.8500$ and accuracy $0.7598$–$0.8501$ over thresholds $0.1$–$0.9$, with both maxima at $0.5$; on STAG the model is almost flat, F1 spanning only $0.7074$–$0.7203$ and accuracy $0.8619$–$0.8827$ across the same range.

Table 7 asks the sharper question the main text raises: it is not whether a model can be tuned to a good threshold, but whether the threshold chosen on validation transfers to test. Each method's threshold is selected on that seed's validation predictions and applied unchanged to test, and the oracle column reports the best achievable test threshold as an upper bound. Our F1 margin over PPIFormer falls from $0.3721$ at the fixed $0.5$ cutoff to $0.1764$ under validation-selected thresholds and $0.0845$ under the oracle, so most of that margin was calibration rather than discrimination. The margins over DeepDTA and PIPR barely move ($0.0622 \rightarrow 0.0590$ and $0.0588 \rightarrow 0.0480$), because those methods are already close to calibrated. PIT-GCL itself gains little from tuning, $0.8413$ at $0.5$ against $0.8439$ on validation-selected and $0.8458$ at the oracle, which is the calibration claim stated as a measurement.

## E Ablation Details

**Reading the sequence arms.** Table 8 reports the sequence arms on the $H_0$-only descriptor. A2$_{H_0}$ keeps residues distinguishable, so the distance-based attention bias still operates while amino acid identity is gone; it is the largest drop outside the floor. Read against it, A3$_{H_0}$ places the value of residue identity at $0.0163$ and the value of the pretrained representation beyond identity at $0.0063$. A4 scales from the 8M ESM-2 checkpoint to the $480$-dimensional 35M one for no gain on either descriptor despite four times the encoder parameters: the three-seed means fall by $0.0013$ on $H_0$-only and $0.0017$ on $H_0{+}H_1$, and on the seed-42 partition the paired differences are $-0.0019$ ($p = 0.25$) and $-0.0037$ ($p = 0.025$), the latter the figure quoted in Section 4.3. We keep the smallest checkpoint, which also keeps the per-protein encoding cost low. A4 is the one arm whose two rows

| Method | AUC↑ | F1 @ 0.5 | F1 @ val-thr. | F1 @ oracle |
|---|---|---|---|---|
| DeepDTA | $0.8691_{\pm 0.0059}$ | $0.7791_{\pm 0.0042}$ | $0.7849_{\pm 0.0041}$ | $0.7865_{\pm 0.0044}$ |
| PIPR | $0.8840_{\pm 0.0033}$ | $0.7825_{\pm 0.0042}$ | $0.7959_{\pm 0.0040}$ | $0.7980_{\pm 0.0042}$ |
| PPIFormer | $0.8334_{\pm 0.0058}$ | $0.4692_{\pm 0.1417}$ | $0.6676_{\pm 0.0411}$ | $0.7613_{\pm 0.0067}$ |
| PIT-GCL (ours) | $\mathbf{0.9302}_{\pm 0.0037}$ | $\mathbf{0.8413}_{\pm 0.0052}$ | $\mathbf{0.8439}_{\pm 0.0051}$ | $\mathbf{0.8458}_{\pm 0.0051}$ |

Table 7: Threshold transfer on PPIRef, five seeds. "val-thr." selects the threshold maximising F1 on that seed's validation predictions and applies it unchanged to test; "oracle" is the best test threshold, an upper bound no method can reach in practice. The gap between the last two columns is how much a method loses to miscalibration: 0.0937 for PPIFormer against 0.0019 for PIT-GCL. GIGN and D-SCRIPT are omitted because they do not expose per-seed validation predictions in the form this comparison needs.

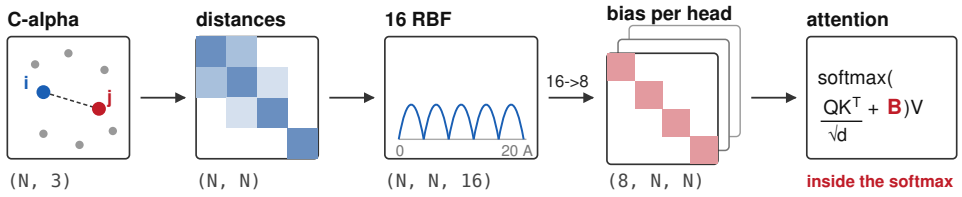



Figure 2: Construction of the structural attention bias $B_{\text{struct}}$. Pairwise C$\alpha$ distances are expanded by 16 radial basis functions whose centres are fixed on $[0, 20]$ Å and whose widths are learned, then projected to one bias matrix per attention head. The bias is added to the attention logits *inside* the softmax, so geometry reweights attention rather than post-processing it, and it is dense over all $N^2$ residue pairs rather than restricted to a neighbourhood graph. The kernel widths and the $16 \to 8$ projection are the only learned parameters of this branch (152 per layer, 608 across the four layers), which is the component removed in arm A6 of Section 4.3.

