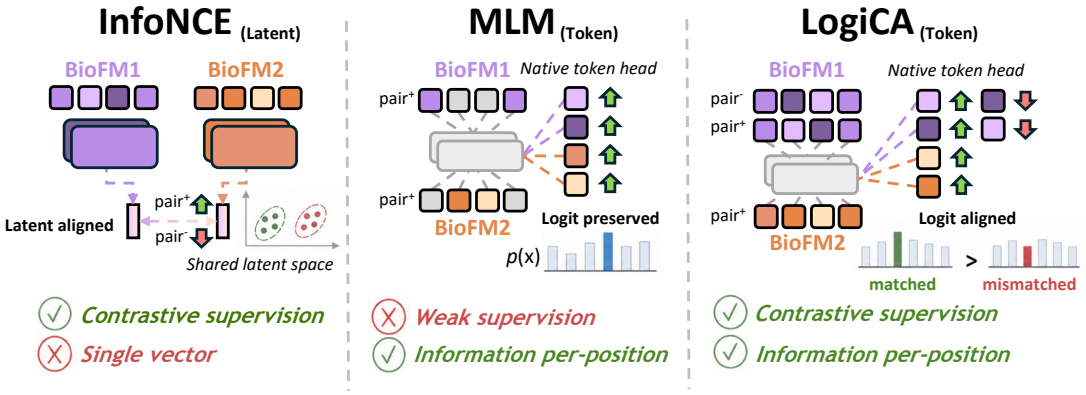
Contextualizing Biological Language Models across Modalities via Logit-Space Contrastive Alignment

Anonymous Authors[1]



Abstract

Pretrained biological language models expose per-token probability distributions through masked-token prediction, providing the likelihood interface central to sequence design, variant scoring, and mechanistic interpretation. Yet these distributions are learned from broad unlabeled corpora and are not naturally conditioned on task-specific biological contexts such as interaction partners, cellular environments, or therapeutic interventions. Existing contextual matching methods distort this interface with pooled embeddings, contrastive latent spaces, or task-specific prediction heads We introduce **LogiCA** (*Logit-space Contrastive Alignment*), a framework for context-conditioned prediction that adapts pretrained biological language models by performing contrastive learning directly in output-logit space. Using gated cross-modal adapters compatible with each model's native token head, LogiCA preserves the pretrained per-token likelihood interface and converts contextualized token log-likelihoods into matching scores. Alignment is therefore defined through context-sensitive token probabilities rather than proximity in a shared embedding space. This enables learning from sparse paired data across models with **distinct vocabularies**, without requiring a shared tokenizer, decoder, or embedding space. LogiCA is particularly effective for **mutation-local variant ranking**, where variant comparisons reduce to context-conditioned likelihoods of mutant tokens at perturbed sites. Across protein–ligand binding, TCR–peptide activity, and drug-conditioned resistance prediction, LogiCA yields substantial gains over prior state-of-the-art methods, including matched latent-contrastive and conditional-MLM baselines, while retaining a token-level interface for interpretation and generation. On held-out-gene single-mutation drug-resistance prediction, LogiCA improves the AUC from the near-random latent-space baselines of ∼0.55 to ∼0.65. Code is available at: https://anonymous.4open.science/r/logica/.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Submitted to the 2026 Workshop on Generative and Agentic AI for Biology (ICML 2026). Do not distribute.

1. Introduction

Pretrained biological language models (BioFMs) have become foundational tools for modeling proteins, genomes,

and molecules because they define normalized token distributions over biological vocabularies. These distributions assign probabilities to biological tokens at each position, enabling zero-shot mutation scoring, sequence design, evolutionary analysis, token-level interpretation, and de novo generation (Lin et al., 2023; Riesselman et al., 2018; Meier et al., 2021; Hie et al., 2024; 2022; Yüksel et al., 2023; Dalla-Torre et al., 2025; Cui et al., 2025; Brixi et al., 2026; Tomaz da Silva et al., 2025; Gong et al., 2024; Zhang et al., 2024b; Madani et al., 2023; Johnson et al., 2021; McCarter et al., 2026). This position-level formalism is especially valuable in biology, where functional changes often arise from small perturbations localized to individual residues, nucleotides, or molecular tokens.

Many therapeutic prediction problems, however, are inherently contextual. A T-cell receptor should be scored under a peptide, a ligand under a protein target, and a resistance mutation under a drug. In such settings, the question is not only whether a sequence is plausible, but whether it is plausible in a specific biological context. Standard pretrained language-model logits capture broad evolutionary, structural, or chemical regularities, but are not trained to directly encode compatibility with a particular binding partner, peptide, or treatment condition.

Most existing approaches address contextual matching in one of two ways (Appendix A). The first class leaves the language model's token interface behind, learning scalar compatibility scores from pooled representations, contrastive embedding alignment, or task-specific prediction heads (Abramson et al., 2024; Gao et al., 2023; Passaro et al., 2025; Huang et al., 2021; Jia et al., 2026; Singh et al., 2023; Bai et al., 2023; Liu et al., 2025b; Yu et al., 2025; Shoshan et al., 2026). Such scores are effective for retrieval and binary matching, but are no longer token likelihoods, and therefore do not naturally localize to mutated residues, support likelihood-based generation, or expose the position-wise probabilities that make pretrained biological language models useful. The second class preserves the token interface by fine-tuning with conditional masked-language-modeling objectives (Ullanat et al., 2026; Mizrahi et al., 2023; Meynard-Piganeau et al., 2024; Karthikeyan et al., 2025; Chen et al., 2025; Burbach & Briney, 2024; Liu et al., 2025a; Lupo et al., 2024). However, paired biological datasets are often sparse, noisy, and context-specific; reconstruction losses encourage token recovery but do not directly separate matched from mismatched contexts. This is especially limiting when functional signal is localized to small sequence perturbations: pooled objectives can dilute the effect, while reconstruction losses may fail to distinguish the correct context from plausible alternatives.

Our contribution. We introduce **LogiCA**, a logit-space contrastive alignment framework for contextual biological prediction. Rather than using pooled latents or classifier heads, LogiCA uses gated cross-modal adapters that preserve each pretrained model's native token head, and uses the resulting contextualized token log-likelihoods as matching scores. The learned representations are therefore not the object of alignment; they are a mechanism for producing context-sensitive native token distributions. This enables **contrastive learning directly in logit space**, preserving the probabilistic interface of pretrained language models while aligning sparse paired data across **distinct vocabularies**. LogiCA is especially suited to **mutation-local variant ranking**: when variants share a wild-type reference and mutated sites, shared sequence terms cancel, leaving pairwise comparisons over the context-conditioned likelihoods of mutant tokens. This yields a token-level contrastive objective analogous to InfoNCE, but defined over logits at perturbed positions rather than global sequence embeddings. Empirically, we apply LogiCA to protein–ligand binding, TCR–peptide ranking, and drug-conditioned resistance scoring by contextualizing ESM-2 (Lin et al., 2023), TCRLang (Raybould et al., 2024), and SELFormer (Yüksel et al., 2023) while **retaining their native output heads**. LogiCA achieves state-of-the-art zero-shot variant-ranking performance on deep mutational scanning (DMS) assays for peptide activity and drug resistance directly from its logit outputs, outperforming latent-space, conditional-MLM, and existing external baselines.

2. LogiCA: Logit-space Contrastive Alignment

2.1. Contextual ranking in logit space

Let $x = (x_1, \ldots, x_L)$ be a sequence to be scored, let y be an external biological context, and let $A \subseteq [L]$ denote the positions at which the score is evaluated. A contextualized biological language model defines, for each position i, a token distribution

$$\pi_\theta(\cdot \mid x_{\setminus i}, y)$$

over the native vocabulary of x. Our goal is to use these conditional token probabilities not only for reconstruction, but as compatibility scores in a contextual ranking problem. Given an anchor a, a candidate set $\mathcal{C}$, and any scalar compatibility score $s(a, c)$, the induced ranking distribution is

$$p(c \mid a, \mathcal{C}) = \frac{\exp(s(a,c)/\tau)}{\sum_{c' \in \mathcal{C}} \exp(s(a,c')/\tau)}. \tag{1}$$

The anchor and candidates may be either biological sequences or external contexts, so this form covers both context retrieval and variant ranking. With multiple candidates, Eq. 1 yields the InfoNCE objective (van den Oord et al., 2018); with two candidates, it reduces to the Bradley–Terry preference loss (Bradley & Terry, 1952; Burges et al., 2005; Rafailov et al., 2023). Thus, the main modeling choice

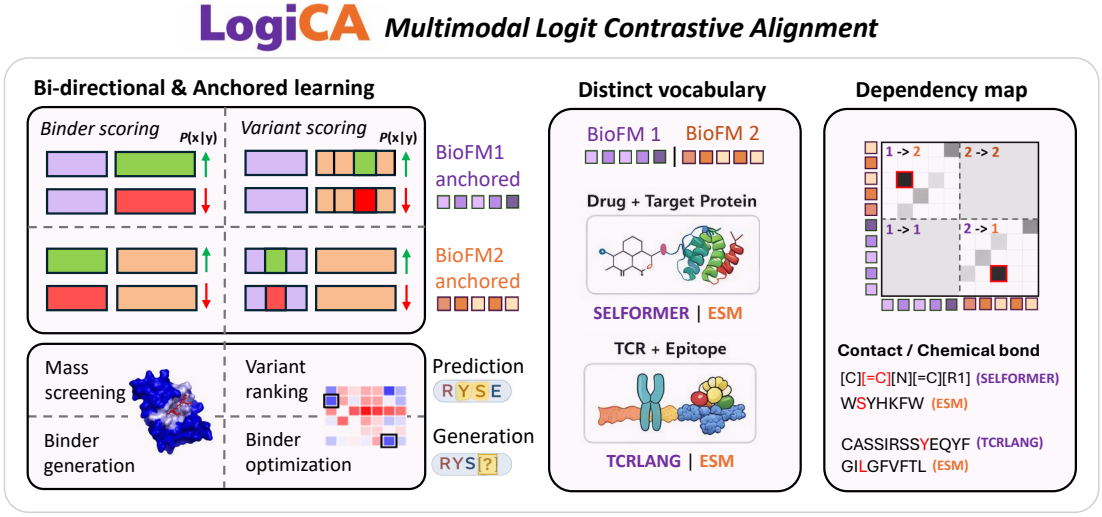


Figure 1. Overview of LogiCA: pretrained biological language models are coupled by cross-modal adapters that preserve native token heads, enabling contrastive alignment of context-conditioned logit-probability distributions across distinct modal vocabularies.

is not the ranking loss itself, but the biological quantity used to define $s(a, c)$. Latent-space contrastive models typically choose a pooled-representation score, such as $s_z(x, y) = \langle f_\theta(x), g_\phi(y) \rangle$. This CLIP-style objective can align modalities effectively (Radford et al., 2021), but the resulting compatibility score is detached from the model's original residue-level likelihoods.

LogiCA keeps the same contextual ranking template, but moves the score into the language model's native logit space. Thus, the embeddings are not trained to be directly comparable across modalities; they are trained only insofar as they induce useful context-conditioned token probabilities through the native output heads. We define compatibility by the site-averaged conditional log-likelihood

$$s_{\text{LogiCA}}(x, y; A) = \ell_A(x \mid y) = \frac{1}{|A|} \sum_{i \in A} \log \pi_\theta(x_i \mid x_{\setminus i}, y). \tag{2}$$

For sequence-level matching we set $A = [L]$. For localized tasks, A can be restricted to mutation sites, binding-interface residues, or other biologically meaningful subsets. The same scalar score can therefore be used inside Eq. 1, while remaining decomposable into position-level token probabilities.

2.2. Mutation-local variant ranking

For variant comparison, the token-level probabilistic interface preserved by LogiCA allows scores to focus on perturbed positions. Let x^{wt} be a reference sequence and $\mathcal{M}(x, x^{\text{wt}}) = \{i : x_i \neq x_i^{\text{wt}}\}$ denote the mutated sites. We consider position-aligned substitutions, excluding insertions and deletions. We define the context-conditioned mutation score

$$s_{\mathcal{M}}(x, y; x^{\text{wt}}) = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \log \frac{\pi_\theta(x_i \mid x_{\setminus i}, y)}{\pi_\theta(x_i^{\text{wt}} \mid x_{\setminus i}^{\text{wt}}, y)}. \tag{3}$$

The score is positive when context y favors the substituted residues over their wild-type counterparts.

Proposition 2.1 (Mutation-local cancellation)**.** *Let x^A and x^B be variants of the same wild-type sequence x^{wt} under context y, and suppose they perturb the same nonempty mutation set*

$$\mathcal{M} = \mathcal{M}(x^A, x^{\text{wt}}) = \mathcal{M}(x^B, x^{\text{wt}}).$$

Define

$$\Delta = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \Big[\log \pi_\theta(x_i^A \mid x_{\setminus i}^A, y) - \log \pi_\theta(x_i^B \mid x_{\setminus i}^B, y) \Big].$$

Then

$$s_{\mathcal{M}}(x^A, y; x^{\text{wt}}) - s_{\mathcal{M}}(x^B, y; x^{\text{wt}}) = \Delta.$$

Consequently, for the two-candidate Bradley–Terry ranking model,

$$\Pr(x^A \succ x^B \mid y, \{x^A, x^B\}) = \sigma(\Delta / \tau).$$

Thus, for matched mutation sets, the wild-type likelihood terms cancel exactly, and the ranking objective reduces to a direct comparison of the context-conditioned mutant-token likelihoods at the perturbed sites. Unchanged residues affect these likelihoods through the conditioning sequence,

but they do not appear as explicit score terms in the pairwise gap. This exact mutation-local reduction relies on the native token-likelihood interface preserved by LogiCA; latent scores do not generally admit an analogous cancellation because they compare global sequence representations. Appendix C provides the proof, score-level gradient analysis, and multi-site concentration bound.

2.3. Bidirectional logit-space matching

When both modalities have pretrained token heads, LogiCA can evaluate compatibility in both conditional directions. This bidirectionality is natural for pairwise biological interactions: compatibility is a property of the pair, even though token likelihoods are directional. The score $\ell_{A_x}(x \mid y)$ asks whether context y makes the evaluated tokens of x likely, whereas $\ell_{A_y}(y \mid x)$ asks the reciprocal question. Because these directional likelihoods may differ in sequence length, vocabulary size, entropy scale, and pretrained head calibration, we combine them with a learned convex mixture:

$$s_\alpha(x, y) = \alpha\, \ell_{A_x}(x \mid y) + (1-\alpha)\, \ell_{A_y}(y \mid x), \qquad \alpha \in [0, 1]. \tag{4}$$

Here A_x and A_y denote the evaluated token positions in the two modalities. The endpoints recover the two anchored conditional scores, while intermediate values let the model balance evidence from the two native logit spaces.

Substituting s_α into Eq. 1 yields the LogiCA contrastive objective. Alignment is therefore performed directly in logit space: positives and mismatched partners are separated by the likelihoods assigned by each model's original token head, without requiring a shared vocabulary, shared decoder, pooled-latent similarity, or separate pair-classification head.

2.4. Native-head-preserving cross-modal adapters

LogiCA contextualizes pretrained biological language models by introducing native-head-preserving cross-modal adapters (Figure S4). Cross-attention is a standard mechanism for coupling pretrained encoders across modalities (Tan & Bansal, 2019; Garau-Luis et al., 2024); here, we use it to produce contextual updates that remain compatible with each backbone's original token head.

Let H^x and H^y be token-level hidden states extracted from pretrained encoders for the two modalities. The adapter projects both streams into a shared interaction width, yielding Z_0^x and Z_0^y, applies N bidirectional cross-attention blocks to obtain Z_N^x and Z_N^y, and returns the accumulated update to each native hidden space through a gated residual:

$$H_c^m = H^m + \sigma(g_m)\, \phi_m(Z_N^m - Z_0^m), \qquad m \in \{x, y\}. \tag{5}$$

Here, $Z_N^m - Z_0^m$ is the adapter update for modality m, ϕ_m maps it back to the corresponding native hidden dimension, and the near-zero gate initialization keeps H_c^m close to H^m at the start of training. The contextualized states H_c^m are then scored by the original language-model heads, preserving probabilities over native token vocabularies.

2.5. Training with anchored negatives

The ranking objective in Eq. 1 is defined by the training candidate set, making pair construction a central design choice in LogiCA. For each matched pair (x, y), training constructs anchored negatives by holding one modality fixed and replacing the other with corrupted, mutated, or mismatched alternatives.

Alternating the anchor provides supervision for both $\ell_{A_x}(x \mid y)$ and $\ell_{A_y}(y \mid x)$, supporting the bidirectional formulation in Eq. 4. Effective training uses a negative pool that combines local single- or few-token perturbations, which emphasize mutation-sensitive positions, with more distant negatives that preserve global pair-level discrimination.

2.6. Using trained logits for ranking, interpretation, and generation

Because LogiCA preserves native token-probability outputs, the same trained model can be used for ranking, interpretation, and generation without adding task-specific heads (Appendix B). Variant candidates are ranked directly by the mutation-local likelihood-ratio score in Eq. 3.

To interpret cross-modal interactions, we probe how perturbing one token changes the conditional distribution at another. For a context-token substitution $y_j \to a$, define

$$D_{ij}^{y \to x} = \frac{1}{|\mathcal{A}_j^y|} \sum_{a \in \mathcal{A}_j^y} \left\| \pi_{\theta,i}(\cdot \mid x_{\setminus i}, y^{(j \to a)}) - \pi_{\theta,i}(\cdot \mid x_{\setminus i}, y) \right\|_2. \tag{6}$$

Here, $\mathcal{A}_j^y$ denotes the set of valid substitutions considered at context position j. Large $D_{ij}^{y \to x}$ indicates that token j in the context strongly influences the model's belief about token i in the scored sequence. These cross-modal dependency maps expose which residues or molecular tokens drive the learned compatibility score. Finally, since LogiCA remains a conditional language model, its logits can be used for context-conditioned generation by Gibbs sampling over selected design positions, as described in Appendix B.3.

3. Experiments

3.1. Protein–ligand LogiCA for binding and resistance scoring

We evaluate whether LogiCA can use contextualized token likelihoods as protein–ligand compatibility scores. Proteins are encoded with ESM-2 650M (Lin et al., 2023), ligands are represented as SELFIES strings and encoded with SELF-

ormer (Yüksel et al., 2023) (86.7M parameters), and cross-modal adapters condition each modality on the other while preserving the native token heads. We use SELFormer rather than ChemBERTa (Chithrananda et al., 2020) because its SELFIES-based masked-token objective provides a native ligand-side likelihood interface (Krenn et al., 2020), matching the token-level protein formulation used by LogiCA.

Binding prediction. We first pretrain on ∼20M protein–ligand binding measurements from BindingDB (Liu et al., 2025c). High-affinity interactions are defined as the top quartile within each assay modality (K_d, K_i, IC_{50}, EC_{50}) and contrasted against partner-swapped negatives drawn from the remaining quartiles. To prevent leakage, we remove from the pretraining corpus any protein–ligand pair that appears in the downstream validation or test splits, with details provided in Appendix D.1. We then evaluate on established drug–target interaction (DTI) benchmark splits from prior work, including DAVIS (Davis et al., 2011), BindingDB-test (Huang et al., 2021), and BioSNAP (Zitnik et al., 2018), using each benchmark's published split protocol (Appendix F.1).

For controlled ablations, we fix the ESM-2–SELFormer architecture and training data, varying only the objective: latent contrastive alignment (LatentCA), conditional masked-token adaptation (LogiMLM), or logit-space ranking (LogiCA). LatentCA mean-pools each tower and scores pairs by cosine similarity, replacing token likelihoods with a single latent score. LogiMLM preserves native heads but trains only on positive binding quartiles with a standard masked-token objective, without negative contrastive pairs (Appendix E.1). Table 1 shows that LogiCA outperforms both ablations on every benchmark, indicating that contrastive alignment and token-level scoring are jointly useful. Compared with external DTI baselines (Appendix E.2), LogiCA is competitive with strong sequence-only and classifier-based methods, and remains close to structure-informed baselines despite not using structural inputs. Overall, token-level contrastive learning is not detrimental relative to latent alignment; it improves binding prediction while preserving the position-specific scoring interface needed for mutation-local analysis.

Drug resistance prediction. We next test whether the pretrained protein–ligand likelihood interface can be adapted for drug-conditioned mutation resistance scoring and transfer to held-out resistance settings. The drug-screening assays are protein DMS experiments, where variants are scored by their measured resistance phenotype under each drug (Appendix D.2). After fine-tuning on resistance assay, LogiCA and LogiMLM scores held-out variants directly from contextualized logit outputs at the mutated positions (Eq. 3). For drug-conditioned variant ranking, cosine similarity between protein and drug pools is degenerate because every variant is paired with the same drug; we therefore use *w/ LatentFuse*, an MLP over pooled protein and drug vectors, as the matched latent-score ablation (Appendix E.2).

On the multi-oncogene resistance panel from Coelho et al. (Coelho et al., 2024), the two native-likelihood models, LogiMLM and LogiCA, are the only methods that substantially improve over near-random protein-only and structural baselines when SELFormer drug context is added to the ESM-2 protein backbone. LogiCA achieves the strongest performance within this contextualized sequence-model family, topping the rankings for 8 out of 10 genes (Table S12). By contrast, LatentFuse, DrugBAN (Bai et al., 2023), Boltz-2 (Passaro et al., 2025), and DrugCLIP (Jia et al., 2026) remain close to random despite task-matched fine-tuning on the same mutation-local resistance objective (Table 2). Thus, the gain does not come from drug context alone, but from coupling that context to a mutation-local likelihood interface.

Generic protein fitness is often insufficient to determine whether a mutation is beneficial or deleterious under a particular therapeutic context, while hidden representations pretrained for global protein–ligand compatibility are not well suited to exposing local, drug-conditioned mutational effects through a downstream probe or head. In contrast, LogiCA's pretraining aligns biological context through native token likelihoods, so fine-tuning can directly refine a mutation-local scoring interface rather than extract local effects from global pair representations. The EGFR-focused panel (Kim et al., 2025) is a notable exception, strong sequence-only priors remain competitive, suggesting that generic fitness already explains a large fraction of the measured resistance signal.

Scaling of protein–ligand LogiCA. We next examine how the likelihood interface scales with backbone capacity and downstream supervision. For pretraining scale, we train LogiCA with ESM-2 backbones from 8M to 650M parameters while keeping the SELFormer ligand encoder fixed (Appendix F.3). As shown in Figures 2A,B, larger protein backbones yield higher held-out matched-versus-mismatched likelihood margins, with the peak margin following $\propto N^{0.16}$ over the $80\times$ parameter range. For downstream data scale, we vary the fraction of available target-gene resistance labels from 0% to 15% and compare LogiCA with LogiMLM under the same backbone setting. Both objectives benefit from additional supervision, but LogiCA remains consistently stronger, with the largest gains in the intermediate-label regime (10–15%; Figure 2C).

3.2. TCR–peptide LogiCA: zero-shot variant ranking and biomolecular contacts

We next evaluate LogiCA in TCR–peptide recognition, a notoriously challenging problem with sparse and noisy

Table 1. Protein–ligand binding prediction on DTI benchmarks. Reported reproduced multi-run cells are five-run means ± standard deviations. The blue block compares contextualized sequence-model variants (SELFormer–ESM-2). The gray block lists external sequence-only DTI baselines, which share the same input modality and are the direct comparators for LogiCA. The orange block lists structure-informed baselines (marked ‡); these consume additional structural information and are reported for reference, not as direct comparators. **Bold** marks the best result per column among sequence-only methods (blue + gray blocks); underline marks the second best in the same scope. The † marker denotes methods that retain a native-vocabulary generative interface.

Method	DAVIS		BindingDB (Test)		BioSNAP	
	AUC	AUPR	AUC	AUPR	AUC	AUPR
Contextualized backbone (ours)						
w/ LatentCA latent contrastive	0.904 ± 0.007	0.371 ± 0.012	0.812 ± 0.040	0.481 ± 0.061	0.907 ± 0.012	0.915 ± 0.013
w/ LogiMLM† logit-level MLM	0.739 ± 0.001	0.197 ± 0.007	0.754 ± 0.001	0.332 ± 0.004	0.815 ± 0.001	0.846 ± 0.002
w/ LogiCA† **token-score contrastive**	**0.924** ± 0.005	**0.446** ± 0.019	0.906 ± 0.003	**0.635** ± 0.014	**0.921** ± 0.003	**0.929** ± 0.004
Sequence-only baselines						
MolTrans (Huang et al., 2021) classifier head	0.901 ± 0.006	0.343 ± 0.024	0.904 ± 0.005	0.584 ± 0.010	0.886 ± 0.007	0.894 ± 0.009
ConPLex (Singh et al., 2023) latent contrastive	0.887 ± 0.007	0.441 ± 0.033	0.854 ± 0.004	0.614 ± 0.009	0.866 ± 0.006	0.888 ± 0.005
DrugBAN (Bai et al., 2023) classifier head	0.883 ± 0.006	0.349 ± 0.022	**0.912** ± 0.002	0.617 ± 0.006	0.907 ± 0.002	0.916 ± 0.004
Structure-informed baselines						
DrugCLIP‡ (Jia et al., 2026) latent contrastive	0.925 ± 0.009	0.473 ± 0.025	0.863 ± 0.010	0.581 ± 0.025	0.788 ± 0.006	0.821 ± 0.007
SP-DTI‡ (Liu et al., 2025b) classifier head	0.924 ± 0.002	0.424 ± 0.023	0.921 ± 0.001	0.645 ± 0.014	0.924 ± 0.003	0.923 ± 0.005
GS-DTI‡ (Yu et al., 2025) latent contrastive	0.916 ± 0.012	0.430 ± 0.038	0.920 ± 0.004	0.669 ± 0.012	0.934 ± 0.003	0.934 ± 0.004

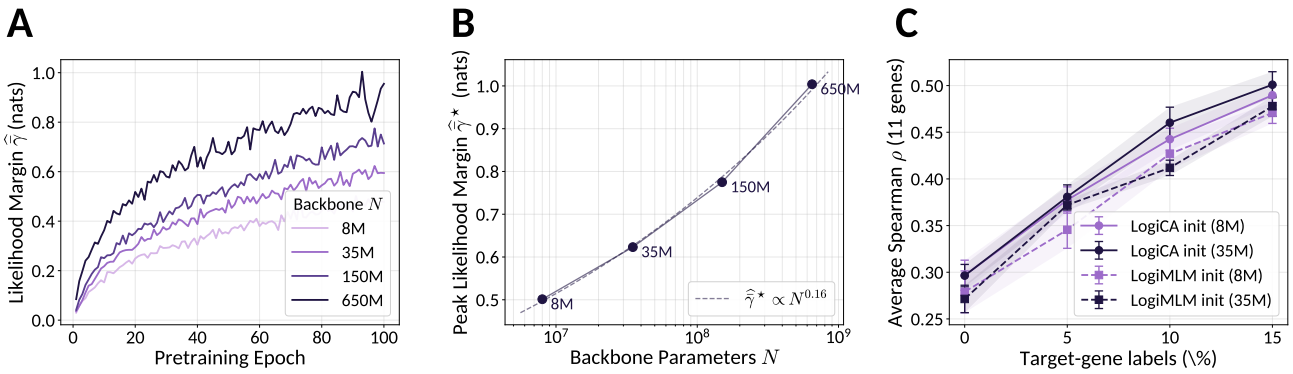


Figure 2. Two scaling regimes for protein–ligand LogiCA. (A) Held-out likelihood-margin trajectories during pretraining for ESM-2 backbones at $\{8, 35, 150, 650\}$M parameters. (B) Peak margin versus backbone size; dashed line shows a log-log fit, $\widehat{\gamma}^\star \propto N^{0.16}$. (C) Few-shot drug-resistance ranking as a fraction of target-gene labels is used for adaptation and the remaining variants are held out for evaluation.

paired data in which binding changes are often driven by a small number of residues at the interface (Banerjee et al., 2025). This makes the task a natural test bed for token-level likelihood scoring. We encode peptides with ESM-2 35M (Lin et al., 2023) and paired TCR CDR3α–CDR3β sequences with TCRLang (Raybould et al., 2024) (44.8M parameters), which is pretrained on paired TCR chains. We use the 35M ESM-2 peptide encoder because peptides are short linear amino-acid sequences, where larger protein backbones provide limited additional benefit and can overfit under scarce paired TCR–peptide supervision (Appendix F.4). We pretrain on experimentally validated TCR–peptide binders from IEDB (Vita et al., 2019), which predominantly provide CDR3β annotations, and supplement these data with paired CDR3α–CDR3β annotations curated in prior studies (Zhang et al., 2024a; Kwee et al., 2023). The resulting corpus contains approximately 250k training pairs, covering about 150k unique TCRs and 1.5k unique

Table 2. Drug-resistance variant scoring. Each cell is mean ± std across drugs (per gene aggregate). **Coelho (10g)** aggregates 10 leave-one-gene-out folds (KRAS, BRAF, MAP2K1/2, PIK3CA, AKT1, MYC, BCL2, PARP1/2) (Coelho et al., 2024); the cross-gene mean of pooled per-gene metrics is reported with cross-gene std. **Kim (EGFR)** (Kim et al., 2025) is a single-gene fold so ± is the cross-drug spread. Per-gene results and few-shot curves: Tables S12 and S10. Best per column is **bold**; second-best is underlined.

Method	Coelho (10g)		Kim (EGFR)	
	ρ	AUC	ρ	AUC
Contextualized backbone (fine-tuned)				
w/ LatentFuse (35M) concat embeddings + MLP	0.091 ± 0.082	0.547 ± 0.046	0.029 ± 0.021	0.519 ± 0.016
w/ LatentFuse (150M) concat embeddings + MLP	0.059 ± 0.061	0.536 ± 0.035	0.019 ± 0.067	0.511 ± 0.030
w/ LogiMLM (35M) logit-level MLM	0.214 ± 0.082	0.613 ± 0.047	0.245 ± 0.064	0.620 ± 0.031
w/ LogiMLM (150M) logit-level MLM	0.258 ± 0.060	0.632 ± 0.034	0.272 ± 0.083	0.633 ± 0.048
w/ LogiCA (35M) **token-score contrastive**	0.256 ± 0.054	0.636 ± 0.036	0.260 ± 0.067	0.630 ± 0.034
w/ LogiCA (150M) **token-score contrastive**	**0.271** ± 0.064	**0.644** ± 0.032	**0.295** ± 0.074	**0.638** ± 0.046
Unconditional baselines				
ESM-1v (Meier et al., 2021) masked LM	0.076 ± 0.079	0.540 ± 0.041	0.195 ± 0.088	0.601 ± 0.041
ESM-2 (Lin et al., 2023) (35M) masked LM	0.024 ± 0.078	0.516 ± 0.050	0.155 ± 0.094	0.577 ± 0.046
ESM-2 (Lin et al., 2023) (150M) masked LM	0.052 ± 0.055	0.532 ± 0.036	0.153 ± 0.105	0.580 ± 0.046
EVE (Frazer et al., 2021) MSA VAE	0.114 ± 0.155	0.568 ± 0.094	0.156 ± 0.081	0.580 ± 0.033
Tranception (Notin et al., 2022a) retrieval LM	0.041 ± 0.056	0.525 ± 0.027	0.170 ± 0.062	0.591 ± 0.031
Contextualized baselines (fine-tuned)				
DrugBAN (Bai et al., 2023) DTI classifier + MLP	0.014 ± 0.018	0.520 ± 0.015	0.018 ± 0.036	0.507 ± 0.025
Boltz-2 (Passaro et al., 2025) structure features + MLP	0.015 ± 0.033	0.506 ± 0.021	0.011 ± 0.047	0.506 ± 0.020
DrugCLIP (Jia et al., 2026) DTI contrastive + MLP	0.001 ± 0.034	0.505 ± 0.011	-0.000 ± 0.040	0.497 ± 0.019

peptides (Appendix D.3). Because reliable non-binding annotations remain difficult to define (Gao et al., 2023; Dens et al., 2023), LogiCA uses online synthetic negatives generated by random point mutations in binding pairs, reflecting the empirical prior that most local perturbations disrupt binding (92%, Figure S1C). Since CDR3β-only annotations outnumber paired-chain annotations by more than an order of magnitude, the final fine-tuning stage uses only paired CDR3α–CDR3β binding data. To prevent leakage, we remove any TCR–peptide pair appearing in downstream zero-shot splits from the pretraining corpus.

We evaluate zero-shot variant ranking on experimental TCR–peptide activity and binding-energy DMS studies (Banerjee et al., 2025), which assay single-residue substitutions in either the peptide or the TCR for a fixed TCR–peptide pair (Appendix D.3). These assays test whether LogiCA can rank functional near-neighbor variants without observing their experimental readouts. We also evaluate unseen-peptide generalization on IMMREP25 (Richardson et al., 2026); however, under the current level of paired TCR–peptide data scarcity, all existing sequence-based models remain close to random in this setting (Table S13). We therefore focus the main analysis on zero-shot variant ranking, where the mutation-local likelihood interface can be directly evaluated.

Zero-shot directional variant ranking. We evaluate two mutation directions against experimental TCR–peptide DMS measurements. In the peptide-variant setting, peptide mutants are ranked against a fixed TCR and evaluated against measurements from prior studies (Drost et al., 2025; Banerjee et al., 2025; Borrman et al., 2017). In the TCR-variant setting, TCR mutants are ranked against a fixed peptide and evaluated against binding energies in the ATLAS-TCR database (Borrman et al., 2017). In both cases, variants are scored directly with the mutation-local likelihood score in Eq. 3. For a controlled comparison, we keep the TCRLang–ESM-2 backbones and training data fixed, varying only the learning objective and anchor direction. We observe a clear directionality effect: TCR-anchored LogiCA (LogiCA-TCR) performs best for ranking peptide variants against a fixed TCR, whereas peptide-anchored LogiCA (LogiCA-Pep) performs best for ranking TCR variants against a fixed peptide (Table 3, Figures S5A–B). The dual-anchor model (LogiCA-Dual) provides the most balanced performance across both directions, making it preferable when mutations may occur on either side of the TCR–peptide interface. External paired TCR–peptide models provide stringent baselines, spanning latent-contrastive methods (Zhang et al., 2024a), classifier-based approaches (Springer et al., 2021; Moris et al., 2021; Zhang et al., 2023; Peng et al., 2023), and MLM-based models (Meynard-Piganeau et al., 2024; Karthikeyan et al., 2025). Across these baselines, LogiCA achieves the strongest correlations, suggesting that global embeddings and classifier scores do not transfer as effectively to mutation-local ranking as native-vocabulary likelihoods evaluated at the mutated positions.

The contrast with the conditional MLM ablation is especially pronounced in this setting. While LogiMLM remains near random on both the TCR and peptide-variant benchmarks, LogiCA ranks first on the ePytope binary mutation-classification benchmark (Drost et al., 2025), increasing AUC from 0.52 to 0.67 under the same zero-shot, mutation-local scoring protocol (Appendix F.5). This separation is larger than in the protein–ligand setting, consistent with the greater sparsity and imbalance of TCR–peptide supervision: when paired data are limited and dominated by dominated by overrepresented contexts (Figure S2), reconstruction alone provides a weak contextual matching signal, whereas contrastive alignment directly separates functional near-neighbors from disrupted pairs.

Table 3. TCR–epitope variant ranking. Columns are grouped by mutation direction: peptide variants under a fixed TCR and TCR variants under a fixed peptide. The blue block reports controlled TCRLang–ESM-2 variants; the gray block reports external paired TCR–epitope baselines. The † marker denotes methods that retain a native-vocabulary generative interface. Cells show mean Pearson or Spearman correlation ± standard deviation; best values are bolded and second-best values are underlined.

Method	Peptide mutations						TCR mutations	
	ePytope		BATCAVE		ATLAS-PEP		ATLAS-TCR	
	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman
Contextualized backbone								
w/ LogiMLM† continued MLM	0.039 ± 0.230	-0.040 ± 0.216	0.063 ± 0.227	0.000 ± 0.225	0.428 ± 0.438	0.527 ± 0.312	0.087 ± 0.515	-0.017 ± 0.457
w/ LogiCA-TCR† TCR-anchored token scoring	**0.296** ± 0.219	**0.229** ± 0.168	**0.223** ± 0.229	**0.170** ± 0.191	**0.632** ± 0.286	**0.731** ± 0.238	-0.017 ± 0.507	-0.115 ± 0.493
w/ LogiCA-Pep† peptide-anchored token scoring	0.176 ± 0.288	0.147 ± 0.313	0.144 ± 0.268	0.109 ± 0.294	0.398 ± 0.466	0.287 ± 0.468	0.131 ± 0.551	**0.287** ± 0.468
w/ LogiCA-Dual† dual-anchor token scoring	0.227 ± 0.333	0.180 ± 0.314	0.177 ± 0.310	0.132 ± 0.295	0.534 ± 0.425	0.654 ± 0.380	0.046 ± 0.476	0.188 ± 0.347
External baselines								
EPACT (Zhang et al., 2024a) embedding contrastive	0.016 ± 0.180	0.047 ± 0.211	0.037 ± 0.271	0.016 ± 0.220	0.357 ± 0.454	0.233 ± 0.465	0.179 ± 0.619	0.257 ± 0.528
ERGO-II (Springer et al., 2021) classifier head	-0.007 ± 0.201	-0.031 ± 0.173	-0.029 ± 0.173	-0.044 ± 0.169	0.182 ± 0.732	0.185 ± 0.581	0.128 ± 0.597	0.141 ± 0.623
ImRex (Moris et al., 2021) classifier head	0.089 ± 0.183	0.094 ± 0.182	0.089 ± 0.177	0.102 ± 0.173	0.535 ± 0.423	0.543 ± 0.351	**0.189** ± 0.410	0.162 ± 0.397
iTCep (Zhang et al., 2023) classifier head	0.094 ± 0.190	0.207 ± 0.219	0.070 ± 0.172	0.162 ± 0.222	0.219 ± 0.475	0.321 ± 0.333	0.026 ± 0.703	0.162 ± 0.397
TEIM (Peng et al., 2023) classifier head	0.051 ± 0.171	0.041 ± 0.161	0.076 ± 0.179	0.053 ± 0.166	0.225 ± 0.659	0.360 ± 0.563	-0.124 ± 0.288	-0.029 ± 0.326
TCR-T5† (Karthikeyan et al., 2025) generative MLM	0.028 ± 0.129	0.023 ± 0.139	0.032 ± 0.125	0.027 ± 0.148	0.047 ± 0.556	-0.031 ± 0.541	0.091 ± 0.360	0.098 ± 0.387
TULIP-TCR† (Meynard-Piganeau et al., 2024) generative MLM	0.162 ± 0.178	0.123 ± 0.167	0.160 ± 0.164	0.120 ± 0.160	0.607 ± 0.347	0.690 ± 0.235	-0.023 ± 0.641	0.034 ± 0.613

Dependency-map interpretation. Because LogiCA preserves token-probability outputs, it can be probed directly for residue-level statistical dependencies without introducing auxiliary attribution machinery. Specifically, Eq. 6 asks whether perturbing one residue changes the probability assigned to another residue. This is not intended as a state-of-the-art contact-prediction method; rather, it tests whether the model's native logits encode biologically meaningful inter-residue structure. Figures S5C–E illustrate one representative complex (PDB 5TEZ), where high dependency scores concentrate around the CDR3β–peptide interface and partially overlap the structural contact map. We evaluate this systematically using 250 crystallized TCR–pMHC structures from the TCR3d database (Gowthaman & Pierce, 2019; Lin et al., 2025), defining TCR–peptide contacts by an 5 Å heavy-atom distance threshold. Table S14 shows that LogiCA dependency scores are enriched for observed TCR–peptide contacts, with AUCs of ~ 0.59 for peptide–CDR3α contacts and ~ 0.74 for peptide–CDR3β contacts. These values are substantially higher than external logit-based baselines such as TCR-T5 (Karthikeyan et al., 2025) and TULIP-TCR (Meynard-Piganeau et al., 2024), indicating that LogiCA's probabilities carry interpretable structural signal.