are parameter-identical (3,030,881), so the pair also serves as a second reading of the descriptor change on a matched architecture.

The two one-hot rows differ in how the code enters the tower. A3$_{H_0}$ projects a 21-dimensional code with a correspondingly narrow input layer (2,913,377 parameters); A3 writes the same code into the leading channels of the full-width input and leaves the projection at its full size (2,989,921, matched to the full model), so its drop is not confounded with a smaller encoder. The two land at $0.9220 \pm 0.0049$ and $0.9058 \pm 0.0114$ against their respective baselines, costs of 0.0063 and 0.0247. We do not read that difference as an effect of either change on its own: the arms differ in the descriptor and in the input projection at once, and A3 carries the widest seed spread in the table ($\pm 0.0114$ against $\pm 0.0028$ for A4 on the same three seeds). The conservative reading is that one-hot identity is a worse substitute for the pretrained embedding than the $H_0$-only row suggested, by an amount this pair of arms cannot localise.

**Homology dimension.** Because the seed also selects the data partition, the effect of a change to the descriptor is measured on a matched partition rather than against a multi-seed mean. On the seed-42 partition, replacing the $H_0$-only descriptor with the $H_0{+}H_1$ descriptor used throughout this paper raises AUC from 0.9325 to 0.9358 ($+0.0033$), PR-AUC from 0.9377 to 0.9418, and F1 from 0.8469 to 0.8500, while lowering RMSE from 0.3302 to 0.3224. All five metrics improve, though the margin is comparable to the seed-to-seed standard deviation in Table 1, so we read this as evidence that $H_1$ does not hurt rather than as a large effect.

**Why A2$_{H_0}$ uses a sinusoidal encoding, and what A2 measures instead.** The natural reading of "no sequence" is to replace $H_{\text{seq}}$ with a single constant, but that makes every token identical, and attention over identical values returns that value regardless of the additive structural bias: the model becomes blind to geometry. Measured directly, the output moved by 0.000000 under a fivefold perturbation of the input coordinates under that configuration, so it measures nothing about the sequence pathway. A2$_{H_0}$ therefore uses a parameter-free sinusoidal encoding of residue index, which removes amino acid identity while keeping residues distinguishable so that the distance-based bias remains active.

| | Arm | AUC↑ | ACC↑ | F1↑ | PR-AUC↑ | RMSE↓ |
|---|---|---|---|---|---|---|
| $\text{main}_{H_0}$ | Full model | $0.9283_{\pm 0.0044}$ | $0.8407_{\pm 0.0046}$ | $0.8387_{\pm 0.0031}$ | $0.9352_{\pm 0.0034}$ | $0.3322_{\pm 0.0063}$ |
| $\text{A2}_{H_0}$ | seq. → sinusoidal | $0.9057_{\pm 0.0045}$ | $0.8124_{\pm 0.0055}$ | $0.8159_{\pm 0.0046}$ | $0.9083_{\pm 0.0046}$ | $0.3565_{\pm 0.0049}$ |
| $\text{A3}_{H_0}$ | seq. → one-hot | $0.9220_{\pm 0.0049}$ | $0.8348_{\pm 0.0061}$ | $0.8310_{\pm 0.0078}$ | $0.9315_{\pm 0.0046}$ | $0.3447_{\pm 0.0072}$ |
| $\text{A4}_{H_0}$ | seq. → ESM-35M | $0.9270_{\pm 0.0040}$ | $0.8385_{\pm 0.0071}$ | $0.8380_{\pm 0.0063}$ | $0.9338_{\pm 0.0033}$ | $0.3354_{\pm 0.0053}$ |
| $\text{A5}_{H_0}$ | floor | 0.4947 | 0.5000 | 0.6667 | 0.4975 | 0.5371 |