Case study on NLVPMVATV optimization. As a representative zero-shot peptide-optimization case study, we analyze the CMV pp65 peptide NLVPMVATV, a clinically relevant HLA-A*02:01-restricted antigen widely used to monitor CMV-specific CD8 T-cell immunity in immunocompromised and transplant patients (Gratama et al., 2001). We use LogiCA to rank 171 experimentally measured single mutants of this peptide for recognition by the NLV3 TCR (Kula et al., 2019). Experimentally, only four variants exceed wild-type activity: L2I, L2V, A7P, and V3L. LogiCA ranks these variants 3rd, 8th, 29th, and 48th out of 171, respectively (Figure S5F). The recovery of L2I and L2V among the top-ranked candidates is particularly notable because both are conservative hydrophobic substitutions at position 2, a canonical HLA-A*02:01 anchor position. Such anchor-preserving variants are clinically relevant because they are expected to maintain efficient HLA-A*02:01 presentation while modulating TCR activation, making them plausible candidates for improved monitoring reagents or peptide agonists without necessarily disrupting antigen presentation (Kula et al., 2019).

4. Outlook

LogiCA offers a logit-space view of multimodal alignment for biological language models. Rather than forcing embeddings from different modalities into a shared representation space, cross-modal context modulates each backbone's na-

tive token distribution. Alignment is therefore expressed not as geometric proximity between pooled latents, but as changes in context-conditioned token likelihoods. This preserves the probabilistic interface that makes pretrained biological language models useful for scoring, interpretation, and generation, while making that interface sensitive to biological context.

This distinction is important because many biological questions are naturally token-local: a ligand may change the plausibility of a residue substitution, a TCR may change the likelihood of an peptide mutation, or a peptide residue may induce localized changes in a receptor sequence. By retaining pretrained token heads and injecting context through native-head-preserving cross-modal adapters, LOGICA keeps these questions in the output space where pretrained models already encode sequence semantics. Across protein–ligand binding and TCR–pMHC modeling, this logit-space formulation matches or improves on strong latent-alignment alternatives for pairwise prediction, with its largest gains in variant-ranking settings where native token likelihoods directly score local sequence changes under biological context.

This token-likelihood-preserving design comes with a computational tradeoff: because sequence–context scores require pair-specific contextualization, LOGICA is less efficient than dual-encoder models for exhaustive large-scale retrieval. It is therefore best viewed as a reranking, variant-scoring, or generative interaction model rather than a replacement for first-stage embedding retrieval in massive screening settings (Appendix G).

More broadly, biological foundation models need not be adapted by replacing their learned vocabularies with task-specific classifiers. External biological context can instead be trained to reshape the token probabilities of existing models, making the framework modular across protein, immune-receptor, peptide, small-molecule, DNA/RNA, genomic-state, microbial, and synthetic-biology backbones (Dalla-Torre et al., 2025; Penić et al., 2025; Ji et al., 2021; Avsec et al., 2021; Gao et al., 2024; Avsec et al., 2026; Zvyagin et al., 2023; Wiatrak et al., 2025). This is especially relevant in biological settings where paired cross-modal data are sparse: contrastive supervision can extract contextual signal from matched and mismatched pairs while preserving each backbone's native likelihood interface.

Finally, preserving native logits creates a direct path to conditional generation. The contextualized token distributions produced by LOGICA can be used as sampling distributions in Gibbs-style procedures over protein, receptor, peptide, or molecular tokens (Appendix B.3). We therefore view LOGICA less as a single architecture than as a general principle for aligning biological language models: preserve their token-level probabilistic semantics, and make those semantics context-sensitive.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024. doi: 10.1038/s41586-024-07487-w. URL https://doi.org/10.1038/s41586-024-07487-w.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. doi: 10.48550/arXiv.2204.14198. URL https://arxiv.org/abs/2204.14198.

Ashuach, T., Reidenbach, D. A., Gayoso, A., and Yosef, N. MultiVI: deep generative model for the integration of multimodal data. *Nature Methods*, 20:1222–1231, 2023. doi: 10.1038/s41592-023-01909-9. URL https://doi.org/10.1038/s41592-023-01909-9.

Avsec, Ž., Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., Assael, Y., Jumper, J., Kohli, P., and Kelley, D. R. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021. doi: 10.1038/s41592-021-01252-x. URL https://doi.org/10.1038/s41592-021-01252-x.

Avsec, Ž., Latysheva, N., Cheng, J., Novati, G., Taylor, K. R., Ward, T., Bycroft, C., Nicolaisen, L., Arvaniti, E., Pan, J., Thomas, R., Dutordoir, V., Perino, M., De, S., Karollus, A., Gayoso, A., Sargeant, T., Mottram, A., Wong, L. H., Drotár, P., Kosiorek, A., Senior, A., Tanburn, R., Applebaum, T., Basu, S., Hassabis, D., and Kohli, P. Advancing regulatory variant effect prediction with AlphaGenome. *Nature*, 649:1206–1218,

2026. doi: 10.1038/s41586-025-10014-0. URL https://doi.org/10.1038/s41586-025-10014-0.

Bai, P., Miljković, F., John, B., and Lu, H. Interpretable bilinear attention network with domain adaptation improves drug–target prediction. *Nature Machine Intelligence*, 5(2):126–136, 2023. doi: 10.1038/s42256-022-00605-1. URL https://doi.org/10.1038/s42256-022-00605-1.

Banerjee, A., Pattinson, D. J., Wincek, C. L., Bunk, P., Axhemi, A., Chapin, S. R., Navlakha, S., and Meyer, H. V. T cell receptor cross-reactivity prediction improved by a comprehensive mutational scan database. *Cell Systems*, 16(8):101345, 2025. doi: 10.1016/j.cels.2025.101345. URL https://doi.org/10.1016/j.cels.2025.101345.

Borrman, T., Cimons, J., Cosiano, M., Purcaro, M., Pierce, B. G., Baker, B. M., and Weng, Z. ATLAS: a database linking binding affinities with structures for wild-type and mutant TCR–pMHC complexes. *Proteins: Structure, Function, and Bioinformatics*, 85(5):908–916, 2017. doi: 10.1002/prot.25260. URL https://doi.org/10.1002/prot.25260.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. doi: 10.2307/2334029. URL https://doi.org/10.2307/2334029.

Brbić, M., Cao, K., Hickey, J. W., Tan, Y., Snyder, M. P., Nolan, G. P., and Leskovec, J. Annotation of spatially resolved single-cell data with STELLAR. *Nature Methods*, 19:1411–1418, 2022. doi: 10.1038/s41592-022-01651-8. URL https://doi.org/10.1038/s41592-022-01651-8.

Brixi, G., Durrant, M. G., Ku, J., Naghipourfar, M., Poli, M., Sun, G., Brockman, G., Chang, D., Fanton, A., Gonzalez, G. A., King, S. H., Li, D. B., Merchant, A. T., Nguyen, E., Ricci-Tam, C., Romero, D. W., Schmok, J. C., Taghibakhshi, A., Vorontsov, A., Yang, B., Deng, M., Gorton, L., Nguyen, N., Wang, N. K., Pearce, M. T., Simon, E., Adams, E., Amador, Z. J., Ashley, E. A., Baccus, S. A., Dai, H., Dillmann, S., Ermon, S., Guo, D., Herschl, M. H., Ilango, R., Janik, K., Lu, A. X., Mehta, R., Mofrad, M. R. K., Ng, M. Y., Pannu, J., Ré, C., St. John, J., Sullivan, J., Tey, J., Viggiano, B., Zhu, K., Zynda, G., Balsam, D., Collison, P., Costa, A. B., Hernandez-Boussard, T., Ho, E., Liu, M.-Y., McGrath, T., Powell, K., Pinglay, S., Burke, D. P., Goodarzi, H., Hsu, P. D., and Hie, B. L. Genome modelling and design across all domains of life with Evo 2. *Nature*, pp. 1–13, 2026. doi: 10.1038/s41586-026-10176-5. URL https://doi.org/10.1038/s41586-026-10176-5.

Burbach, S. M. and Briney, B. Improving antibody language models with native pairing. *Patterns*, 5(5):100967, 2024. doi: 10.1016/j.patter.2024.100967. URL https://doi.org/10.1016/j.patter.2024.100967.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pp. 89–96. Association for Computing Machinery, 2005. doi: 10.1145/1102351.1102363. URL https://doi.org/10.1145/1102351.1102363.

Chen, L. T., Quinn, Z., Dumas, M., Peng, C., Hong, L., Lopez-Gonzalez, M., Mestre, A., Watson, R., Vincoff, S., Zhao, L., Wu, J., Stavrand, A., Schaepers-Cheu, M., Wang, T. Z., Srijay, D., Monticello, C., Vure, P., Pulugurta, R., Pertsemlidis, S., Kholina, K., Goel, S., DeLisa, M. P., Chi, J.-T. A., Truant, R., Aguilar, H. C., and Chatterjee, P. Target sequence-conditioned design of peptide binders using masked language modeling. *Nature Biotechnology*, 2025. doi: 10.1038/s41587-025-02761-2. URL https://doi.org/10.1038/s41587-025-02761-2.

Chithrananda, S., Grand, G., and Ramsundar, B. ChemBERTa: Large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020. doi: 10.48550/arXiv.2010.09885. URL https://arxiv.org/abs/2010.09885.

Coelho, M. A., Strauss, M. E., Watterson, A., Cooper, S., Bhosle, S., Illuzzi, G., Karakoc, E., Dinçer, C., Vieira, S. F., Sharma, M., Moullet, M., Conticelli, D., Koeppel, J., McCarten, K., Cattaneo, C. M., Veninga, V., Picco, G., Parts, L., Forment, J. V., Voest, E. E., Marioni, J. C., Bassett, A., and Garnett, M. J. Base editing screens define the genetic landscape of cancer drug resistance mechanisms. *Nature Genetics*, 56(11):2479–2492, 2024. doi: 10.1038/s41588-024-01948-8. URL https://doi.org/10.1038/s41588-024-01948-8.

Cornman, A., West-Roberts, J., Camargo, A. P., Roux, S., Beracochea, M., Mirdita, M., Ovchinnikov, S., and Hwang, Y. The OMG dataset: An open MetaGenomic corpus for mixed-modality genomic language modeling. *bioRxiv*, 2024. doi: 10.1101/2024.08.14.607850. URL https://doi.org/10.1101/2024.08.14.607850.

Cui, H., Tejada-Lapuerta, A., Brbić, M., Saez-Rodriguez, J., Cristea, S., Goodarzi, H., Lotfollahi, M., Theis, F. J., and Wang, B. Towards multimodal foundation models in molecular cell biology. *Nature*, 640(8059):623–633, 2025. doi: 10.1038/s41586-025-08710-y. URL https://doi.org/10.1038/s41586-025-08710-y.

Dalla-Torre, H., Gonzalez, L., Mendoza-Revilla, J., Lopez Carranza, N., Grzywaczewski, A. H., Oteri, F., Dallago, C., Trop, E., de Almeida, B. P., Sirelkhatim, H., Richard, G., Skwark, M., Beguir, K., Lopez, M., and Pierrot, T. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025. doi: 10.1038/s41592-024-02523-z. URL https://doi.org/10.1038/s41592-024-02523-z.

Davis, M. I., Hunt, J. P., Herrgard, S., Ciceri, P., Wodicka, L. M., Pallares, G., Hocker, M., Treiber, D. K., and Zarrinkar, P. P. Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology*, 29 (11):1046–1051, 2011. doi: 10.1038/nbt.1990. URL https://doi.org/10.1038/nbt.1990.

Dens, C., Laukens, K., Bittremieux, W., and Meysman, P. The pitfalls of negative data bias for the T-cell epitope specificity challenge. *Nature Machine Intelligence*, 2023. doi: 10.1038/s42256-023-00727-0. URL https://doi.org/10.1038/s42256-023-00727-0.

Deutschmann, N., Pelissier, A., Weber, A., Gao, S., Bogojeska, J., and Martínez, M. R. Do domain-specific protein language models outperform general models on immunology-related tasks? *ImmunoInformatics*, 14:100036, 2024. doi: 10.1016/j.immuno.2024.100036. URL https://doi.org/10.1016/j.immuno.2024.100036.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019. doi: 10.18653/v1/N19-1423. URL https://arxiv.org/abs/1810.04805.

Drost, F., Chernysheva, A., Albahah, M., Kocher, K., Schober, K., and Schubert, B. Benchmarking of T cell receptor–epitope predictors with ePytope-TCR. *Cell Genomics*, 5(8):100946, 2025. doi: 10.1016/j.xgen.2025.100946. URL https://doi.org/10.1016/j.xgen.2025.100946.

Fabian, B., Edlich, T., Gaspar, H., Segler, M., Meyers, J., Fiscato, M., and Ahmed, M. Molecular representation learning with language models and domain-relevant auxiliary tasks. *arXiv preprint arXiv:2011.13230*, 2020. doi: 10.48550/arXiv.2011.13230. URL https://arxiv.org/abs/2011.13230.

Frazer, J., Notin, P., Dias, M., Gomez, A., Min, J. K., Brock, K., Gal, Y., and Marks, D. S. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021. doi: 10.1038/s41586-021-04043-8. URL https://doi.org/10.1038/s41586-021-04043-8.

Gao, Y., Gao, Y., Fan, Y., Zhu, C., Wei, Z., Zhou, C., Chuai, G., Chen, Q., Zhang, H., and Liu, Q. Pan-peptide meta learning for T-cell receptor–antigen binding recognition. *Nature Machine Intelligence*, 2023. doi: 10.1038/s42256-023-00619-3. URL https://doi.org/10.1038/s42256-023-00619-3.

Gao, Z., Liu, Q., Zeng, W., Jiang, R., and Wong, W. H. Epigept: a pretrained transformer-based language model for context-specific human epigenomics. *Genome Biology*, 25(1):1–30, 2024. doi: 10.1186/s13059-024-03449-7. URL https://doi.org/10.1186/s13059-024-03449-7.

Garau-Luis, J. J., Bordes, P., Gonzalez, L., Roller, M., de Almeida, B. P., Hexemer, L., Blum, C., Laurent, S., Grzegorzewski, J., Lang, M., Pierrot, T., and Richard, G. Multi-modal transfer learning between biological foundation models. *Advances in Neural Information Processing Systems*, 37:78431–78450, 2024. doi: 10.48550/arXiv.2406.14150. URL https://arxiv.org/abs/2406.14150.

Gayoso, A., Steier, Z., Lopez, R., Regier, J., Nazor, K. L., Streets, A., and Yosef, N. Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nature Methods*, 18:272–282, 2021. doi: 10.1038/s41592-020-01050-x. URL https://doi.org/10.1038/s41592-020-01050-x.

Gong, C., Klivans, A. R., Loy, J. M., Chen, T., Liu, Q., and Diaz, D. J. Evolution-inspired loss functions for protein representation learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15893–15906. PMLR, 2024. URL https://proceedings.mlr.press/v235/gong24e.html.

Gowthaman, R. and Pierce, B. G. TCR3d: The T cell receptor structural repertoire database. *Bioinformatics*, 35(24):5323–5325, 2019. doi: 10.1093/bioinformatics/btz517. URL https://doi.org/10.1093/bioinformatics/btz517.

Gratama, J. W., van Esser, J. W., Lamers, C. H., Tournay, C., Lowenberg, B., Bolhuis, R. L., and Cornelissen, J. J. Tetramer-based quantification of cytomegalovirus (CMV)–specific CD8+ T lymphocytes in T-cell–depleted stem cell grafts and after transplantation may identify patients at risk for progressive CMV infection. *Blood, The Journal of the American Society of Hematology*, 98 (5):1358–1364, 2001. doi: 10.1182/blood.v98.5.1358.

URL https://doi.org/10.1182/blood.v98.5.1358.

Hawkins-Hooker, A., Kmec, J., Bent, O., and Duckworth, P. Likelihood-based fine-tuning of protein language models for few-shot fitness prediction and design. In *ICML 2024 Workshop on Accessible and Efficient Foundation Models for Biological Discovery*, 2024. doi: 10.1101/2024.05.28.596156. URL https://doi.org/10.1101/2024.05.28.596156.

Hejna, J., Rafailov, R., Sikchi, H., Finn, C., Niekum, S., Knox, W. B., and Sadigh, D. Contrastive preference learning: Learning from human feedback without reinforcement learning. In *The Twelfth International Conference on Learning Representations*. doi: 10.48550/arXiv.2310.13639. URL https://arxiv.org/abs/2310.13639.

Hie, B. L., Yang, K. K., and Kim, P. S. Evolutionary velocity with protein language models predicts evolutionary dynamics of diverse proteins. *Cell Systems*, 13(4):274–285, 2022. doi: 10.1016/j.cels.2022.01.003. URL https://doi.org/10.1016/j.cels.2022.01.003.

Hie, B. L., Shanker, V. R., Xu, D., Bruun, T. U. J., Weidenbacher, P. A., Tang, S., Wu, W., Pak, J. E., and Kim, P. S. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2):275–283, 2024. doi: 10.1038/s41587-023-01763-2. URL https://doi.org/10.1038/s41587-023-01763-2.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*, 2022. doi: 10.48550/arXiv.2106.09685. URL https://arxiv.org/abs/2106.09685.

Huang, K., Xiao, C., Glass, L. M., and Sun, J. MolTrans: Molecular interaction transformer for drug–target interaction prediction. *Bioinformatics*, 37(6):830–836, 2021. doi: 10.1093/bioinformatics/btaa880. URL https://doi.org/10.1093/bioinformatics/btaa880.

Hwang, Y., Cornman, A. L., Kellogg, E. H., Ovchinnikov, S., and Girguis, P. R. Genomic language model predicts protein co-regulation and function. *Nature Communications*, 15(1):2880, 2024. doi: 10.1038/s41467-024-46947-9. URL https://doi.org/10.1038/s41467-024-46947-9.

Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021. doi: 10.1093/bioinformatics/btab083. URL https://doi.org/10.1093/bioinformatics/btab083.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021. doi: 10.48550/arXiv.2102.05918. URL https://arxiv.org/abs/2102.05918.

Jia, Y., Gao, B., Tan, J., Zheng, J., Hong, X., Zhu, W., Tan, H., Xiao, Y., Tan, L., Cai, H., Huang, Y., Deng, Z., Wu, X., Jin, Y., Yuan, Y., Tian, J., He, W., Ma, W., Zhang, Y., Liu, L., Yan, C., Zhang, W., and Lan, Y. Deep contrastive learning enables genome-wide virtual screening. *Science*, 391(6781):eads9530, 2026. doi: 10.1126/science.ads9530. URL https://doi.org/10.1126/science.ads9530.

Johnson, S. R., Monaco, S., Massie, K., and Syed, Z. Generating novel protein sequences using gibbs sampling of masked language models. *bioRxiv*, pp. 2021–01, 2021. doi: 10.1101/2021.01.26.428322. URL https://doi.org/10.1101/2021.01.26.428322.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. doi: 10.48550/arXiv.2001.08361. URL https://arxiv.org/abs/2001.08361.

Karthikeyan, D., Bennett, S. N., Reynolds, A. G., Vincent, B. G., and Rubinsteyn, A. Conditional generation of real antigen-specific T cell receptor sequences. *Nature Machine Intelligence*, 7(9):1494–1509, 2025. doi: 10.1038/s42256-025-01096-6. URL https://doi.org/10.1038/s42256-025-01096-6.

Kim, Y., Oh, H.-C., Lee, S., and Kim, H. H. Saturation profiling of drug-resistant genetic variants using prime editing. *Nature Biotechnology*, 43(9):1471–1484, 2025. doi: 10.1038/s41587-024-02465-z. URL https://doi.org/10.1038/s41587-024-02465-z.

Krenn, M., Häse, F., Nigam, A., Friederich, P., and Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020. doi: 10.1088/2632-2153/aba947. URL https://doi.org/10.1088/2632-2153/aba947.

Kula, T., Dezfulian, M. H., Wang, C. I., Abdelfattah, N. S., Hartman, Z. C., Wucherpfennig, K. W., Lyerly, H. K.,

and Elledge, S. J. T-Scan: a genome-wide method for the systematic discovery of T cell epitopes. *Cell*, 178(4):1016–1028, 2019. doi: 10.1016/j.cell.2019.07.009. URL https://doi.org/10.1016/j.cell.2019.07.009.

Kwee, B. P. Y., Messemaker, M., Marcus, E., Oliveira, G., Scheper, W., Wu, C. J., Teuwen, J., and Schumacher, T. N. STAPLER: Efficient learning of TCR-peptide specificity prediction from full-length TCR-peptide data. *bioRxiv*, pp. 2023.04.25.538237, 2023. doi: 10.1101/2023.04.25.538237. URL https://doi.org/10.1101/2023.04.25.538237.

Lee, M., Lee, K., and Shin, J. Fine-tuning protein language models by ranking protein fitness. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023. URL https://openreview.net/forum?id=DUjUJCqqA7.

Li, H., Zhao, D., and Zeng, J. Kpgt: knowledge-guided pre-training of graph transformer for molecular property prediction. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 857–867, 2022. doi: 10.1145/3534678.3539426. URL https://doi.org/10.1145/3534678.3539426.

Li, M. M., Huang, Y., Sumathipala, M., Liang, M. Q., Valdeolivas, A., Ananthakrishnan, A. N., Liao, K., Marbach, D., and Zitnik, M. Contextual AI models for single-cell protein biology. *Nature Methods*, 21(8):1546–1557, 2024. doi: 10.1038/s41592-024-02341-3. URL https://doi.org/10.1038/s41592-024-02341-3.

Lin, V., Cheung, M., Gowthaman, R., Eisenberg, M., Baker, B. M., and Pierce, B. G. TCR3d 2.0: expanding the T cell receptor structure database with new structures, tools and interactions. *Nucleic Acids Research*, 53(D1): D604–D608, 2025. doi: 10.1093/nar/gkae840. URL https://doi.org/10.1093/nar/gkae840.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., Dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. 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://doi.org/10.1126/science.ade2574.

Liu, D., Young, F., Lamb, K. D., Quiros, A. C., Pancheva, A., Miller, C. J., Macdonald, C., Robertson, D. L., and Yuan, K. PLM-interact: extending protein language models to predict protein-protein interactions. *Nature Communications*, 16(1):9012, 2025a. doi: 10.1038/s41467-025-64512-w. URL https://doi.org/10.1038/s41467-025-64512-w.

Liu, S., Liu, Y., Xu, H., Xia, J., and Li, S. Z. SP-DTI: subpocket-informed transformer for drug–target interaction prediction. *Bioinformatics*, 41(3):btaf011, 03 2025b. ISSN 1367-4811. doi: 10.1093/bioinformatics/btaf011. URL https://doi.org/10.1093/bioinformatics/btaf011.

Liu, T., Lin, Y., Wen, X., Jorissen, R. N., and Gilson, M. K. BindingDB in 2024: a FAIR knowledgebase of protein-small molecule binding data. *Nucleic Acids Research*, 53(D1):D1633–D1644, 2025c. doi: 10.1093/nar/gkae1075. URL https://doi.org/10.1093/nar/gkae1075.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. doi: 10.48550/arXiv.1907.11692. URL https://arxiv.org/abs/1907.11692.

Lupo, U., Sgarbossa, D., and Bitbol, A.-F. Pairing interacting protein sequences using masked language modeling. *Proceedings of the National Academy of Sciences*, 121(27):e2311887121, 2024. doi: 10.1073/pnas.2311887121. URL https://doi.org/10.1073/pnas.2311887121.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos Jr, J. L., Xiong, C., Sun, Z. Z., Socher, R., Fraser, J. S., and Naik, N. Large language models generate functional protein sequences across diverse families. *Nature biotechnology*, 41(8):1099–1106, 2023. doi: 10.1038/s41587-022-01618-2. URL https://doi.org/10.1038/s41587-022-01618-2.

McCarter, C., Bhattacharya, N., Ober, S. W., and Elliott, H. How to make the most of your masked language model for protein engineering. *arXiv preprint arXiv:2603.10302*, 2026. doi: 10.48550/arXiv.2603.10302. URL https://arxiv.org/abs/2603.10302.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29287–29303, 2021. doi: 10.1101/2021.07.09.450648. URL https://doi.org/10.1101/2021.07.09.450648.

Meynard-Piganeau, B., Feinauer, C., Weigt, M., Walczak, A. M., and Mora, T. TULIP: A transformer-based unsupervised language model for interacting peptides

and T cell receptors that generalizes to unseen epitopes. *Proceedings of the National Academy of Sciences*, 121(24):e2316401121, 2024. doi: 10.1073/pnas.2316401121. URL https://doi.org/10.1073/pnas.2316401121.

Mizrahi, D., Bachmann, R., Kar, O. F., Yeo, T., Gao, M., Dehghan, A., and Zamir, A. 4M: Massively multimodal masked modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 58363–58408, 2023. URL https://arxiv.org/abs/2312.06647.

Montemurro, A., Schuster, V., Povlsen, H. R., Bentzen, A. K., Jurtz, V., Chronister, W. D., Crinklaw, A., Hadrup, S. R., Winther, O., Peters, B., Jessen, L. E., and Nielsen, M. NetTCR-2.0 enables accurate prediction of TCR–peptide binding by using paired TCRα and β sequence data. *Communications Biology*, 4(1):1060, 2021. doi: 10.1038/s42003-021-02610-3. URL https://doi.org/10.1038/s42003-021-02610-3.

Moris, P., De Pauw, J., Postovskaya, A., Gielis, S., De Neuter, N., Bittremieux, W., Ogunjimi, B., Laukens, K., and Meysman, P. Current challenges for unseen-epitope TCR interaction prediction and a new perspective derived from image classification. *Briefings in Bioinformatics*, 22(4):bbaa318, 2021. doi: 10.1093/bib/bbaa318. URL https://doi.org/10.1093/bib/bbaa318.

Nagano, Y., Pyo, A. G. T., Milighetti, M., Henderson, J., Shawe-Taylor, J., Chain, B., and Tiffeau-Mayer, A. Contrastive learning of T cell receptor representations. *Cell Systems*, 16(1):101165, 2025. doi: 10.1016/j.cels.2024.12.006. URL https://doi.org/10.1016/j.cels.2024.12.006.

Notin, P., Dias, M., Frazer, J., Marchena-Hurtado, J., Gomez, A. N., Marks, D., and Gal, Y. Tranception: Protein fitness prediction with autoregressive transformers and inference-time retrieval. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16990–17017. PMLR, 2022a. doi: 10.48550/arXiv.2205.13760. URL https://arxiv.org/abs/2205.13760.

Notin, P., Van Niekerk, L., Kollasch, A. W., Ritter, D., Gal, Y., and Marks, D. S. TranceptEVE: Combining family-specific and family-agnostic models of protein sequences for improved fitness prediction. In *NeurIPS 2022 Workshop on Learning Meaningful Representations of Life (LMRL)*, 2022b. doi: 10.1101/2022.12.07.519495. URL https://doi.org/10.1101/2022.12.07.519495.

Notin, P., Kollasch, A. W., Ritter, D., Van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Gal, Y., and Marks, D. S. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, volume 36, 2023. doi: 10.52202/075280-2810. URL https://doi.org/10.52202/075280-2810.

Olsen, T. H., Moal, I. H., and Deane, C. M. Addressing the antibody germline bias and its effect on language models for improved antibody design. *Bioinformatics*, 40(11):btae618, 2024. doi: 10.1093/bioinformatics/btae618. URL https://doi.org/10.1093/bioinformatics/btae618.

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., Kwabi-Addo, D., Beaini, D., Jaakkola, T., and Barzilay, R. Boltz-2: Towards accurate and efficient binding affinity prediction, 2025. URL https://doi.org/10.1101/2025.06.14.659707. bioRxiv preprint.

Peng, X., Lei, Y., Feng, P., Jia, L., Ma, J., Zhao, D., and Zeng, J. Characterizing the interaction conformation between T-cell receptors and epitopes with deep learning. *Nature Machine Intelligence*, 5(4):395–407, 2023. doi: 10.1038/s42256-023-00634-4. URL https://doi.org/10.1038/s42256-023-00634-4.

Penić, R. J., Vlašić, T., Huber, R. G., Wan, Y., and Šikić, M. Rinalmo: General-purpose rna language models can generalize well on structure prediction tasks. *Nature Communications*, 16(1):5671, 2025. doi: 10.1038/s41467-025-60872-5. URL https://doi.org/10.1038/s41467-025-60872-5.

Pugh, C. W. J., Núñez-Valencia, P. G., Dias, M., and Frazer, J. From likelihood to fitness: Improving variant effect prediction in protein and genome language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.1101/2025.05.20.655154. URL https://doi.org/10.1101/2025.05.20.655154.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. doi: 10.48550/arXiv.2103.00020. URL https://arxiv.org/abs/2103.00020.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023. doi: 10.52202/075280-2338. URL https://arxiv.org/abs/2305.18290.

Raybould, M. I. J., Greenshields-Watson, A., Agarwal, P., Aguilar-Sanjuan, B., Olsen, T. H., Turnbull, O. M., Quast, N. P., and Deane, C. M. The observed T cell receptor space database enables paired-chain repertoire mining, coherence analysis, and language modeling. *Cell Reports*, 43(9):114704, 2024. doi: 10.1016/j.celrep.2024.114704. URL https://doi.org/10.1016/j.celrep.2024.114704.

Richardson, E., Aarts, Y. J. M., Altin, J. A., Baakman, C. A. B., Bradley, P., Chen, B., Clifford, J., Dhar, M., Diepenbroek, D., Fast, E., Gowthaman, R., He, J., Karnaukhov, V., Marzella, D. F., Meysman, P., Nielsen, M., Nilsson, J. B., Deleuran, S. N., Parizi, F. M., Pelissier, A., Pierce, B. G., Rodriguez Martinez, M., Roran A R, D., Saravanakumar, S., Shao, Y., Smit, N., Van Houcke, M., Visani, G. M., Wan, Y.-T. R., Wang, X., Woods, L., Wuyts, S., Xiao, C., Xue, L. C., Barton, J., Noakes, M., May, D. H., and Peters, B. Immrep25: Unseen peptides. *bioRxiv*, pp. 2026–03, 2026. doi: 10.64898/2026.03.30.715276. URL https://doi.org/10.64898/2026.03.30.715276.

Riesselman, A. J., Ingraham, J. B., and Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018. doi: 10.1038/s41592-018-0138-4. URL https://doi.org/10.1038/s41592-018-0138-4.

Rosen, Y., Brbić, M., Roohani, Y., Swanson, K., Li, Z., and Leskovec, J. Toward universal cell embeddings: integrating single-cell RNA-seq datasets across species with SATURN. *Nature Methods*, 21:1492–1500, 2024. doi: 10.1038/s41592-024-02191-z. URL https://doi.org/10.1038/s41592-024-02191-z.

Ruffolo, J. A., Gray, J. J., and Sulam, J. Deciphering antibody affinity maturation with language models and weakly supervised learning. *arXiv preprint arXiv:2112.07782*, 2021. doi: 10.48550/arXiv.2112.07782. URL https://arxiv.org/abs/2112.07782.

Shoshan, Y., Raboh, M., Ozery-Flato, M., Ratner, V., Golts, A., Weber, J. K., Barkan, E., Rabinovici-Cohen, S., Polaczek, S., Amos, I., et al. MAMMAL – molecular aligned multi-modal architecture and language for biomedical discovery. *npj Drug Discovery*, 2026. doi: 10.1038/s44386-026-00047-4. URL https://doi.org/10.1038/s44386-026-00047-4.

Shugay, M., Bagaev, D. V., Zvyagin, I. V., Vroomans, R. M., Crawford, J. C., Dolton, G., Komech, E. A., Sycheva, A. L., Koneva, A. E., Egorov, E. S., Eliseev, A. V., Van Dyk, E., Dash, P., Attaf, M., Rius, C., Ladell, K., McLaren, J. E., Matthews, K. K., Clemens, E. B., Douek, D. C., Luciani, F., van Baarle, D., Kedzierska, K., Kesmir, C., Thomas, P. G., Price, D. A., Sewell, A. K., and Chudakov, D. M. VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Research*, 46(D1):D419–D427, 2018. doi: 10.1093/nar/gkx760. URL https://doi.org/10.1093/nar/gkx760.

Singh, R., Sledzieski, S., Bryson, B., Cowen, L., and Berger, B. Contrastive learning in protein language space predicts interactions between drugs and protein targets. *Proceedings of the National Academy of Sciences*, 120(24):e2220778120, 2023. doi: 10.1073/pnas.2220778120. URL https://doi.org/10.1073/pnas.2220778120.

Springer, I., Besser, H., Tickotsky-Moskovitz, N., Dvorkin, S., and Louzoun, Y. Prediction of specific TCR–peptide binding from large dictionaries of TCR–peptide pairs. *Frontiers in Immunology*, 11:1803, 2020. doi: 10.3389/fimmu.2020.01803. URL https://doi.org/10.3389/fimmu.2020.01803.

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

Su, J., Zhou, X., Zhang, X., and Yuan, F. A trimodal protein language model enables advanced protein searches. *Nature Biotechnology*, pp. 1–7, 2025. doi: 10.1038/s41587-025-02836-0. URL https://doi.org/10.1038/s41587-025-02836-0.

Sun, J., Liu, S., Su, Z., Zhong, X., Jiang, P., Jin, B., Li, P., Shi, W., and Han, J. GRACE: Generative representation learning via contrastive policy optimization. In *International Conference on Learning Representations*, 2026. doi: 10.48550/arXiv.2510.04506. URL https://arxiv.org/abs/2510.04506.

Tan, H. and Bansal, M. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 5100–5111, 2019. doi: 10.18653/v1/

D19-1514. URL https://arxiv.org/abs/1908.07490.

Tickotsky, N., Sagiv, T., Prilusky, J., Shifrut, E., and Friedman, N. McPAS-TCR: a manually curated catalogue of pathology-associated T cell receptor sequences. *Bioinformatics*, 33(18):2924–2929, 2017. doi: 10.1093/bioinformatics/btx286. URL https://doi.org/10.1093/bioinformatics/btx286.

Tomaz da Silva, P., Karollus, A., Hingerl, J., Galindez, G., Wagner, N., Hernandez-Alias, X., Incarnato, D., and Gagneur, J. Nucleotide dependency analysis of genomic language models detects functional elements. *Nature Genetics*, 57:2589–2602, 2025. doi: 10.1038/s41588-025-02347-3. URL https://doi.org/10.1038/s41588-025-02347-3.

Ullanat, V., Jing, B., Sledzieski, S., and Berger, B. Learning the language of protein-protein interactions. *Nature Communications*, 17:1199, 2026. doi: 10.1038/s41467-025-67971-3. URL https://doi.org/10.1038/s41467-025-67971-3.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. doi: 10.48550/arXiv.1807.03748. URL https://arxiv.org/abs/1807.03748.

Vita, R., Mahajan, S., Overton, J. A., Dhanda, S. K., Martini, S., Cantrell, J. R., Wheeler, D. K., Sette, A., and Peters, B. The Immune Epitope Database (IEDB): 2018 update. *Nucleic Acids Research*, 47(D1):D339–D343, 2019. doi: 10.1093/nar/gky1006. URL https://doi.org/10.1093/nar/gky1006.

Wang, Z., Wang, Z., Srinivasan, B., Ioannidis, V. N., Rangwala, H., and Anubhai, R. BioBridge: Bridging biomedical foundation models via knowledge graphs. In *International Conference on Learning Representations (ICLR)*, 2024. doi: 10.48550/arXiv.2310.03320. URL https://arxiv.org/abs/2310.03320.

Weber, A., Born, J., and Rodríguez Martínez, M. TITAN: T-cell receptor specificity prediction with bimodal attention networks. *Bioinformatics*, 37(Supplement_1): i237–i244, 2021. doi: 10.1093/bioinformatics/btab294. URL https://doi.org/10.1093/bioinformatics/btab294.

Wiatrak, M., Vinas Torne, R., Ntemourtsidou, M., Dinan, A. M., Abelson, D. C., Arora, D., Brbic, M., Weimann, A., and Floto, R. A. A contextualised protein language model reveals the functional syntax of bacterial evolution. *bioRxiv*, 2025. doi: 10.1101/2025.07.20.665723. URL https://doi.org/10.1101/2025.07.20.665723.

Wu, W., Li, Q., Li, M., Fu, K., Feng, F., Ye, J., Xiong, H., and Wang, Z. Generator: A long-context generative genomic foundation model. *arXiv preprint arXiv:2502.07272*, 2025. doi: 10.48550/arXiv.2502.07272. URL https://arxiv.org/abs/2502.07272.

Xu, H., Sharaf, A., Chen, Y., Tan, W., Shen, L., Van Durme, B., Murray, K., and Kim, Y. J. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *International Conference on Machine Learning*. PmLR, 2024. doi: 10.48550/arXiv.2401.08417. URL https://arxiv.org/abs/2401.08417.

Yu, Q., Zhou, C., Jiang, J., Shi, X., and Li, Y. GS-DTI: A graph-structure-aware framework leveraging large language models for drug–target interaction prediction. *Bioinformatics*, 41(8):btaf445, 08 2025. ISSN 1367-4811. doi: 10.1093/bioinformatics/btaf445. URL https://doi.org/10.1093/bioinformatics/btaf445.

Yüksel, A., Ulusoy, E., Ünlü, A., and Doğan, T. SELFormer: Molecular representation learning via SELFIES language models. *Machine Learning: Science and Technology*, 4(2):025035, 2023. doi: 10.1088/2632-2153/acdb30. URL https://doi.org/10.1088/2632-2153/acdb30.

Zhang, X., Shivashankar, G. V., and Uhler, C. Partially shared multi-modal embedding learns holistic representation of cell state. *Nature Computational Science*, 2026. doi: 10.1038/s43588-025-00948-w. URL https://doi.org/10.1038/s43588-025-00948-w.

Zhang, Y., Jian, X., Xu, L., Zhao, J., Lu, M., Lin, Y., and Xie, L. iTCep: a deep learning framework for identification of T cell epitopes by harnessing fusion features. *Frontiers in Genetics*, 14:1141535, 2023. doi: 10.3389/fgene.2023.1141535. URL https://doi.org/10.3389/fgene.2023.1141535.

Zhang, Y., Wang, Z., Jiang, Y., Littler, D. R., Gerstein, M., Purcell, A. W., Rossjohn, J., Ou, H.-Y., and Song, J. Epitope-anchored contrastive transfer learning for paired CD8+ T cell receptor–antigen recognition. *Nature Machine Intelligence*, 6(11):1344–1358, 2024a. doi: 10.1038/s42256-024-00913-8. URL https://doi.org/10.1038/s42256-024-00913-8.

Zhang, Z., Wayment-Steele, H. K., Brixi, G., Wang, H., Kern, D., and Ovchinnikov, S. Protein language models learn evolutionary statistics of interacting sequence motifs. *Proceedings of the National Academy of Sciences*, 121(45):e2406285121, 2024b. doi: 10.1073/pnas.

2406285121. URL https://doi.org/10.1073/pnas.2406285121.

Zhao, J., Zhang, C., and Luo, Y. Contrastive fitness learning: Reprogramming protein language models for low-n learning of protein fitness landscape. In *International Conference on Research in Computational Molecular Biology (RECOMB)*, pp. 470–474. Springer, 2024. doi: 10.1007/978-1-0716-3989-4_55. URL https://doi.org/10.1007/978-1-0716-3989-4_55.

Zitnik, M., Sosic, R., Maheshwari, S., and Leskovec, J. BioSNAP datasets: Stanford biomedical network dataset collection. https://snap.stanford.edu/biodata/, 2018. URL https://snap.stanford.edu/biodata/.

Zvyagin, M., Brace, A., Hippe, K., Deng, Y., Zhang, B., Orozco Bohorquez, C., Clyde, A., Kale, B., Perez-Rivera, D., Ma, H., Mann, C. M., Irvin, M., Ozgulbas, D. G., Vassilieva, N., Pauloski, J. G., Ward, L., Hayot-Sasson, V., Emani, M., Foreman, S., Xie, Z., Lin, D., Shukla, M., Nie, W., Romero, J., Dallago, C., Vahdat, A., Xiao, C., Gibbs, T., Foster, I., Davis, J. J., Papka, M. E., Brettin, T., Stevens, R., Anandkumar, A., Vishwanath, V., and Ramanathan, A. GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics. *The International Journal of High Performance Computing Applications*, 37(6):683–705, 2023. doi: 10.1177/10943420231201154. URL https://doi.org/10.1177/10943420231201154.