Table 8: The sequence arms on the $H_0$-only descriptor, against its own full model. Two of them are a different ablation rather than the same one on a different descriptor, which is why both versions are reported: $\text{A2}_{H_0}$ substitutes sinusoidal encodings for the language-model embedding (2,908,001 parameters) where A2 zeroes it and keeps the projection trained (2,989,921, parameter-matched to the full model), and $\text{A3}_{H_0}$ projects a 21-d one-hot code directly (2,913,377) where A3 places the same code in the leading channels of the full-width input (2,989,921). A4 and $\text{A4}_{H_0}$ are the one like-for-like pair, identical at 3,030,881 parameters, so their difference isolates the descriptor change alone. $\text{A5}_{H_0}$'s 0.4947 is the same constant-output artefact as A5's 0.5401, landing on the other side of chance.

A2 is the constant-substitution arm that argument warns against, and we report it because the configuration is still informative once it is read correctly: unlike A5 its topology branch is live, so its output does vary across pairs (10,237 distinct values over 10,242 test pairs on seed 42, against A5's two) and its $0.8253 \pm 0.0092$ is a real measurement of a real, if narrower, model. We verified the degeneracy on the trained checkpoint rather than assuming it: scaling every C$\alpha$ coordinate by a factor of five leaves the predicted probability unchanged to all printed digits on every pair we tested, while the same perturbation moves A3 by up to $1.1 \times 10^{-3}$ and A1z by up to $6.9 \times 10^{-3}$. What survives in A2 is the persistence landscape, which is computed from the point cloud outside the attention stack and so is unaffected. Nothing else protein-specific survives, since the encoder sees geometry only through the distance bias of Section 3. A2's score is therefore a topology-only reading and belongs beside A1 - which deletes topology and keeps sequence and geometry - rather than beside $\text{A2}_{H_0}$. A5, the floor arm, degenerates both terms and is a leakage check rather than a structure-only arm. It behaves as the argument above predicts: with both $H_{\text{seq}}$ and $h_{\text{topo}}$ constant the network has no protein-specific input left, and its output takes two adjacent float values over the whole test set, varying by $1.0$–$1.4 \times 10^{-7}$ across the three seeds - two units in the last place of `float32` at that magnitude, and a single number once rounded to six decimals. Its accuracy of 0.5000 and F1 of 0.6667 follow from that constant sitting above the 0.5 cut-off on a balanced split. We flag the nominal AUROC of $0.5401 \pm 0.0040$ rather than report it as a result: a ranking metric on a constant score is decided entirely by how the tie-breaking falls out of last-bit arithmetic, and the $H_0$-only row's 0.4947 is the same artefact landing on the other side of chance. Neither number measures the model. What the arm establishes is the constancy itself, which is what a leakage check needs.

## F Implementation Details

**The NT-Xent objective in full.** With $\mathcal{P} = \{i : y_i = 1\}$ the positive pairs of a mini-batch and $\text{sim}_{ij} = \langle z_A^{(i)}, z_B^{(j)} \rangle / \tau$ the similarity between $\ell_2$-normalised pooled embeddings,

$$\mathcal{L}_{\text{NT-Xent}} = \frac{1}{2|\mathcal{P}|} \sum_{i \in \mathcal{P}} \left[ -\log \frac{\exp(\text{sim}_{ii})}{\sum_j \exp(\text{sim}_{ij})} - \log \frac{\exp(\text{sim}_{ii})}{\sum_j \exp(\text{sim}_{ji})} \right].$$

Each positive is its own row's and column's only positive; every other index in the batch acts as an in-batch negative. Note that the benchmark's labelled negatives never serve as anchors, which Section 6 takes up.

We implement PIT-GCL in PyTorch 2.6 with HuggingFace Transformers 5.3, GUDHI 3.11, and BioPython 1.81. All experiments are conducted on a single NVIDIA CUDA GPU. The complete configuration is summarized in Table 9.