A. Related Work

Contextualized biological sequence models. Biological foundation models increasingly augment sequence representations with external biological context, including cellular or tissue state, genomic neighborhoods, interaction networks, and molecular graphs (Li et al., 2024; Hwang et al., 2024; Avsec et al., 2026; Wiatrak et al., 2025; Wang et al., 2024). Closest to our setting are paired or target-conditioned models for peptide design, TCR–peptide binding, interacting proteins, antibody and TCR chain pairing, and partner-specific sequence modeling (Ullanat et al., 2026; Mizrahi et al., 2023; Meynard-Piganeau et al., 2024; Karthikeyan et al., 2025; Chen et al., 2025; Burbach & Briney, 2024; Liu et al., 2025a; Lupo et al., 2024). Many of these methods condition one biological sequence on another through co-encoding, cross-attention, adapters, or conditional masked-language-modeling objectives. They therefore retain, to varying degrees, the token-level interface needed for residue scoring and generation. However, their matching or interaction objectives are typically not defined directly on contextualized token likelihoods. In contrast, LogiCA uses token log-likelihoods themselves as the contrastive matching scores, preserving compatibility with each pretrained model's native output head.

Fine-tuning and adapter-based conditioning. A common way to adapt pretrained masked language models is to fine-tune the entire model or insert parameter-efficient modules such as low-rank adapters (Hu et al., 2022). Such approaches can be applied to paired biological inputs, including protein–protein interactions, antibody heavy/light chains, TCRα/β chains, and other cross-sequence settings (Lupo et al., 2024; Zhang et al., 2024a; Meynard-Piganeau et al., 2024; Burbach & Briney, 2024; Ullanat et al., 2026; Nagano et al., 2025; Deutschmann et al., 2024). Standard masked language modeling trains a model to reconstruct masked tokens from their surrounding context,

$$\mathcal{L}_{\mathrm{MLM}} = -\frac{1}{|\Omega|}\sum_{i\in\Omega}\log p_\theta(x_i \mid x_{\setminus i}) = -\frac{1}{|\Omega|}\sum_{i\in\Omega}\log\frac{\exp(\ell_{i,x_i})}{\sum_{a\in\mathcal{A}}\exp(\ell_{i,a})}, \tag{7}$$

where Ω is the set of masked positions, $\ell_{i,a}$ is the logit for token a at position i and $\mathcal{A}$ is the model vocabulary. Conditional MLMs extend this objective by providing an additional context y, yielding token probabilities of the form $p_\theta(x_i \mid x_{\setminus i}, y)$. While this preserves the probabilistic token interface, reconstruction alone is often a weak signal for biological compatibility: it encourages recovery of observed tokens, but does not directly separate matched from mismatched contexts. LogiCA instead uses paired supervision contrastively in logit space, so that the contextualized token likelihoods are optimized as compatibility scores.

Latent-space contrastive matching. A second line of work learns compatibility in a shared latent space using CLIP-style contrastive supervision (Radford et al., 2021). This template has been applied to drug–target interaction and virtual screening (Jia et al., 2026; Singh et al., 2023), protein retrieval (Su et al., 2025), TCR–antigen recognition (Zhang et al., 2024a; Nagano et al., 2025), and multimodal biological integration (Gayoso et al., 2021; Ashuach et al., 2023; Zhang et al., 2026). The standard InfoNCE or NT-Xent objective scores a sequence–context pair using a pooled-latent similarity, for example

$$s_z(x, y) = \langle f_\theta(x),\, g_\phi(y)\rangle, \tag{8}$$

where f_θ and g_ϕ map the sequence and context into a shared latent space. These scores are effective for retrieval and binary matching, but they no longer correspond to token likelihoods. Consequently, the per-position distribution $p_\theta(x_i \mid x_{\setminus i}, y)$ is not recoverable from $s_z(x, y)$, making it difficult to localize scores to mutated residues, perform likelihood-based generation, or analyze position-wise probabilistic signals. LogiCA keeps the contrastive template but changes the scored object: rather than contrasting pooled latents, it contrasts context-conditioned token log-likelihoods.

Likelihood-based variant ranking. Variant-effect prediction is naturally comparative: experiments often ask which variant is more functional, resistant, or compatible under a fixed biological condition. Protein language models commonly score mutations using wild-type-normalized log-likelihood ratios at the mutated positions $\mathcal{M}$,

$$s^{\circ}_{\mathcal{M}}(x;\, x^{\mathrm{wt}}) = \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}}\log\frac{p_\theta(x_i \mid x_{\setminus i})}{p_\theta(x_i^{\mathrm{wt}} \mid x_{\setminus i}^{\mathrm{wt}})}, \tag{9}$$

and such scores are widely used in protein variant-effect benchmarks (Meier et al., 2021; Notin et al., 2023). Related likelihood-based approaches incorporate evolutionary context, retrieval-augmented protein families, or pairwise ranking supervision (Gong et al., 2024; Notin et al., 2022a; Pugh et al., 2025; Lee et al., 2023; Zhao et al., 2024). These methods

motivate the use of token likelihoods for ranking, but the conditioning signal is usually the sequence itself, an evolutionary family, or preference supervision over variants. LOGICA instead makes the conditioning variable an explicit external biological context, such as a drug, epitope, ligand, or binding partner, yielding scores based on $p_\theta(x_i \mid x_{\setminus i}, y)$.

Preference and logit-space contrastive objectives. Preference-based fine-tuning provides another route for adapting pretrained language models to task-specific comparisons. Direct Preference Optimization (DPO) (Rafailov et al., 2023) and related methods optimize models so that preferred outputs receive higher likelihood than dispreferred ones. Extensions to masked language models and biological sequence models often use pseudo-log-likelihood or average token likelihood as a sequence-level reward (Lee et al., 2023; Zhao et al., 2024; Hawkins-Hooker et al., 2024), for example

$$r_\theta(x) = \frac{1}{L}\sum_{i=1}^{L} \log p_\theta(x_i \mid x_{\setminus i}), \tag{10}$$

and optimize pairwise preferences with a soft-margin or Bradley–Terry-style loss. Recent work has also explored contrastive supervision in log-likelihood space for autoregressive models, including contrastive preference learning and contrastive preference optimization (Hejna et al.; Xu et al., 2024). These methods compare outputs conditioned on a single input, typically for generation or translation. LOGICA differs in both setting and scoring: it performs contrastive learning over structured biological pairs and uses contextualized token likelihoods from masked biological language models as bidirectional compatibility scores.

How LOGICA differs. LOGICA sits at the intersection of contextual biological modeling, contrastive alignment, and likelihood-based variant ranking. Unlike latent-space contrastive methods, it does not replace the pretrained token head with a pooled similarity score. Unlike standard conditional MLM fine-tuning, it directly contrasts matched and mismatched biological contexts. Unlike existing likelihood-based ranking methods, it conditions token probabilities on explicit external partners. This makes LOGICA particularly suited to mutation-local variant ranking, where variants share a wild-type reference and mutated sites, so shared sequence terms cancel and comparisons reduce to context-conditioned mutant-token likelihoods at the perturbed positions.

B. Emerging capabilities of the trained token logits in LOGICA

A key consequence of logit-space alignment is that the fine-tuned model remains a conditional token model. The same probabilities used for contrastive or preference training can therefore be reused for ranking, interpretation, and generation without introducing new task-specific heads.

B.1. Direct likelihood-based ranking

For a fixed context y, reference sequence x^{wt}, and variant x, LOGICA ranks candidates by the mutation-local score introduced in Eq. 3:

$$r(x; y, x^{\mathrm{wt}}) = s_{\mathcal{M}}(x, y; x^{\mathrm{wt}}), \qquad \mathcal{M} = \{i : x_i \neq x_i^{\mathrm{wt}}\}. \tag{11}$$

Given a candidate set $\mathcal{X} = \{x^{(1)}, \ldots, x^{(K)}\}$, variants are sorted in decreasing order of $r(x^{(k)}; y, x^{\mathrm{wt}})$. Equivalently, the scores define a soft ranking distribution

$$p(x^{(k)} \mid y, x^{\mathrm{wt}}, \mathcal{X}) = \frac{\exp\big(s_{\mathcal{M}_k}(x^{(k)}, y; x^{\mathrm{wt}})/\tau\big)}{\sum_{\ell=1}^{K} \exp\big(s_{\mathcal{M}_\ell}(x^{(\ell)}, y; x^{\mathrm{wt}})/\tau\big)}, \tag{12}$$

where $\mathcal{M}_k = \{i : x_i^{(k)} \neq x_i^{\mathrm{wt}}\}$. Thus inference uses exactly the same likelihood-ratio quantity optimized during preference training.

B.2. Cross-modality dependency maps

Because LOGICA preserves normalized token distributions, it can be probed by perturbing one token and measuring the induced change in another token's predicted distribution. This gives a dependency map over positions. Prior token-probability perturbation analyses have shown that such sensitivities can reveal structural contacts, interacting motifs, and evolutionary constraints within a single sequence (Tomaz da Silva et al., 2025; Zhang et al., 2024b; Cornman et al., 2024). LOGICA extends this idea across the conditioning interface.

For within-sequence dependencies, let $x^{(j\to a)}$ denote the sequence obtained by replacing token x_j with a, and let $\mathcal{A}_j^x$ be the set of allowed substitutions at position j. We define

$$D_{ij}^{x\to x}(y) = \frac{1}{|\mathcal{A}_j^x|} \sum_{a\in\mathcal{A}_j^x} \left\| \pi_{\theta,i}(\cdot \mid x_{\setminus i}^{(j\to a)}, y) - \pi_{\theta,i}(\cdot \mid x_{\setminus i}, y) \right\|_2 . \tag{13}$$

Large $D_{ij}^{x\to x}(y)$ indicates that perturbing position j in the scored sequence changes the predicted distribution at position i, under the fixed context y.

For cross-modality dependencies, we instead perturb the context. Let $y^{(j\to a)}$ be the context obtained by substituting token y_j with a, and let $\mathcal{A}_j^y$ be the allowed substitution set for that context position. We define

$$D_{ij}^{y\to x} = \frac{1}{|\mathcal{A}_j^y|} \sum_{a\in\mathcal{A}_j^y} \left\| \pi_{\theta,i}(\cdot \mid x_{\setminus i}, y^{(j\to a)}) - \pi_{\theta,i}(\cdot \mid x_{\setminus i}, y) \right\|_2 . \tag{14}$$

This quantity measures how strongly token j in the context affects the model's predicted distribution at token i in the scored sequence. When both directions are available, the reverse map $D_{ji}^{x\to y}$ is computed analogously by perturbing x and measuring changes in the predicted token distribution over y. Together, these maps provide a token-level view of the intermodal dependencies learned by the model.

B.3. Context-conditioned generation by Gibbs sampling

Since LOGICA remains a conditional language model, it can also be used as a generative model under a fixed context. Let $A \subseteq [L]$ be a set of designable positions and let $x_{\setminus A}$ denote the fixed sequence background. The model defines the pseudo-likelihood

$$p_\theta(x_A \mid x_{\setminus A}, y) \propto \prod_{i\in A} \pi_\theta(x_i \mid x_{\setminus i}, y). \tag{15}$$

We sample from this distribution with Gibbs updates. Starting from an initial sequence $x^{(0)}$, each step selects a design position $i \in A$ and resamples

$$x_i^{(t+1)} \sim \pi_\theta\Big(\cdot \mid x_{\setminus i}^{(t)}, y\Big), \qquad x_j^{(t+1)} = x_j^{(t)} \text{ for } j \neq i. \tag{16}$$

The sampler can be constrained to valid biological tokens, fixed motif positions, interface residues, or a mutation budget around a reference sequence. When a reference x^{wt} is available, proposals can also be ranked or filtered by the change in the LOGICA score,

$$\Delta s = s_{\mathcal{M}}(x^{\mathrm{new}}, y; x^{\mathrm{wt}}) - s_{\mathcal{M}}(x^{\mathrm{old}}, y; x^{\mathrm{wt}}). \tag{17}$$

Thus the same logits used for ranking can be used to propose and refine context-compatible sequences.

C. Theoretical Analysis of Mutation-Local Variant Scoring

This appendix formalizes why the mutation-local scoring objective used by LOGICA provides localized supervision for variant ranking. We begin by showing that, when two variants share the same wild-type reference and the same set of mutated positions, the wild-type anchoring term in Eq. 3 cancels exactly in pairwise score differences. Consequently, the mutation-local ranking score reduces exactly to the context-conditioned likelihoods of the competing mutant tokens at the perturbed sites. This reduction is specific to the mutation-local objective and does not generally hold for ranking objectives based on pooled-latent similarities. We next show that the associated pairwise preference loss induces direct score-level derivatives on the mutated-site likelihood terms, ensuring that optimization targets the positions that distinguish the variants. Finally, we study multi-site comparisons under a correlated sub-Gaussian noise model that captures non-deterministic scoring at mutated positions. The deterministic cancellation result remains valid for any fixed realization of the scores, while the probabilistic analysis shows that averaging across mutated sites can improve ranking reliability by reducing the probability that noise reverses the correct ordering.

C.1. Exact Cancellation for Matched Mutation Sets

We first consider two variants of the same wild-type sequence x^{wt} under context y. Let x^A and x^B perturb the same nonempty set of positions relative to the wild type:

$$\mathcal{M} = \mathcal{M}(x^A, x^{\mathrm{wt}}) = \mathcal{M}(x^B, x^{\mathrm{wt}}), \qquad m = |\mathcal{M}| \geq 1.$$

The variants must agree on which positions are mutated, but the substituted tokens at those positions may differ. For $V \in \{A, B, \mathrm{wt}\}$ and $i \in [L]$, we define

$$\ell_i^V := \log \pi_\theta(x_i^V \mid x_{\setminus i}^V, y).$$

The per-site likelihood advantage of x^A over x^B is

$$d_i := \ell_i^A - \ell_i^B,$$

and the averaged score gap over the shared mutation set is

$$\Delta := \frac{1}{m} \sum_{i \in \mathcal{M}} d_i.$$

When a candidate is identical to the wild type, we use the convention $s_\varnothing(x^{\mathrm{wt}}, y; x^{\mathrm{wt}}) = 0$, corresponding to zero log-likelihood change relative to the reference. The results below pertain to the nonempty case $m \geq 1$.

Proposition C.1 (Mutation-local reduction)**.** *For two variants x^A and x^B with the same wild-type reference and the same nonempty mutation set $\mathcal{M}$,*

$$s_{\mathcal{M}}(x^A, y; x^{\mathrm{wt}}) - s_{\mathcal{M}}(x^B, y; x^{\mathrm{wt}}) = \Delta.$$

Consequently, for the two-candidate ranking problem $\mathcal{C} = \{x^A, x^B\}$, Eq. 1 gives

$$\Pr(x^A \succ x^B \mid y, \mathcal{C}) = \sigma(\Delta/\tau).$$

Thus, any additive term shared by all candidates with the same context y and mutation set $\mathcal{M}$ cancels from the pairwise ranking, including the wild-type anchor term in Eq. 3.

Proof. By Eq. 3,

$$\begin{aligned}
&s_{\mathcal{M}}(x^A, y; x^{\mathrm{wt}}) - s_{\mathcal{M}}(x^B, y; x^{\mathrm{wt}}) \\
&= \left[\frac{1}{m} \sum_{i \in \mathcal{M}} \log \frac{\pi_\theta(x_i^A \mid x_{\setminus i}^A, y)}{\pi_\theta(x_i^{\mathrm{wt}} \mid x_{\setminus i}^{\mathrm{wt}}, y)}\right] - \left[\frac{1}{m} \sum_{i \in \mathcal{M}} \log \frac{\pi_\theta(x_i^B \mid x_{\setminus i}^B, y)}{\pi_\theta(x_i^{\mathrm{wt}} \mid x_{\setminus i}^{\mathrm{wt}}, y)}\right] \\
&= \frac{1}{m} \sum_{i \in \mathcal{M}} \Big[\log \pi_\theta(x_i^A \mid x_{\setminus i}^A, y) - \log \pi_\theta(x_i^B \mid x_{\setminus i}^B, y)\Big] \\
&= \frac{1}{m} \sum_{i \in \mathcal{M}} (\ell_i^A - \ell_i^B) = \Delta.
\end{aligned}$$

The wild-type anchor term cancels because both candidates are evaluated against the same reference sequence over the same mutation set. Substituting this score difference into the two-candidate form of Eq. 1 gives the stated Bradley–Terry probability. □

The cancellation above is exact for matched mutation sets. If two variants perturb different positions, the wild-type anchor terms are evaluated over different sets and need not cancel. For matched mutation sets, however, the pairwise objective reduces exactly to a comparison of the contextualized log-likelihoods of the competing mutant tokens at the perturbed sites.

Why latent scores do not admit the same reduction. This reduction is specific to the mutation-local likelihood score. Pooled latent similarities used in CLIP-style contrastive learning do not generally have the same additive structure or shared wild-type anchor. For example, if

$$s_z(x, y) = \langle f_\theta(x),\, g_\phi(y)\rangle,$$

then

$$s_z(x^A, y) - s_z(x^B, y) = \langle f_\theta(x^A),\, g_\phi(y)\rangle - \langle f_\theta(x^B),\, g_\phi(y)\rangle.$$

This score difference is generally a nonlinear function of the full-sequence embeddings $f_\theta(x^A)$ and $f_\theta(x^B)$. Even when the two variants differ only on $\mathcal{M}$, their pooled representations may change globally, and there is no shared wild-type anchor term to remove algebraically. Pooled contrastive objectives can therefore learn useful pair-level compatibility scores, but they do not provide the same exact reduction from pairwise ranking to mutant-token likelihoods.

Takeaway. For matched variant comparisons, LOGICA compares candidates through the native token probabilities of the language-model head. In contrast, pooled latent methods compare global sequence representations, so their pairwise score differences do not identify an exact algebraic path back to the specific mutant-token likelihoods.

C.2. Score-Level Locality of Preference Gradients

At the level of score variables, the exact cancellation above implies that the pairwise preference loss is directly supported on the mutant-token likelihoods at the perturbed sites. This is formalized by the following score-level derivative calculation.

Corollary C.2 (Mutation-local gradients)**.** *Under the pairwise preference loss*

$$\mathcal{L}_{\mathrm{BT}} = -\log \sigma(\Delta/\tau), \qquad \Delta = \frac{1}{m}\sum_{i\in\mathcal{M}} (\ell_i^A - \ell_i^B),$$

the direct partial derivatives with respect to the site log-likelihood terms are

$$\frac{\partial \mathcal{L}_{\mathrm{BT}}}{\partial \ell_i^A} = -\frac{1}{m\tau}\sigma(-\Delta/\tau), \qquad \frac{\partial \mathcal{L}_{\mathrm{BT}}}{\partial \ell_i^B} = \frac{1}{m\tau}\sigma(-\Delta/\tau), \qquad i \in \mathcal{M},$$

and these direct partial derivatives are zero for $i \notin \mathcal{M}$.

Proof. Since

$$\frac{d}{dt}\log\sigma(t) = \sigma(-t),$$

we have

$$\frac{\partial \mathcal{L}_{\mathrm{BT}}}{\partial \Delta} = -\frac{1}{\tau}\sigma(-\Delta/\tau).$$

For $i \in \mathcal{M}$,

$$\frac{\partial \Delta}{\partial \ell_i^A} = \frac{1}{m}, \qquad \frac{\partial \Delta}{\partial \ell_i^B} = -\frac{1}{m}.$$

The stated derivatives follow by the chain rule. If $i \notin \mathcal{M}$, then ℓ_i^A and ℓ_i^B do not appear as direct terms in Δ, so the corresponding direct partial derivatives are zero. □

Remark. Corollary C.2 describes score-level partial derivatives, not full parameter gradients. Because neural network parameters are shared across positions, and because masked-token likelihoods at mutant sites condition on the surrounding sequence, parameter gradients can still depend on unmutated residues through the model architecture. The key point is that the preference loss is directly supported on the mutated-site likelihood terms.