One input limit is ours and we state it alongside the baseline limit of the reproducibility statement: ESM-2 accepts at most 1,022 residues, and we truncate the C$\alpha$ coordinates to match so the two modalities stay aligned. No PPIRef or STAG chain reaches that length. On PPB-Affinity 152 of

| Hyperparameter | Value |
|---|---|
| *Sequence features* | |
| ESM-2 model | `facebook/esm2_t6_8M_UR50D` |
| ESM hidden dimension $d_{\text{esm}}$ | 320 |
| *Topological features* | |
| Filtration | Vietoris-Rips |
| Max edge length | 20 Å |
| Max homology dimension | 1 ($H_0 + H_1$) |
| Number of landscapes $n_{\text{land}}$ | 3 |
| Resolution $n_{\text{pts}}$ | 50 |
| *Architecture* | |
| Model dimension $d_{\text{model}}$ | 256 |
| Graph Transformer layers $L$ | 4 |
| Attention heads $H$ | 8 |
| FFN hidden dimension | 512 |
| Dropout | 0.1 |
| RBF kernels (structural bias) | 16 (cutoff 20 Å) |
| Cross-attention heads | 8 |
| Activation | GELU |
| Normalization | Pre-LayerNorm |
| *Loss* | |
| $\alpha$ (contrastive) | 0.5 |
| $\beta$ (BCE) | 0.5 |
| NT-Xent temperature $\tau$ | 0.07 |
| *Optimization* | |
| Optimizer | AdamW |
| Learning rate | $1\times10^{-4}$ |
| Weight decay | $1\times10^{-5}$ |
| Gradient clipping (max-norm) | 1.0 |
| Batch size | 32 on PPIRef, 16 on STAG and PPB-Affinity |
| Maximum epochs | 50 on PPIRef, 100 on STAG and PPB-Affinity |
| Early-stopping patience | 10 on PPIRef, 20 on STAG and PPB-Affinity |
| Validation split (when held-out absent) | 10% |

Table 9: PIT-GCL hyperparameters used across all experiments unless otherwise stated.

20,968 chain slots do, affecting 144 of 10,484 pairs and 2 of the 1,034 test pairs, so the effect on Table 2 is negligible, but a longer-chain benchmark would need a windowed encoder.

## G SCREENING THROUGHPUT

The dual-tower design is motivated by screening cost, so we measure it rather than assert it. All numbers are single-GPU (one NVIDIA A6000), batch size one, on a 60-chain sample of PPB-Affinity whole chains (mean 191 residues; the benchmark's own mean is 242), which is the input a screening pipeline would actually hold. Table 10 gives the per-stage cost.

| Stage | Cost | Recomputed per candidate pair? |
|---|---|---|
| Persistence landscape | 101 ms / protein (median 49) | no |
| Tower encoding (Phase 1) | 2.3 ms / protein | no |
| Fusion + head (Phase 2) | 0.78 ms / pair | yes |

Table 10: Per-stage cost. Only the last row scales with the number of candidates.

Scoring one fixed target against 10,000 candidates therefore costs 17 minutes end to end from structures, of which 16.9 minutes is the one-off persistence computation over the candidate library and 0.5 minutes is tower encoding plus fusion. Re-screening a second target against the same library recomputes only the last row of Table 10, 8 seconds for 10,000 pairs, provided the tower representations are kept; recomputing them as well returns the full 0.5 minutes. Co-folding the same 10,000 pairs at the minutes-per-pair rate cited in Section 1 is on the order of $10^3$ GPU-hours.

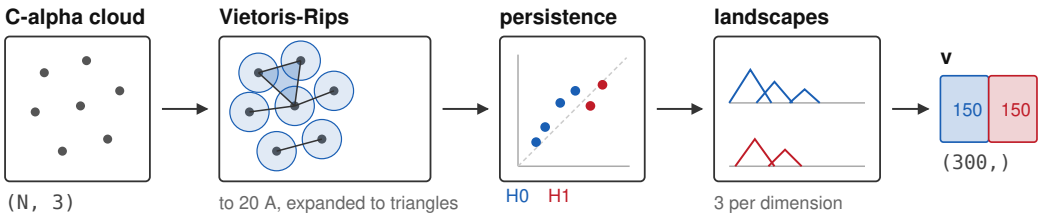



Figure 3: Construction of the topological descriptor. A Vietoris-Rips filtration on the C$\alpha$ cloud, expanded one dimension above the homology dimension so that triangles are present, gives $H_0$ and $H_1$ persistence; each is summarised by three landscapes on a 50-point grid and the two blocks are concatenated into one 300-dimensional vector.

The persistence step is both the dominant term and the one that scales worst. Across the chains we timed it ranges from a few milliseconds to 839 ms, and the slowest cases are the longest: 814 residues costs 0.84 s against 49 ms at the median. The Vietoris-Rips complex is expanded one dimension above the homology dimension being computed, so the simplex count grows quickly with both chain length and the 20 Å cutoff. For a library that is fixed across many targets this is amortised and the descriptor can be shipped alongside the structures; for a library that changes every query it would become the bottleneck, and an alpha complex would be the natural substitute.

## H Persistent Homology Primer

We provide a self-contained summary of the topological constructions used by PIT-GCL. The exposition follows the persistent homology pipeline employed in the topology branch of our model: *point cloud* $\rightarrow$ *simplicial complex* $\rightarrow$ *filtration* $\rightarrow$ *persistence diagram* $\rightarrow$ *persistence landscape*.

**Persistent homology.** Persistent homology studies how topological features of a dataset evolve across multiple scales. Given a point cloud or, more generally, a filtered space, it provides a multi-scale summary of topological invariants, such as connected components ($H_0$), loops ($H_1$), and higher-dimensional voids ($H_k$ for $k \geq 2$), by tracking when these features appear and disappear as a scale parameter varies. Unlike local geometric statistics, this multi-scale summary is invariant under translations and rotations of the input and is provably stable: small perturbations of the input cause only small changes in the resulting summary (Cohen-Steiner et al., 2007). Its construction relies on four key components, simplicial complexes, filtrations, persistence diagrams, and functional summaries such as persistence landscapes (Bubenik, 2015), which we describe in turn below.

**Simplicial complex.** A simplicial complex is a combinatorial object used to encode the topology of a dataset. Formally, a finite abstract simplicial complex $K$ over a vertex set $V$ is a collection of subsets $\sigma \subseteq V$ such that if $\sigma \in K$, then every subset $\tau \subseteq \sigma$ also belongs to $K$. Each $\sigma$ is called a simplex, and its dimension is $\dim(\sigma) = |\sigma| - 1$. In applications, simplicial complexes are often constructed from data points $X \subset \mathbb{R}^d$, for example via the Vietoris-Rips complex (Vietoris, 1927; Edelsbrunner & Harer, 2010),

$$\mathrm{VR}_\epsilon(X) = \{\sigma \subseteq X \mid \|x - y\| \leq \epsilon \quad \forall\, x, y \in \sigma\},$$

which connects all subsets of points whose pairwise distances are below the scale $\epsilon$.

**Homology dimension versus simplex dimension.** These two indices are distinct and are a common source of confusion, so we state the relationship explicitly. A simplex $\sigma$ has dimension $|\sigma| - 1$: a vertex is a 0-simplex, an edge a 1-simplex, a triangle a 2-simplex. The homology group $H_k$ is computed from the chain complex $\cdots \rightarrow C_{k+1} \rightarrow C_k \rightarrow C_{k-1} \rightarrow \cdots$, where $C_k$ is generated by the $k$-simplices of $K$, and $H_k = \ker \partial_k / \operatorname{im} \partial_{k+1}$. Because $H_k$ quotients by the image of $\partial_{k+1}$, a complex that contains no simplices of dimension $k+1$ has $\operatorname{im} \partial_{k+1} = 0$ and therefore reports every $k$-cycle as a feature that never dies. Concretely, $H_1$ counts loops that are not filled in by triangles, so a Vietoris-Rips complex expanded only to dimension 1 (edges) has no triangles with which to fill any loop and yields no finite $H_1$ intervals at all. Computing $H_k$ therefore requires expanding the