Takeaway. The loss gives direct positive pressure to increase the likelihood of the preferred mutant tokens and direct negative pressure to decrease the likelihood of the dispreferred mutant tokens at the same perturbed positions. Unchanged positions still shape the conditional distribution through the sequence context, but they are not explicit score terms in the mutation-local objective.

C.3. Dependence-Aware Concentration of Noisy Site-Level Evidence

The preceding results are algebraic and hold for any fixed model scores. We now ask how the averaged gap Δ behaves when the per-site likelihood advantages are viewed as noisy measurements of an underlying preference signal. This captures a statistical setting in which different mutated sites provide imperfect but positively biased evidence for the same preferred variant.

Unlike the deterministic cancellation argument, this concentration result requires a probabilistic model for the site-level score gaps. Because masked language models condition each masked-token prediction on the surrounding sequence and share parameters across positions, the site-level advantage terms for multi-site variants need not be statistically independent. We therefore use a dependence-aware sub-Gaussian model: the noise vector may have cross-site dependence, summarized by a positive semidefinite dependence proxy.

Assumption C.3 (Sub-Gaussian advantages)**.** View $\{d_i\}_{i\in\mathcal{M}}$ as random site-level advantages induced by a population of mutation comparisons under a fixed context. Write

$$d_i = \mu_i + \epsilon_i,$$

where μ_i is the mean advantage at site i. Let

$$\epsilon_{\mathcal{M}} = (\epsilon_i)_{i\in\mathcal{M}} \in \mathbb{R}^m$$

denote the vector of centered site-level noise terms. We assume that $\epsilon_{\mathcal{M}}$ is jointly sub-Gaussian in the sense that, for every $a \in \mathbb{R}^m$,

$$\mathbb{E}\big[\exp(a^\top \epsilon_{\mathcal{M}})\big] \le \exp\left(\frac{1}{2} a^\top \Sigma_{\mathcal{M}} a\right),$$

for some positive semidefinite matrix $\Sigma_{\mathcal{M}} \in \mathbb{R}^{m\times m}$.
Let

$$\bar{\mu}_{\mathcal{M}} = \frac{1}{m}\sum_{i\in\mathcal{M}} \mu_i, \qquad \nu^2_{\mathcal{M}} = \frac{1}{m^2}\mathbf{1}^\top \Sigma_{\mathcal{M}} \mathbf{1},$$

where $\mathbf{1} \in \mathbb{R}^m$ is the all-ones vector.

Remark on the modeling assumption. Assumption C.3 is intended as a simple concentration model for the averaged site-level score gap, not as a complete generative model of masked language model predictions. The joint sub-Gaussian condition allows the site-level advantage terms to be statistically dependent. This is important for multi-site variants, where masked language model predictions are coupled through the shared sequence context and shared model parameters. The dependence is summarized through $\mathbf{1}^\top \Sigma_{\mathcal{M}} \mathbf{1}$, the sub-Gaussian proxy for the averaged noise direction. The deterministic cancellation result in Theorem C.1 and the score-level gradient statement in Corollary C.2 do not rely on this probabilistic assumption.

Corollary C.4 (Misranking bound)**.** *Under Assumption C.3,* $\mathbb{E}[\Delta] = \bar{\mu}_{\mathcal{M}}$ *and* $\Delta - \bar{\mu}_{\mathcal{M}}$ *is* $\nu^2_{\mathcal{M}}$*-sub-Gaussian. If* $\bar{\mu}_{\mathcal{M}} > 0$*, then*

$$\Pr\big[s_{\mathcal{M}}(x^A, y; x^{\mathrm{wt}}) \le s_{\mathcal{M}}(x^B, y; x^{\mathrm{wt}})\big] \le \exp\left(-\frac{\bar{\mu}^2_{\mathcal{M}}}{2\nu^2_{\mathcal{M}}}\right).$$

Proof. By definition,

$$\Delta = \frac{1}{m}\sum_{i\in\mathcal{M}} d_i = \frac{1}{m}\sum_{i\in\mathcal{M}} \mu_i + \frac{1}{m}\sum_{i\in\mathcal{M}} \epsilon_i = \bar{\mu}_{\mathcal{M}} + \frac{1}{m}\mathbf{1}^\top \epsilon_{\mathcal{M}}.$$

Thus $\mathbb{E}[\Delta] = \bar{\mu}_{\mathcal{M}}$. Moreover, for any $\lambda \in \mathbb{R}$, applying Assumption C.3 with $a = (\lambda/m)\mathbf{1}$ gives

$$\begin{aligned}\mathbb{E}[\exp(\lambda(\Delta - \bar{\mu}_{\mathcal{M}}))] &= \mathbb{E}\left[\exp\left(\frac{\lambda}{m}\mathbf{1}^\top \epsilon_{\mathcal{M}}\right)\right] \\ &\leq \exp\left(\frac{1}{2}\frac{\lambda^2}{m^2}\mathbf{1}^\top \Sigma_{\mathcal{M}}\mathbf{1}\right) \\ &= \exp\left(\frac{\lambda^2 \nu_{\mathcal{M}}^2}{2}\right).\end{aligned}$$

Therefore $\Delta - \bar{\mu}_{\mathcal{M}}$ is $\nu_{\mathcal{M}}^2$-sub-Gaussian.

The misranking event is

$$\{\Delta \leq 0\} = \{\Delta - \bar{\mu}_{\mathcal{M}} \leq -\bar{\mu}_{\mathcal{M}}\}.$$

Applying the standard one-sided sub-Gaussian tail bound yields

$$\Pr[\Delta \leq 0] \leq \exp\left(-\frac{\bar{\mu}_{\mathcal{M}}^2}{2\nu_{\mathcal{M}}^2}\right).$$

Finally, Theorem C.1 identifies Δ with the difference between the two mutation-local scores, so this is the stated misranking bound. □

Independent-site special case. If the site-level noise terms are independent and each ϵ_i is σ_i^2-sub-Gaussian, then $\Sigma_{\mathcal{M}}$ may be taken to be diagonal with entries σ_i^2. In that case,

$$\nu_{\mathcal{M}}^2 = \frac{1}{m^2}\sum_{i \in \mathcal{M}} \sigma_i^2,$$

which recovers the independent-site bound as a special case. If additionally $\sigma_i^2 \leq \sigma^2$ for all $i \in \mathcal{M}$, then

$$\nu_{\mathcal{M}}^2 \leq \frac{\sigma^2}{m},$$

and hence

$$\Pr\left[s_{\mathcal{M}}(x^A, y; x^{\mathrm{wt}}) \leq s_{\mathcal{M}}(x^B, y; x^{\mathrm{wt}})\right] \leq \exp\left(-\frac{m\bar{\mu}_{\mathcal{M}}^2}{2\sigma^2}\right).$$

Bounded-correlation interpretation. The dependence-aware bound also clarifies how cross-site coupling changes the benefit of averaging. Suppose, for example, that $\Sigma_{ii} \leq \sigma^2$ and $\Sigma_{ij} \leq \rho\sigma^2$ for $i \neq j$. Then

$$\nu_{\mathcal{M}}^2 = \frac{1}{m^2}\mathbf{1}^\top \Sigma_{\mathcal{M}}\mathbf{1} \leq \frac{\sigma^2}{m}\{1 + (m-1)\rho\}.$$

Consequently,

$$\Pr[\Delta \leq 0] \leq \exp\left(-\frac{m\bar{\mu}_{\mathcal{M}}^2}{2\sigma^2\{1 + (m-1)\rho\}}\right).$$

When $\rho = 0$, this reduces to the independent-site rate. Larger positive cross-site dependence weakens the concentration benefit of averaging, whereas small cross-site dependence preserves much of the stabilizing effect of multi-site evidence.

Takeaway. The concentration result does not change the deterministic cancellation argument. It says that if each mutated site provides a noisy estimate of a positive preference signal, then averaging the mutation-local likelihood advantages reduces the chance that noise reverses the pairwise ranking. The general bound allows these site-level noise terms to be statistically dependent, which is more appropriate for masked language models whose predictions are coupled across positions through sequence context and shared parameters.

D. Datasets and Preprocessing

D.1. Protein–ligand binding

BindingDB (Liu et al., 2025c) provides both the protein–ligand pretraining corpus and one of the held-out DTI fine-tuning split. We describe them together to make explicit how the large-scale pretraining data relate to the downstream BindingDB benchmark.

Pretraining corpus. We convert ligand SMILES strings to SELFIES (Yüksel et al., 2023), remove entries with missing protein sequences or invalid ligand conversions, and truncate inputs to 512 protein tokens and 128 ligand tokens. We retain four measurement types (K_d, K_i, IC_{50}, and EC_{50}) convert all affinity values to nanomolar units, and define positives using the first-quartile threshold within each measurement type. To prevent leakage, we remove from the pretraining corpus any protein–ligand pair that appears in the downstream validation or test splits. The resulting processed table contains 21,461,880 protein–ligand rows spanning 1,203,672 ligands and 8914 proteins. Thresholding yields 7,798,830 positives and 13,663,050 negatives. The measurement-type composition is 66.73% IC_{50}, 20.44% K_i, 9.32% EC_{50}, and 3.51% K_d.

Single-residue mutant sites in the pretraining corpus. The mutation-local score $s_{\mathcal{M}}$ in Eq. 3 relies on context-conditioned changes in the per-residue distribution at mutated sites. To check whether this regime is represented in pretraining, we searched for near-neighbor protein pairs among the 8914 unique pretraining proteins using a length-bucketed, pigeonhole-hashed Hamming search over same-length sequences. This procedure excludes indels and identifies naturally occurring substitution pairs up to Hamming distance 20. The resulting graph contains 4942 mutational edges over 1214 proteins, including 713 single-substitution pairs. These distance-1 pairs directly match the single-residue setting targeted by $s_{\mathcal{M}}$. After joining them with BindingDB ligand annotations, they yield 50,274 ligand-anchored mutation rows during pretraining, providing naturally observed single-amino-acid contrasts under multiple drug contexts. The full distance distribution is shown in Figure S1.

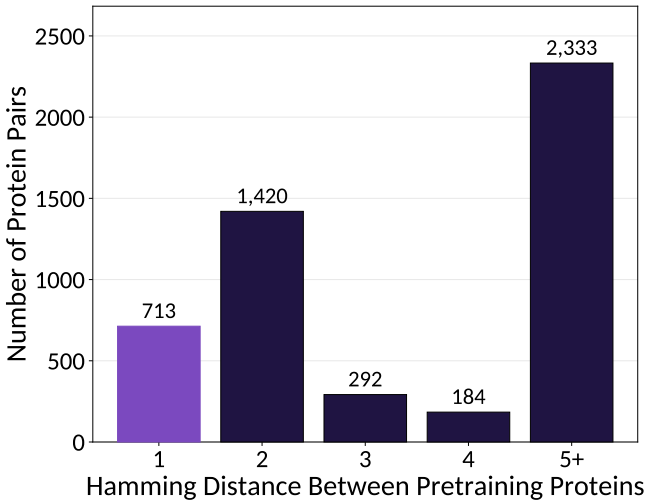


Figure S1. Mutation-level structure in the BindingDB pretraining corpus. Near-neighbor protein pairs are enriched for low Hamming distances, including a visible set of single-residue substitutions. These distance-1 pairs define the mutation-local regime targeted by $s_{\mathcal{M}}$ and yield ligand-anchored mutation contrasts after joining to BindingDB annotations.

Protein–ligand benchmarks: BindingDB(Test), DAVIS, and BioSNAP. For downstream protein–ligand evaluation, we follow the fixed train/validation/test splits introduced by MolTrans for DAVIS (Davis et al., 2011), BindingDB (Liu et al., 2025c), and BioSNAP (Zitnik et al., 2018). DAVIS provides a kinase–inhibitor benchmark, with pairs labeled positive when the reported K_d is below 30 nM. BindingDB is used as the larger held-out transfer benchmark. After filtering for complete protein–ligand pairs, valid SMILES strings, and available pockets, the reproducible BindingDB split contains 12,662 training pairs, 6637 validation pairs, and 13,279 test pairs. Following the standard MolTrans/ConPLex protocol, the training set is balanced, while validation and test retain the natural positive rate of approximately 14%.

D.2. Oncogene–drug deep mutational scanning

The variant-ranking benchmark combines two experimental resources for measuring how oncogene mutations alter drug response: a broad multi-oncogene screen (Coelho et al., 2024) and an EGFR-focused prime-editing panel (Kim et al., 2025). These screens assay single-amino-acid substitutions in cancer-associated genes and measure their effects under targeted therapies, producing variant–drug response scores that can be used to evaluate whether a model ranks resistance-associated mutations above sensitive or neutral mutations. We rank variants using the contextualized variant score $s_{\mathcal{M}}(x, y; x^{\mathrm{wt}})$ from Eq. 3, where y is the SELFIES-encoded drug.

The benchmark covers 11 oncogenes (*AKT1, BCL2, BRAF, EGFR, KRAS, MAP2K1, MAP2K2, MYC, PARP1, PARP2, PIK3CA*) and 10 therapies (Dabrafenib, Trametinib, Pictilisib, Adagrasib, Sotorasib, Osimertinib, Gefitinib, Olaparib, Niraparib, and Afatinib). These gene–drug combinations span several clinically relevant targeted-therapy settings, including MAPK-pathway inhibition, EGFR inhibition, PI3K/AKT-pathway inhibition, KRAS inhibition, and PARP inhibition.

Table S1. Oncogene–drug benchmark composition by gene. *Variants* counts unique single-amino-acid substitutions for each gene; *drugs with data* counts therapies, out of 10 total, with at least one scored variant for that gene; and *measurements* sums the non-missing variant–drug scores across those therapies. EGFR has the largest number of measurements because it is the only gene covered by the EGFR-focused panel.

Gene	Variants	Drugs with data	Measurements
AKT1	237	9	2,013
BCL2	136	9	1,152
BRAF	250	9	1,980
EGFR	2,387	10	7,874
KRAS	57	9	447
MAP2K1	152	9	1,264
MAP2K2	203	9	1,733
MYC	164	9	1,378
PARP1	410	9	3,380
PARP2	169	9	1,341
PIK3CA	280	9	2,208
Total	4,445	—	24,770

After removing synonymous changes and variants beyond position 1023, which lies outside the variant-fine-tune protein context window of 1024 tokens (raised from the 512 used during pretraining and DTI fine-tuning to cover the longer oncogenes; see Appendix E.1), the benchmark contains 4,445 distinct single-amino-acid substitutions across the 11 genes. Per-gene variant counts range from 57 to 2,387, with a median of 203 and a mean of 404 (Table S1). Drug coverage is not uniform across genes: nine therapies from the broad multi-oncogene panel are screened against every gene, whereas Afatinib is restricted to EGFR. This produces 100 gene–drug screening pairs in total and 24,770 scored variant–drug entries overall. Each gene–drug pair contains an average of 247.7 measured variant–drug scores (median 169, minimum 24, maximum 2,387).

For each gene–drug pair, evaluation uses all measured non-synonymous single-amino-acid substitutions within the protein context window. We compute Spearman ρ and binary resistance AUC using variants with non-missing experimental scores, and Table 2 reports gene-wise means across drugs.

D.3. TCR–peptide pretraining, fine-tuning, and evaluation data

Pretraining corpus. We construct the TCR–peptide pretraining corpus by combining CDR–peptide binders from IEDB (Vita et al., 2019) with paired CDR3α–CDR3β–peptide annotations curated in prior studies (Zhang et al., 2024a; Kwee et al., 2023). Each example is standardized as $(\mathbf{t}, \mathbf{p})$, where $\mathbf{t} = \texttt{CDR3}\beta \mid \texttt{CDR3}\alpha$ denotes the paired TCR sequence when both chains are available, and $\mathbf{p}$ denotes the target peptide sequence. For entries where CDR3α is unavailable, we use the CDR3β sequence alone.

After concatenation, de-duplication, and removal of 14 TCR–peptide pairs that appear in downstream evaluation test sets, the final pretraining corpus contains 260,163 experimentally supported TCR–peptide pairs. Of these, 118,062 pairs include paired CDR3α/CDR3β annotations, while 142,101 contain CDR3β only. Overall, the corpus covers 166,179 unique TCRs and 1,593 unique peptides (Figure S2). We split the pretraining corpus 90%/10% into training and validation sets using

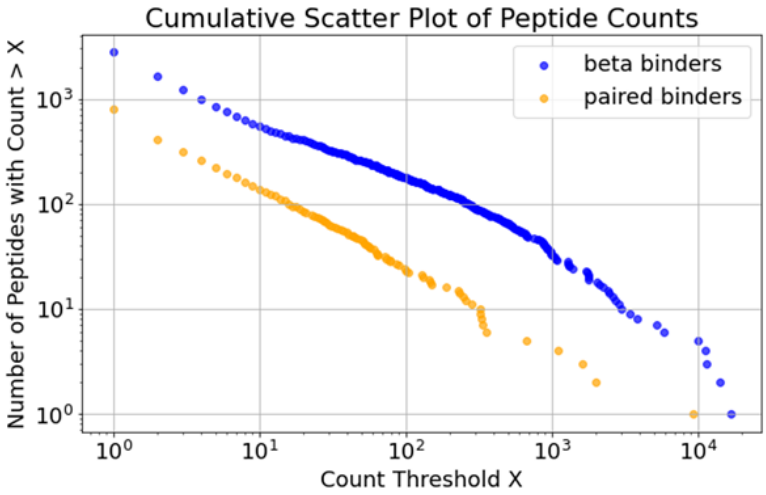


Figure S2. Cumulative number of TCR–peptide pairs in the LOGICA pretraining corpus with at least a given number of binding assays in the curated dataset. Pairs with both CDR3 chains available are substantially scarcer than pairs with only CDR3β annotated.

source-stratified sampling, and use the validation set to select the best checkpoint.

Pretraining negatives. Negative examples are generated online during pretraining using mutation-based corruption. For each positive pair $(\mathbf{t}, \mathbf{p})$, we hold one sequence fixed as the anchor and generate K negatives by randomly mutating one amino-acid position in the non-anchor sequence. This provides local contrastive supervision: the model is trained to assign higher contextualized token likelihoods to the experimentally observed partner than to nearby mutation-based alternatives under the same anchor. Although a random mutation is not guaranteed to abolish binding, most local substitutions reduce compatibility (92%, see Fig. S3), effectively forcing the model to learn locally sensitive hidden states.

We train three TCR–peptide LOGICA variants that differ in anchoring direction and number of negatives:

1. **LOGICA-TCR** fixes the TCR $\mathbf{t}$ and generates $K = 1$ mutated peptide negative.
2. **LOGICA-Pep** fixes the peptide $\mathbf{p}$ and generates $K = 5$ mutated TCR negatives.
3. **LOGICA-Dual** alternates between peptide-anchored batches with $K = 5$ and TCR-anchored batches with $K = 1$, using a 5:1 ratio.

This objective uses the same token log-likelihood primitive as conditional MLM, but places it inside a contrastive ranking loss over local alternatives. Thus, pretraining encourages the native token heads to encode partner-specific compatibility rather than only reconstructing observed tokens in isolation.

Supervised fine-tuning. After mutation-based pretraining, we further fine-tune on paired CDR3α–CDR3β TCR–peptide examples only. For this supervised stage, positive binding pairs are drawn from the paired-TCR binding annotations, and negatives are generated by random peptide shuffling rather than local mutation (Gao et al., 2023; Dens et al., 2023). Specifically, for each positive pair $(\mathbf{t}, \mathbf{p})$, we sample one negative peptide uniformly from the pool of unique peptides not observed to bind the corresponding TCR, producing a balanced 1:1 positive-to-negative dataset. Because the negative peptide is sampled from unrelated observed peptides, these labels are not ambiguous like the single-mutation negatives used during pretraining. The fine-tuning corpus is split 90%/10% into training and validation sets using source-stratified sampling, and the validation set is used to select the best fine-tuned checkpoint.