| Model Name | AUC↑ | ACC↑ | F1↑ | PR-AUC↑ | RMSE↓ |
|---|---|---|---|---|---|
| STAG | $0.8564_{\pm 0.0077}$ | $0.7671_{\pm 0.0073}$ | $0.6106_{\pm 0.0065}$ | $0.6801_{\pm 0.0154}$ | $0.3907_{\pm 0.0059}$ |
| TEPCAM | $0.6634_{\pm 0.0618}$ | $0.7248_{\pm 0.0856}$ | $0.2939_{\pm 0.0396}$ | $0.3794_{\pm 0.0449}$ | $0.4531_{\pm 0.0443}$ |
| TRAP | $0.6705_{\pm 0.0956}$ | $0.7721_{\pm 0.0105}$ | $0.1668_{\pm 0.0988}$ | $0.4089_{\pm 0.0996}$ | $0.4403_{\pm 0.0240}$ |
| ConTCR | $0.7033_{\pm 0.0030}$ | $0.6358_{\pm 0.0174}$ | $0.4644_{\pm 0.0037}$ | $0.3982_{\pm 0.0084}$ | $0.4655_{\pm 0.0019}$ |
| epiTCR | $\mathbf{0.9031_{\pm 0.0025}}$ | $\underline{0.8120}_{\pm 0.0034}$ | $\underline{0.6766}_{\pm 0.0039}$ | $\mathbf{0.8187_{\pm 0.0029}}$ | $\underline{0.3692}_{\pm 0.0010}$ |
| epiTCR-MHC | $\underline{0.8945}_{\pm 0.0020}$ | $0.7869_{\pm 0.0038}$ | $0.6468_{\pm 0.0044}$ | $0.7412_{\pm 0.0032}$ | $0.3866_{\pm 0.0011}$ |
| NetTCR-2.0 | $0.8003_{\pm 0.0162}$ | $0.8051_{\pm 0.0118}$ | $0.5172_{\pm 0.0600}$ | $0.5858_{\pm 0.0374}$ | $0.3754_{\pm 0.0067}$ |
| TITAN | $0.6751_{\pm 0.0004}$ | $0.7691_{\pm 0.0000}$ | $0.0000_{\pm 0.0000}$ | $0.3640_{\pm 0.0010}$ | $0.4075_{\pm 0.0000}$ |
| PIT-GCL (ours) | $0.8844_{\pm 0.0032}$ | $\mathbf{0.8714_{\pm 0.0037}}$ | $\mathbf{0.7152_{\pm 0.0085}}$ | $\underline{0.7637}_{\pm 0.0101}$ | $\mathbf{0.3319_{\pm 0.0052}}$ |

Table 11: Performance on the TCR-pMHC benchmark, STAG test split.

complex to dimension $k+1$: PIT-GCL uses $H_0$ and $H_1$, and expands the Vietoris-Rips complex to dimension 2.

**Filtration.** A filtration is a nested sequence of simplicial complexes indexed by a scale parameter,

$$K_{\epsilon_1} \subseteq K_{\epsilon_2} \subseteq \cdots \subseteq K_{\epsilon_T}, \qquad \epsilon_1 \leq \epsilon_2 \leq \cdots \leq \epsilon_T.$$

More generally, a filtration can be induced by a function $f : K \to \mathbb{R}$, where each sublevel set $K_\alpha = \{\sigma \in K \mid f(\sigma) \leq \alpha\}$ forms a simplicial complex. Persistent homology studies how topological features evolve along this filtration.

**Persistence diagram.** For each dimension $k$, applying homology to a filtration yields a sequence of homology groups $H_k(K_\epsilon)$ and tracks the birth and death of homological features as $\epsilon$ varies. Each feature is associated with a persistence interval $(b_i, d_i)$, where $b_i$ and $d_i$ denote the parameter values at which the feature appears and disappears, respectively. The persistence diagram is defined as the multiset

$$D_k = \{(b_i, d_i) \in \mathbb{R}^2 \mid i = 1, \ldots, n_k\}.$$

Each point represents a topological feature, and its persistence $d_i - b_i$ reflects its scale stability: features with larger persistence are typically interpreted as more salient.

**Persistence landscape.** To enable statistical analysis and compatibility with machine learning pipelines, persistence diagrams are often mapped to functional representations. The persistence landscape (Bubenik, 2020) is a sequence of functions $\lambda_k : \mathbb{R} \to \mathbb{R}$ defined as follows. For each interval $(b_i, d_i)$, define a triangular function

$$\Lambda_i(t) = \max\big(0,\ \min(t - b_i,\ d_i - t)\big).$$

The $k$-th persistence landscape is then the $k$-th largest value of $\{\Lambda_i(t)\}_i$ at each $t$,

$$\lambda_k(t) = k\text{-th largest value of } \{\Lambda_i(t)\}_i.$$

The collection $\{\lambda_k(t)\}_{k \geq 1}$ provides a stable, vectorizable representation of the persistence diagram and can be discretized on a fixed grid to yield a finite dimensional feature vector. PIT-GCL uses persistence landscapes derived from the $H_0$ and $H_1$ persistence diagrams of a Vietoris-Rips filtration (Vietoris, 1927; Edelsbrunner & Harer, 2010) on the C$\alpha$ point cloud as a global topological descriptor of each protein.