TCR–peptide variant-ranking benchmarks. We evaluate TCR–peptide zero-shot variant ranking on ePytope (Drost et al., 2025), BATCAVE (Banerjee et al., 2025), and ATLAS (Borrman et al., 2017) datasets. These benchmarks test whether LOGICA can rank peptide or TCR variants under a fixed binding context using mutation-local token likelihoods.

The ePytope benchmark (Drost et al., 2025) consists of deep mutational scans for two human 9-mer peptides. The neopeptide VPSVWRSSL contains 804 TCR–peptide measurements across 6 TCRs and 134 peptide variants plus the wild type. The CMV peptide NLVPMVATV contains 3,440 measurements across 20 TCRs and 172 peptide variants plus the wild type. Each TCR–variant pair was measured by NFAT reporter expression using flow cytometry, which we use for correlation-based variant ranking. The original benchmark also provides binary binding labels derived from peptide-specific NFAT thresholds of 66.09% for VPSVWRSSL and 40.0% for NLVPMVATV; we report binary classification results using these labels in Table S15.

For BATCAVE (Banerjee et al., 2025), we restrict evaluation to TCR–peptide pairs measured by NFAT luminescence. BATCAVE contains multiple assay types, including TScan-II, CD137 expression, ELISA, ELISpot, TCR-MAP, TNF secretion, and multimer depletion. We focus on NFAT luminescence because it is a direct functional T-cell activation readout and is closest to the reporter signal used in ePytope. The resulting BATCAVE benchmark contains 5,754 TCR–peptide measurements across 35 unique TCRs and 478 variant peptides, spanning three index peptides: NLVPMVATV, TPQDLNTML, and VPSVWRSSL. Across BATCAVE studies, only a small fraction of peptide mutations improve activity over wild type in most complexes (Figure S3).

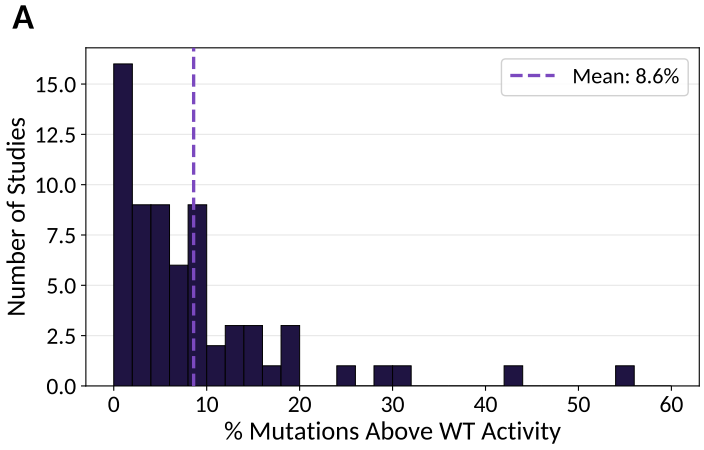


Figure S3. TCR–peptide mutation tolerance in the BATCAVE deep-mutational-scanning corpus (Banerjee et al., 2025). Across studies, only a small fraction of peptide mutations improve activity over wild type in most complexes, consistent with strong sequence specificity and a smaller tail of more permissive recognition settings. The dashed line marks the across-study mean.

The ATLAS benchmark (Borrman et al., 2017) contains both peptide-mutation and TCR-mutation measurements. We derive two evaluation tables. In the peptide-mutation table, paired TCRs are held fixed and peptides are varied. In the TCR-mutation table, peptides are held fixed and CDR3 sequences are varied. For both tasks, we restrict to MHC wild-type constructs, clean dissociation constant (K_D) measurements, and remove mutation sets with fewer than three measured TCR–peptide pairs. After preprocessing, the peptide-mutation test set contains 7 index-peptide scaffolds, 11 distinct paired TCR sequences, 38 distinct peptide sequences, and 58 unique TCR–peptide pairs. The TCR-mutation test set contains 7 wild-type TCR references, 8 index peptides, 75 distinct TCR sequences, and 81 unique TCR–peptide pairs.

Structural contact maps. We use TCR3d (Gowthaman & Pierce, 2019) to evaluate whether LogiCA's contextualized logits recover biologically meaningful residue-level interaction signals. As of January 2026, TCR3d contains ∼250 TCR–pMHC complexes. From these structures, we extract residue-level contact maps and compare them to the cross-modal dependency score in Eq. 6. This analysis tests whether perturbing one sequence component induces logit changes at spatially proximal residues in the paired biological context.

E. Reproducibility

E.1. Protein-ligand LogiCA setup.

Unless noted otherwise, all experiments use the cross-attention interaction tower described in Section 2.4: two bidirectional layers, four attention heads per layer, dropout 0.1, and scalar gate logits initialized to -6.0. For pretraining and DTI fine-tuning, protein and ligand inputs are truncated to 512 and 128 tokens, respectively. For variant ranking, we increase the protein limit to 1024 tokens to cover the oncogene panel used in Appendix D; variants whose mutated residue falls beyond the retained sequence window are excluded from evaluation.

The protein encoder is ESM-2 8M, 35M, 150M, or 650M depending on the experiment, and the ligand encoder is SELFormer. During pretraining, pretrained encoder weights are frozen and only the interaction tower and scoring parameters are optimized. During downstream fine-tuning, we add LoRA adapters (Hu et al., 2022) to the pretrained encoders. Scores are computed from the contextualized token-likelihood interface rather than from pooled-latent similarities.

All experiments were run on NVIDIA H100 80 GB SXM or H200 GPUs. Unless otherwise noted, protein–ligand pretraining uses the same base recipe across ESM-2 backbone sizes: 100 epochs, 4-GPU data parallelism on a single H100 node, per-GPU micro-batch size 4, gradient accumulation 2, AdamW with learning rate 1×10^{-4}, and two negatives in each anchor direction. On 4×H100, the 35M LogiCA model requires approximately 18 GPU-hours for 100-epoch pretraining on the 779k-anchor BindingDB split; larger backbones scale accordingly. Downstream fine-tuning uses one H100 per random seed and requires approximately 1 GPU-hour per dataset seed for the 35M encoder with LoRA rank 8.

The cross-attention adapter dimensions used at each backbone scale are summarized in Table S2 ; layer count, head count, dropout, and gate initialization are shared across scales. Tables S3–S5 summarize the hyperparameters used in the reported experiments.

Table S2. Cross-attention adapter dimensions per backbone scale.

Backbone	d_{protein}	d_s	**Heads** H	**Layers** N
ESM-2 8M	320	320	4	2
ESM-2 35M	480	384	4	2
ESM-2 150M	640	384	4	2
ESM-2 650M	1280	384	4	2

Training of controlled baselines. For controlled ablations, we keep the pretrained backbones, cross-attention architecture, input preprocessing, task data, and optimization schedule matched to LogiCA, and vary only the scoring or training objective. This gives three ablations: conditional masked-language modeling (LogiMLM), latent contrastive alignment (*LatentCA*), and pooled latent fusion (*LatentFuse*).

LogiMLM is the conditional masked-token ablation. It uses the same backbones and cross-attention adapter as LogiCA, preserves the native token heads, and is trained with a standard masked-language-modeling loss only on positive paired examples. Unlike LogiCA, it is not exposed to partner-swapped or mutation-derived negative pairs during pretraining, so it tests whether preserving the token-likelihood interface is sufficient without contrastive alignment.

LatentCA is the latent-contrastive ablation used for drug–target binding. It mean-pools each tower's contextualized hidden states and scores the pair by cosine similarity between the pooled protein and ligand representations. LatentCA is trained with the same InfoNCE/Bradley–Terry objective and negative construction as LogiCA, isolating latent-space contrastive alignment from logit-space token scoring.

LatentFuse is the corresponding latent ablation for drug-conditioned variant ranking. It is initialized from the LatentCA protein–ligand pretrained model, so it inherits the same latent contrastive pretraining as the binding baseline. Because cosine similarity between protein and drug pools is degenerate when every variant in an assay is paired with the same drug, LatentFuse replaces the cosine score with a two-layer MLP over the concatenated mean-pooled protein and drug representations. It is then fine-tuned on the same mutation-local resistance objective as LogiCA, but predicts resistance with a pooled scalar head rather than the native token-likelihood interface.

E.2. Evaluation of protein-ligand external baselines

We evaluated external baselines under a common protocol covering three model families: drug–target interaction models, drug-agnostic protein variant scorers, and structure-based models. Whenever possible, we used the authors' public implementations and default training settings. All supervised baselines were evaluated on the splits described in Appendix D; for mutation-effect experiments, we used leave-one-protein-out splits matching the LOGICA evaluation protocol.

Because most drug–target interaction baselines are designed to predict binding between a drug and a protein, rather than mutation effects under a fixed drug context, we adapted them by extracting the model representation immediately before the final prediction layer and training a small regression MLP head within each leave-one-protein-out fold. Unless otherwise noted, the original backbone was kept fixed. Drug-agnostic protein language models were evaluated zero-shot from the pseudo-log-likelihood difference between mutant and wild type. These baselines do not receive the drug as input, so they test whether general protein fitness or evolutionary plausibility alone explains drug-resistance effects. For MSA-based methods, we built one MSA per gene with `jackhmmer` against UniRef90 (2022_05) and reused the same MSAs across baselines.

MolTrans (Huang et al., 2021) (https://github.com/kexinhuang12345/MolTrans). We evaluated MolTrans using the authors' released implementation and standard tokenization, interaction transformer, and classifier design. The model was trained on the standard DAVIS, BindingDB, and BioSNAP splits, and results were averaged across five random seeds.

ConPLex (Singh et al., 2023) (https://github.com/samsledje/ConPLex). We trained ConPLex with its default protein and molecule featurizers and the released contrastive co-embedding objective. Results were averaged over five seeds.

DrugBAN (Bai et al., 2023) (https://github.com/peizhenbai/DrugBAN). We followed the authors' default configuration, including the graph-based drug encoder, protein encoder, bilinear co-attention module, and prediction head. The model was trained on the same splits and averaged across five seeds. For the mutation-effect task, we additionally evaluated a frozen-backbone DrugBAN variant. We used the representation before the final classifier as the drug–protein embedding and trained a small regression head within each leave-one-protein-out fold.

DrugCLIP (Jia et al., 2026) (https://github.com/bowen-gao/DrugCLIP). We initialized DrugCLIP from the released BindingDB-pretrained checkpoint and evaluated it on each downstream split using the authors' encoder architecture. For the mutation-effect setting, we also report a frozen-backbone variant in which the learned drug and pocket embeddings are passed to a small regression head trained within each leave-one-protein-out fold.

SP-DTI (Liu et al., 2025b) (https://github.com/Steven51516/SP-DTI). SP-DTI requires protein pocket information, so we used AlphaFold2 (Abramson et al., 2024) structures and extracted pockets before training. We then evaluated the released pipeline with the authors' default model configuration and averaged results over five seeds.

GS-DTI (Yu et al., 2025) (https://github.com/purvavideha/GSDTI). GS-DTI uses both molecular graph features and protein structural information. We precomputed the required KPGT (Li et al., 2022) drug features (https://github.com/lihan97/KPGT) and AlphaFold2-based (Abramson et al., 2024) protein features, then trained the released model with its default configuration and averaged results over five seeds.

ESM-1v (Meier et al., 2021) (https://github.com/facebookresearch/esm). We evaluated ESM-1v zero-shot using the masked-marginal pseudo-log-likelihood ratio between mutant and wild type, averaged over the released five-model ensemble. Since ESM-1v has no drug input, the resulting score is independent of drug context.

ESM-2 (Lin et al., 2023) (https://github.com/facebookresearch/esm). We evaluated ESM-2 in the same zero-shot manner as ESM-1v, using the 35M and 150M checkpoints. These scores provide drug-agnostic estimates of variant plausibility.

EVE (Frazer et al., 2021) (https://github.com/OATML-Markslab/EVE). We trained one EVE model per gene using the corresponding MSA and the authors' default training procedure. Variants were scored zero-shot by comparing the model likelihood of the mutant and wild-type sequences.

Tranception (Notin et al., 2022a) (https://github.com/OATML-Markslab/Tranception). We evaluated Tranception-Large with the authors' released scoring script and the same per-gene MSAs used for EVE. As with EVE and ESM, Tranception does not condition on the drug.

Boltz-2 (Passaro et al., 2025) (https://github.com/jwohlwend/boltz). Boltz-2 is a structure-based model that provides learned structural representations for protein–ligand complexes. For each mutant–drug and wild-type–drug pair, we extracted the corresponding Boltz-2 latent embeddings and trained a small regression head on top of them within each leave-one-protein-out fold, while keeping Boltz-2 itself frozen.

E.3. TCR–peptide LogiCA setup.

We use the same overall LogiCA setup as the protein–ligand pipeline, with the following differences. The TCR branch is encoded by TCRLang throughout all experiments while the peptide branch uses ESM-2, with backbone size swept over 8M, 35M, 150M, and 650M. Sequence lengths are padded or truncated to a maximum of 64 tokens for the paired TCR chains and 32 tokens for the peptide. In practice, this maximum completely accommodates the TCRs, where no paired sequence exceeds 64 tokens, leaving the TCR branch entirely untruncated. Peptide truncation is similarly minimal, affecting fewer than 0.5% of the longest sequences in the IEDB pretraining corpus. The cross-attention tower matches the protein–ligand side in layer count and dropout, but the head count is inherited from each backbone rather than fixed at four, and we omit the bidirectional scalar gate. The projection (256-dim) and classifier hidden dimension (512-dim) are shared across all peptide backbone sizes. Unlike the protein–ligand setup, both encoders remain trainable during pretraining. Accordingly, we use a single H100 80 GB SXM GPU with per-GPU micro-batch 16 and no gradient accumulation, AdamW at 1×10^{-5}, and 10 epochs. On a single H100, the TCRLang–ESM-2 35M model requires approximately 10 GPU-hours for 10-epoch pretraining on the 234k positive pairs of pretraining corpus with $K = 1$ negative, and runs with $K = 5$ scale to roughly 14–22 GPU-hours depending on anchor mode.

Downstream fine-tuning requires approximately 1 GPU-hour for 15 epochs on a single H100. In contrast to the protein–ligand recipe, we fully fine-tune both encoders and the cross-attention tower without LoRA adapters. Rather than a Bradley–Terry ranking loss, we use plain BCE to handle the wide divergence between our matched and mismatched pairs. Because our anchored comparisons involve completely different sequences rather than close mutants, optimizing for absolute, independent probabilities via BCE provides a more robust training signal. Variant ranking is performed at inference on the binary fine-tuned head via the masked-token mutant log-likelihood, with no separate ranking optimizer. Tables S6–S8 summarize the hyperparameters used in the reported TCR–epitope experiments.

E.4. Evaluation of TCR–peptide external baselines.

ePytope Benchmark Predictors. (Drost et al., 2025) iTCep (Zhang et al., 2023), TULIP-TCR (Meynard-Piganeau et al., 2024), TEIM (Peng et al., 2023), and ERGO-II (Springer et al., 2021) with its McPAS variant, and ImRex (Moris et al., 2021) with its full variant were evaluated through the ePytope benchmarking pipeline (https://github.com/SchubertLab/benchmark_TCRprediction). Each model was loaded with its publicly released pretrained checkpoints and no retraining was performed. In addition to the DMS dataset provided by ePytope, we integrated three additional benchmarks into the pipeline: BATCAVE, ATLAS-Pep, and ATLAS-TCR. Across all four test sets, we uniformly report Pearson and Spearman correlations between predicted binding scores and measured binding activity or affinity.

EPACT (Zhang et al., 2024a). EPACT was evaluated outside the ePytope pipeline using the official repository (https://github.com/zhangyumeng1sjtu/EPACT). We ran inference with all five publicly released cross-validation checkpoints and averaged predictions across folds prior to computing metrics.

TCR-T5 (Karthikeyan et al., 2025). TCR-T5 was also evaluated using the official repository (https://github.com/pirl-unc/tcr_translate). We loaded the publicly released fine-tuned checkpoint and scored each TCR–pMHC pair as the conditional log-likelihood of the CDR3β sequence conditioned on the epitope and MHC allele.

Table S3. Hyperparameters for released LOGICA protein–ligand pretraining runs.

Config key	Description	Value
`train.optimizer.name`	Optimizer used for pretraining.	`AdamW`
`train.optimizer.lr`	Learning rate for pretraining updates.	1×10^{-4}
`train.optimizer.weight_decay`	Weight decay applied during pretraining.	1×10^{-2}
`train.batch_size_per_gpu`	Microbatch size on each GPU.	4
`train.num_gpus`	Number of GPUs used for each pretraining run.	4
`train.grad_accum_steps`	Gradient-accumulation steps.	2
`train.effective_anchor_batch`	Raw anchor examples per optimizer update: batch_size_per_gpu × num_gpus × grad_accum_steps.	$4 \times 4 \times 2 = 32$
`train.freeze_encoders`	Freeze the pretrained protein and drug encoders.	`true`
`objective.contrastive.temperature`	InfoNCE temperature (τ).	0.1
`data.mask_rate`	Masked-token corruption rate for the auxiliary loss.	0.15
`data.negatives_per_anchor`	Negative samples per anchor in LOGICA pretraining.	2
`objective.scored_pair_contexts`	Pair contexts scored per optimizer update, counting one positive pair plus K protein-anchored and K ligand-anchored negatives per anchor.	$32 \times (1 + 2K) = 160$
`train.num_epochs`	Pretraining epochs per scaling run, shared across all backbone sizes.	100
`train.lr_scheduler.name`	Learning-rate schedule.	`none`
`train.lr_scheduler.warmup_steps`	Warmup steps for the learning-rate schedule.	0

Table S4. Hyperparameters for downstream DTI fine-tuning.

Config key	Description	Value
`train.optimizer.name`	Optimizer used for DTI fine-tuning.	`AdamW`
`train.optimizer.lr`	Learning rate for fine-tuning.	2×10^{-5}
`train.optimizer.weight_decay`	Weight decay applied during fine-tuning.	1×10^{-2}
`train.batch_size`	Per-step batch size (B).	4
`train.eval_batch_size`	Evaluation batch size.	16
`train.grad_accum_steps`	Gradient-accumulation steps (effective batch 8).	2
`model.lora_r`	LoRA rank applied to the protein tower.	64
`model.lora_alpha`	LoRA alpha.	128
`model.lora_dropout`	LoRA dropout.	0.1
`data.negatives_per_positive`	Default sampled negatives per positive.	1
`objective.bt.weight_mode`	Bradley–Terry negative-class weighting.	`ratio`
`train.num_epochs`	Number of fine-tuning epochs.	40
`train.lr_scheduler.name`	Learning-rate schedule.	`none`
`train.lr_scheduler.warmup_steps`	Warmup steps for the learning-rate schedule.	0

Table S5. Hyperparameters for variant-ranking fine-tuning on protein-drug resistance.

Config key	Description	Value
`train.optimizer.name`	Optimizer used for variant-ranking updates.	`AdamW`
`train.optimizer.lr`	Learning rate for variant-ranking fine-tuning.	3×10^{-4}
`train.optimizer.weight_decay`	Weight decay applied during fine-tuning.	1×10^{-2}
`train.batch_size`	Per-step batch size (B).	4
`train.eval_batch_size`	Evaluation batch size.	64
`train.grad_accum_steps`	Gradient-accumulation steps (effective batch 8).	2
`model.lora_r`	LoRA rank applied to the protein tower.	8
`model.lora_alpha`	LoRA alpha.	16
`model.lora_dropout`	LoRA dropout.	0.1
`train.scoring`	Variant scoring rule before Bradley–Terry ranking.	`drug_llr`
`train.pair_weighting`	Weighting on within-batch pairs by $\lvert\Delta f\rvert$.	`delta`
`train.num_epochs`	Number of fine-tuning epochs.	40
`train.lr_scheduler.name`	Learning-rate schedule.	`none`
`train.lr_scheduler.warmup_steps`	Warmup steps for the learning-rate schedule.	0

Table S6. Hyperparameters for released LogiCA TCR–peptide pretraining runs.