# I TCR-pMHC Results on STAG

Section 4 summarises this benchmark in one paragraph and leaves the analysis here, because the topological claim the paper argues is about general PPI: STAG is a single-family task in which the fold is largely shared across pairs, and we use it to test whether the architecture transfers, not to attribute modalities.

**Reading the table.** Table 11 compares PIT-GCL against TCR-pMHC predictors on the STAG benchmark. The strongest baseline is epiTCR (AUC $0.9031 \pm 0.0025$, PR-AUC $0.8187 \pm 0.0029$), with its MHC-aware variant close behind and STAG itself at $0.8564 \pm 0.0077$. TRAP and TITAN reach majority-class accuracy with F1 far below it ($0.1668$ and $0.0000$), so their predictions collapse towards one class - TITAN identically in all five runs, TRAP with an F1 standard deviation exceeding its mean. TEPCAM and ConTCR stay below $0.71$ AUC. This may reflect distributional differences from these methods' original training data, though preprocessing and calibration may also contribute.

PIT-GCL leads on the threshold-dependent metrics and trails on the ranking ones, and we state both directions against the combined standard deviation of the two methods being compared. It is ahead on accuracy ($+0.0594$ over epiTCR, $11.8\times$), F1 ($+0.0386$, $4.1\times$) and RMSE ($-0.0373$, $7.0\times$). It is behind epiTCR on AUC ($-0.0187$, $4.6\times$) and PR-AUC ($-0.0550$, $5.2\times$), by margins as far outside seed variation as our own leads. On this benchmark, therefore, PIT-GCL is not the best-ranking model: a sequence-only TCR-pMHC predictor orders pairs better, while PIT-GCL converts its scores into decisions more effectively under the $76.8\%$ class imbalance of this split. We do not claim state of the art on TCR-pMHC, and read the result as evidence that a general-purpose PPI architecture does not automatically dominate a task-specific one on that task's own benchmark.

## J Baseline Model Details

We summarize each baseline by its architecture family and the input modalities it consumes. The *Architecture* column reports the dominant building block (e.g., 1D CNN, Transformer, GNN, contrastive learning), and the *Input Repr.* column describes which protein representations are fed into the model.

**PPI baselines.** PPI baselines take two protein representations as input and produce a binding score. DeepDTA was originally proposed for drug-target affinity and consumes 1D-tokenized SMILES and protein sequences; we adapt it to PPI by feeding two protein sequences in place of the drug and target inputs. PIPR is a sequence-only PPI predictor with a Siamese residual RCNN. D-SCRIPT couples a pretrained Bepler-Berger language model with a CNN that predicts an inter-chain contact map and pools it into a binding probability. PPIFormer operates on the 3D structure of the interface using an SE(3)-equivariant Equiformer encoder pretrained with masked modeling on PPIRef. Table 12 summarizes these baselines.

**Training protocol per baseline.** DeepDTA, PIPR, D-SCRIPT and GIGN were each retrained from scratch on the PPIRef split used by PIT-GCL, so all four are evaluated under the same training distribution and the same train/val/test partition. GIGN was designed for protein-ligand binding affinity; retraining it here adapts a geometric interaction network to a different interaction type, and Appendix B reads its scores accordingly. PPIFormer is already trained on PPIRef in its self-supervised masked-modeling form, so we use the released checkpoint without further training, and it is the one exception in Tables 1 and 2. This separation reflects what is reproducibly available for each method rather than a methodological preference.

**The general-purpose PPI baselines on PPB-Affinity.** DeepDTA, PIPR and D-SCRIPT are carried over to PPB-Affinity and retrained on its training split. Two details differ from their PPIRef runs, both forced by the benchmark. First, our PPB-Affinity negatives are constructed once and written to fixed train/val/test CSVs rather than drawn per run, so a seed varies initialisation and batch order only; this is also how PIT-GCL's PPB seeds vary, and the error bars in Table 2 are comparable for that reason. Second, the sequence truncation length rises from $100$ to $512$. The shorter value was chosen for PPIRef, whose rows are extracted interface fragments; PPB-Affinity rows are whole chains with a median length of $199$ residues and a $95$th percentile near $500$, so keeping $100$ would discard most of every chain and understate the baselines. D-SCRIPT additionally drops pairs in which either chain is shorter than its contact CNN's kernel width, which removes $82$, $20$ and $18$ pairs from train, validation and test respectively; the same floor applies to its PPIRef runs.

On STAG the situation is uniform: every baseline in Table 11 was retrained from scratch on the STAG training split rather than evaluated from a released checkpoint. This matters for reading that table, because several of these predictors are much stronger here than a transfer evaluation of the

| Model | Year | Architecture | Input Repr. |
|---|---|---|---|
| DeepDTA (Öztürk et al., 2018) | 2018 | CNN | Two protein sequences |
| PIPR (Chen et al., 2019) | 2019 | RCNN | Two protein sequences |
| D-SCRIPT (Sledzieski et al., 2021) | 2021 | Pretrained PLM + CNN | Two protein sequences |
| PPIFormer (Bushuiev et al., 2024) | 2024 | Transformer | 3D structure |
| GIGN (Yang et al., 2023) | 2023 | Geometric Interaction GNN | 3D structure |
| GraphDTA (Nguyen et al., 2021) | 2021 | GNN + CNN | Two protein sequences |

Table 12: Summary of the PPI baselines. GraphDTA is evaluated on PPB-Affinity only; the others are evaluated on PPIRef and, where marked in Table 2, retrained on PPB-Affinity as well.

| Model | Year | Architecture |
|---|---|---|
| STAG (Slone et al., 2025) | 2025 | Graph Neural Network |
| TEPCAM (Chen et al., 2024a) | 2024 | CNN + attention |
| TRAP (Ge et al., 2025) | 2025 | Transformer + contrastive learning |
| ConTCR (Shuai et al., 2025) | 2025 | Cross-attention Transformer + contrastive learning |
| epiTCR (Pham et al., 2023) | 2023 | Random Forest |
| epiTCR-MHC (Pham et al., 2023) | 2023 | Random Forest (MHC-aware) |
| NetTCR-2.0 (Montemurro et al., 2021) | 2021 | CNN |
| TITAN (Weber et al., 2021) | 2021 | Bimodal attention |

Table 13: Summary of TCR-pMHC baseline model architectures.

same architectures would suggest - epiTCR, for instance, reaches an AUC of $0.9031$ after retraining. Reported numbers for TCR-pMHC predictors are therefore not comparable across papers unless the training protocol is matched.

**TCR-pMHC baselines.** TCR-pMHC baselines vary in which components of the TCR-pMHC system they consume: some use only CDR3$\beta$ and peptide, while others additionally use CDR3$\alpha$ or MHC information; CDR3$\alpha$ and CDR3$\beta$ are the two hypervariable loops of the TCR. STAG is the only structure-based method in this group and operates directly on the 3D complex. Sequence-based methods differ in encoder choice (1D CNN, Transformer, language model, random forest) and in whether they incorporate contrastive learning to improve generalization. Table 13 lists the architectures and Table 14 the input representations.

| Model | Sequence | | | | Structure | | | |
|---|---|---|---|---|---|---|---|---|
| | **Peptide** | **MHC** | **CDR3$\alpha$** | **CDR3$\beta$** | **Peptide** | **MHC** | **CDR3$\alpha$** | **CDR3$\beta$** |
| STAG | | | | | ✓ | ✓ | ✓ | ✓ |
| TEPCAM | ✓ | | | ✓ | | | | |
| TRAP | ✓ | | | ✓ | ✓‡ | ✓‡ | | |
| ConTCR | ✓ | ✓ | | ✓ | | | | |
| epiTCR | ✓ | | | ✓ | | | | |
| epiTCR-MHC | ✓ | ✓ | | ✓ | | | | |
| NetTCR-2.0 | ✓ | | ✓ | ✓ | | | | |
| TITAN | ✓† | | | ✓ | | | | |

Table 14: Input representations used by TCR-pMHC baseline models. †TITAN represents epitopes using SMILES rather than standard amino-acid sequence encoding. ‡TRAP is the one baseline whose structural input is predicted rather than resolved: it folds each pMHC with AlphaFold-Multimer and extracts features from the epitope's local environment, so the MHC enters through that structure and not as a sequence embedding.