Config key	Description	Value
`train.optimizer.name`	Optimizer used for pretraining.	`AdamW`
`train.optimizer.lr`	Learning rate for pretraining updates.	1×10^{-5}
`train.optimizer.weight_decay`	Weight decay applied during pretraining.	1×10^{-2}
`train.batch_size_per_gpu`	Microbatch size on each GPU.	16
`train.num_gpus`	Number of GPUs used for each pretraining run.	1
`train.grad_accum_steps`	Gradient-accumulation steps.	1
`train.effective_anchor_batch`	Raw anchor examples per optimizer update: batch_size_per_gpu × num_gpus × grad_accum_steps.	$16 \times 1 \times 1 = 16$
`train.freeze_encoders`	Freeze the pretrained TCR and peptide encoders.	`false`
`objective.contrastive.temperature`	InfoNCE temperature (τ).	0.1
`data.mask_rate`	Masked-token corruption rate for the auxiliary MLM loss.	0.15
`data.negatives_per_anchor`	Negative samples per pep-anchored batch.	1 / 5
`data.anchor_type`	Anchor side for each released model.	`a` / `b` / `mixed`
`data.anchor_batch_ratio_pep`	Ratio of peptide-anchored to TCR-anchored batches (`LogiCA-Dual` only).	5
`objective.scored_pair_contexts`	Pair contexts scored per optimizer update, counting one positive plus K negatives per anchor.	$16 \times (1+1) = 32$ / $16 \times (1+5) = 96$
`train.use_mlm_loss`	Add masked-LM auxiliary loss on both branches.	`true`
`train.mlm_loss_weight`	Weight on the auxiliary MLM term.	1.0
`train.num_epochs`	Pretraining epochs per run, shared across all backbone sizes.	10
`train.lr_scheduler.name`	Learning-rate schedule.	`linear`
`train.lr_scheduler.warmup_steps`	Warmup steps for the learning-rate schedule.	1,000

Table S7. Pretraining hyperparameters that differ across the three released LogiCA TCR–peptide models. All other hyperparameters are shared and listed in Table S6.

Hyperparameter	LogiCA-TCR	LogiCA-Pep	LogiCA-Dual
`data.anchor_type`	`a` (TCR)	`b` (Peptide)	`mixed`
`data.negatives_per_anchor` (K_{pep})	none	5	5
`data.negatives_per_anchor_a` (K_{tcr})	1	`none`	1
`data.anchor_batch_ratio_pep`	`none`	`none`	5
`objective.scored_pair_contexts`	$16 \times 2 = 32$	$16 \times 6 = 96$	32 / 96

Table S8. Hyperparameters for downstream TCR–peptide fine-tuning.

Config key	Description	Value
`train.optimizer.name`	Optimizer used for fine-tuning.	`AdamW`
`train.optimizer.lr`	Learning rate for fine-tuning.	2×10^{-5}
`train.optimizer.weight_decay`	Weight decay applied during fine-tuning.	1×10^{-2}
`train.batch_size`	Per-step batch size (B).	16
`train.eval_batch_size`	Evaluation batch size.	32
`train.grad_accum_steps`	Gradient-accumulation steps (effective batch 16).	1
`train.freeze_encoders`	Freeze the pretrained TCR and peptide encoders.	`false`
`train.freeze_cross_attention`	Freeze the cross-attention tower.	`false`
`train.l2_reg_lambda`	ℓ_2 regularization on classifier weights.	1×10^{-2}
`objective.loss`	Classification loss on the binary head.	`BCE`
`train.num_epochs`	Number of fine-tuning epochs.	15
`train.lr_scheduler.name`	Learning-rate schedule.	`linear`
`train.lr_scheduler.warmup_steps`	Warmup steps for the learning-rate schedule.	500

F. Additional Experiments

F.1. DTI fine-tuning ablation

Table S9 ablates the w/ LOGICA fine-tuning recipe along two axes: protein-tower size {8M, 35M, 150M} at fixed LoRA rank r=64 (top block), and LoRA rank $r \in \{8, 32, 64\}$ at fixed 35M backbone (bottom block). All rows use the same downstream DTI fine-tuning learning rate, η=2×10^{-5}, matching Table S4. Increasing the protein tower to 150M improves BindingDB and is the configuration adopted in Table 1; increasing LoRA rank from 8 to 64 at the 35M backbone yields a small monotonic improvement.

Table S9. Ablation of w/ LOGICA fine-tuning hyperparameters. The highlighted 150M row is the configuration reported in Table 1. All cells are five-seed means; standard deviations are reported in the main table for the highlighted configuration.

Method	DAVIS		BindingDB (Test)		BioSNAP	
	AUC	AUPR	AUC	AUPR	AUC	AUPR
Backbone scaling at fixed LoRA rank r=64						
w/ LOGICA†(8M)	0.920	0.425	0.891	0.614	0.914	0.921
w/ LOGICA†(35M)	0.919	0.433	0.894	0.608	0.920	0.927
w/ LOGICA†(150M)	**0.924**	**0.446**	**0.906**	**0.635**	**0.920**	**0.927**
LoRA rank ablation at fixed 35M backbone						
w/ LOGICA†(35M, r=8)	0.916	0.426	0.886	0.595	0.913	0.920
w/ LOGICA†(35M, r=32)	0.919	0.430	0.888	0.600	0.916	0.924
w/ LOGICA†(35M, r=64)	0.919	0.433	0.894	0.608	0.920	0.927

F.2. Variant ranking: few-shot adaptation

Table S10 reports the few-shot adaptation curves for the variant-ranking benchmark. For each target gene, a fraction $f \in \{0, 5, 10, 15\}\%$ of its labeled (variant, drug, resistance score) entries is used for adaptation and the remaining variants are held out for evaluation, contrasting the w/ LOGIMLM and w/ LOGICA pretrains at two backbone scales (8M and 35M). Both initializations improve monotonically with the target-label fraction and w/ LOGICA stays above w/ LOGIMLM at every fraction, with the largest separation at the higher end. The 35M, 15% row is the configuration referenced as LOGICA (15% target labels) in the few-shot scaling figure (Figure 2c).

Table S10. Few-shot adaptation on the variant-ranking benchmark. For each target gene, a small fraction of its labeled (variant, drug, resistance score) entries is used for adaptation and the remaining variants are held out for evaluation. Each column reports the 11-gene leave-one-protein-out average Spearman ρ and binary resistance AUC at the indicated target-label fraction. Best values within each backbone size are shown in **bold**. The 0% setting corresponds to the rows reported in Table 2.

Method	0% target labels		5% target labels		10% target labels		15% target labels	
	ρ	AUC	ρ	AUC	ρ	AUC	ρ	AUC
Few-shot target-gene adaptation								
w/ LOGIMLM†(8M)	0.279	0.624	0.346	0.659	0.427	0.699	0.471	0.713
w/ LOGICA†(8M)	**0.297**	**0.629**	**0.377**	**0.670**	**0.443**	**0.709**	**0.489**	**0.728**
w/ LOGIMLM†(35M)	0.271	0.615	0.372	0.661	0.412	0.666	0.478	0.693
w/ LOGICA†(35M)	**0.296**	**0.636**	**0.381**	**0.669**	**0.460**	**0.719**	**0.501**	**0.741**

F.3. Two scaling regimes for LOGICA

We analyze two forms of scaling: pretraining scale, measured by matched-versus-mismatched likelihood separation, and downstream data scale, measured by few-shot variant-ranking performance. For pretraining, we use the held-out symmetric

likelihood margin

$$\hat{\bar{\gamma}} = \frac{1}{2}\left[\ell(x \mid y) - \frac{1}{K}\sum_{k=1}^{K}\ell(x \mid y_k^-) + \ell(y \mid x) - \frac{1}{K}\sum_{k=1}^{K}\ell(y \mid x_k^-)\right], \tag{18}$$

averaged over held-out matched pairs and sampled negatives. We pretrain LogiCA with ESM-2 (Lin et al., 2023) backbones from 8M to 650M parameters while keeping the SELFormer (Yüksel et al., 2023) ligand tower and training recipe fixed.

F.4. TCR–epitope ESM-2 peptide encoder scaling

Table S11 evaluates the effect of ESM-2 peptide encoder size for LogiCA-TCR. All variants use the same TCRlang-Paired TCR encoder, pretraining objective, and fine-tuning recipe, and differ only in the ESM-2 peptide encoder size. The 35M-parameter encoder performs best across all three peptide-mutation benchmarks. Larger encoders do not improve performance: the 150M model is competitive but lower than 35M, while the 650M model drops further across benchmarks. This suggests that ESM-2 35M is best matched to the scale of our TCR–epitope data, so we use it as the peptide encoder in the main experiments.

Table S11. Experiments with different ESM-2 peptide encoder sizes in LogiCA-TCR. All variants use TCRLang-Paired as the TCR encoder and differ only in the size of the ESM-2 peptide encoder. Performance is reported as mean Pearson and Spearman correlations with standard deviation across runs on ePytope, BATCAVE, and ATLAS-PEP peptide-mutation benchmarks.

Method	ePytope (Drost et al., 2025)		BATCAVE (Banerjee et al., 2025)		ATLAS-PEP (Borrman et al., 2017)	
	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman
Contextualized backbone						
LogiCA-TCR ESM-2 8M peptide encoder	0.098 ± 0.237	0.063 ± 0.230	0.109 ± 0.229	0.074 ± 0.225	0.286 ± 0.683	0.261 ± 0.561
LogiCA-TCR **ESM-2 35M peptide encoder**	**0.296** ± 0.219	**0.229** ± 0.168	**0.223** ± 0.229	**0.170** ± 0.191	**0.632** ± 0.286	**0.731** ± 0.238
LogiCA-TCR ESM-2 150M peptide encoder	0.209 ± 0.211	0.194 ± 0.193	0.165 ± 0.199	0.144 ± 0.192	0.527 ± 0.260	0.609 ± 0.272
LogiCA-TCR ESM-2 650M peptide encoder	0.078 ± 0.207	0.092 ± 0.213	0.064 ± 0.185	0.066 ± 0.192	0.471 ± 0.395	0.533 ± 0.381

F.5. Binary classification on thresholded ePytope labels

Table S15 reports a supplementary binary classification analysis on ePytope using the benchmark-provided labels derived from epitope-specific NFAT thresholds. This evaluation tests whether the variant scores used for mutation ranking also separate the thresholded active and inactive TCR–epitope pairs within this mutation dataset. In addition to AUC and AUPR, we report AUC0.1, BEDROC, and logAUC as early-retrieval metrics that emphasize whether active pairs are ranked near the top of the prediction list. LogiCA-TCR performs best on most metrics, including AUC (0.672), AUPR (0.485), AUC0.1 (0.586), and BEDROC (0.541), while LogiCA-Dual achieves the best logAUC (0.093) and remains competitive across the other metrics.

G. Computational complexity of LogiCA

LOGICA is a token-likelihood-preserving interaction model, not a replacement for scalable dual-encoder retrieval. Its main computational cost comes from pair-specific conditioning. For N sequences and M contexts, a dual encoder can precompute independent representations with encoder cost $O(N + M)$, then score pairs by inexpensive dot products or cosine similarities. In contrast, LOGICA must jointly contextualize each queried sequence–context pair before evaluating token likelihoods, giving $O(NM)$ pair-specific evaluations for exhaustive all-by-all retrieval. This makes LOGICA less suitable as a first-stage model for massive screening, but enables context-conditioned native token likelihoods, mutation-local scoring, token-level interpretation, and conditional generation.

Training complexity. Dual-encoder contrastive models and LOGICA can use the same positive pairs, in-batch negatives, and sampled anchored negatives. The difference is how the $O(BK)$ anchor–candidate scores are computed for a minibatch with B anchors and K candidate partners. A dual encoder computes $O(B + K)$ independent representations and forms all BK scores by matrix multiplication. LOGICA instead requires a pair-specific interaction for each anchor–candidate score, so the interaction cost scales as $O(BK)$. In practice, this cost is controlled by the sampled regime used here, where each anchor is contrasted against a small set of positives and negatives rather than all possible partners.

Paired or conditional MLMs have a similar pair-specific computation structure, because each sequence–context input must be evaluated jointly. However, they optimize token reconstruction on matched pairs, whereas LOGICA explicitly contrasts matched and mismatched contexts.

Inference complexity. At inference time, the gap is largest for large candidate libraries. Dual encoders can reuse precomputed representations and scale naturally to high-throughput retrieval. LOGICA and paired MLMs must evaluate each queried pair jointly, so exhaustive retrieval over N sequences and M contexts requires $O(NM)$ paired evaluations. These models are therefore better used after candidate narrowing, or in settings where the context is fixed and only a finite set of variants is scored.

For variant ranking, the cost is often modest. Given one context and V variants, LOGICA requires $O(V)$ paired evaluations rather than an all-by-all search. The final likelihood-ratio score can also be restricted to the mutated positions, preserving the mutation-local structure of the prediction.

Generation complexity. Dual encoders score completed inputs but do not natively define normalized token distributions for conditional generation. LOGICA retains the language model token head, so it can generate under a fixed context y by Gibbs sampling over a design set A:

$$x_i \sim \pi_\theta(\cdot \mid x_{\setminus i}, y), \qquad i \in A.$$

With T updates, generation costs approximately $O(T)$ contextualized forward passes. This is more expensive than retrieval in a precomputed latent space, but it enables direct context-conditioned token generation.

H. Supplementary Tables

Table S12. Drug-resistance variant scoring, per-gene breakdown of Table 2. Each cell is the cross-drug mean ± std of held-out Spearman ρ and binary resistance AUC for the indicated gene fold (LOPO over 11 oncogenes). The EGFR fold corresponds to the Kim et al. (Kim et al., 2025) prime-editing source; the remaining ten folds (KRAS – PARP2) come from the Coelho et al. (Coelho et al., 2024) multi-oncogene screen. The **Avg.** column reports cross-gene mean ± std of pooled per-gene metrics over all 11 LOPO folds. Best per column is in **bold**; second-best is underlined. The † marker denotes methods that retain a native-vocabulary generative interface.

Method	EGFR		KRAS		BRAF		MAP2K1		MAP2K2		PIK3CA	
	ρ	AUC	ρ	AUC	ρ	AUC	ρ	AUC	ρ	AUC	ρ	AUC
Contextualized backbone (fine-tuned)												
w/ LatentFuse (35M) concat embeddings + MLP	0.029 ± 0.021	0.519 ± 0.016	0.044 ± 0.079	0.496 ± 0.068	0.029 ± 0.127	0.523 ± 0.080	0.195 ± 0.064	0.581 ± 0.032	0.207 ± 0.063	0.616 ± 0.039	0.066 ± 0.058	0.527 ± 0.034
w/ LatentFuse (150M) concat embeddings + MLP	0.019 ± 0.067	0.511 ± 0.030	0.132 ± 0.107	0.570 ± 0.078	-0.034 ± 0.059	0.485 ± 0.035	0.149 ± 0.078	0.581 ± 0.034	0.148 ± 0.041	0.577 ± 0.020	0.027 ± 0.066	0.511 ± 0.044
w/ LogiMLM†(35M) logit-level MLM	0.245 ± 0.064	0.620 ± 0.031	0.050 ± 0.124	0.519 ± 0.094	0.202 ± 0.074	0.610 ± 0.051	0.339 ± 0.094	0.673 ± 0.047	0.245 ± 0.107	0.636 ± 0.059	0.229 ± 0.047	0.616 ± 0.039
w/ LogiMLM†(150M) logit-level MLM	0.272 ± 0.083	0.633 ± 0.048	0.199 ± 0.114	0.566 ± 0.065	0.221 ± 0.078	0.616 ± 0.050	0.351 ± 0.088	0.676 ± 0.054	0.292 ± 0.118	0.639 ± 0.062	**0.343** ± 0.050	**0.673** ± 0.024
w/ LogiCA†(35M) **token-score contrastive**	0.260 ± 0.067	0.630 ± 0.034	0.148 ± 0.057	0.581 ± 0.032	**0.242** ± 0.050	**0.621** ± 0.033	0.320 ± 0.078	0.664 ± 0.045	0.274 ± 0.113	0.646 ± 0.058	0.219 ± 0.041	0.616 ± 0.024
w/ LogiCA†(150M) **token-score contrastive**	**0.295** ± 0.074	**0.638** ± 0.046	**0.237** ± 0.112	**0.623** ± 0.072	0.210 ± 0.064	0.612 ± 0.046	**0.407** ± 0.084	**0.703** ± 0.042	**0.313** ± 0.108	**0.666** ± 0.061	0.264 ± 0.043	0.636 ± 0.030
Unconditional baselines												
ESM-1v (Meier et al., 2021) masked LM	0.195 ± 0.088	0.601 ± 0.041	-0.058 ± 0.126	0.489 ± 0.084	0.164 ± 0.069	0.580 ± 0.045	0.121 ± 0.086	0.562 ± 0.062	0.083 ± 0.090	0.560 ± 0.045	0.124 ± 0.034	0.566 ± 0.030
ESM-2 (Lin et al., 2023) (35M) masked LM	0.155 ± 0.094	0.577 ± 0.046	-0.168 ± 0.081	0.404 ± 0.062	0.100 ± 0.082	0.558 ± 0.047	0.078 ± 0.067	0.550 ± 0.051	0.080 ± 0.080	0.568 ± 0.043	0.083 ± 0.041	0.536 ± 0.022
ESM-2 (Lin et al., 2023) (150M) masked LM	0.153 ± 0.105	0.580 ± 0.046	-0.037 ± 0.098	0.472 ± 0.083	0.088 ± 0.080	0.553 ± 0.039	0.084 ± 0.083	0.553 ± 0.060	0.107 ± 0.093	0.582 ± 0.047	0.117 ± 0.051	0.563 ± 0.025
EVE (Frazer et al., 2021) MSA VAE	0.156 ± 0.081	0.580 ± 0.033	-0.049 ± 0.082	0.499 ± 0.058	0.111 ± 0.117	0.553 ± 0.062	0.044 ± 0.130	0.514 ± 0.077	0.062 ± 0.097	0.547 ± 0.056	0.031 ± 0.110	0.505 ± 0.044
Tranception (Notin et al., 2022a) retrieval LM	0.170 ± 0.062	0.591 ± 0.031	-0.058 ± 0.099	0.500 ± 0.061	0.088 ± 0.091	0.538 ± 0.057	0.015 ± 0.077	0.507 ± 0.063	0.076 ± 0.065	0.545 ± 0.033	0.119 ± 0.029	0.556 ± 0.021
Contextualized baselines (fine-tuned, 10-gene LOPO)												
DrugBAN (Bai et al., 2023) DTI classifier + MLP	0.018 ± 0.036	0.507 ± 0.025	-0.078 ± 0.107	0.439 ± 0.077	-0.012 ± 0.039	0.492 ± 0.024	0.023 ± 0.040	0.484 ± 0.031	0.042 ± 0.059	0.523 ± 0.032	0.037 ± 0.073	0.523 ± 0.035
Boltz-2 (Passaro et al., 2025) structure features + MLP	0.011 ± 0.047	0.506 ± 0.020	0.094 ± 0.145	0.571 ± 0.076	0.004 ± 0.062	0.476 ± 0.037	0.053 ± 0.070	0.533 ± 0.037	0.022 ± 0.061	0.501 ± 0.043	-0.009 ± 0.082	0.494 ± 0.051
DrugCLIP (Jia et al., 2026) DTI contrastive + MLP	-0.000 ± 0.040	0.497 ± 0.019	0.055 ± 0.154	0.520 ± 0.124	0.028 ± 0.099	0.503 ± 0.043	-0.014 ± 0.079	0.500 ± 0.040	-0.014 ± 0.053	0.503 ± 0.037	-0.015 ± 0.062	0.498 ± 0.037

Method	AKT1		MYC		BCL2		PARP1		PARP2		Avg.	
	ρ	AUC	ρ	AUC	ρ	AUC	ρ	AUC	ρ	AUC	ρ	AUC
Contextualized backbone (fine-tuned)												
w/ LatentFuse (35M) concat embeddings + MLP	0.088 ± 0.044	0.541 ± 0.029	0.123 ± 0.092	0.576 ± 0.068	0.129 ± 0.088	0.582 ± 0.067	0.064 ± 0.067	0.537 ± 0.045	-0.085 ± 0.053	0.459 ± 0.031	0.083 ± 0.083	0.542 ± 0.046
w/ LatentFuse (150M) concat embeddings + MLP	0.031 ± 0.065	0.503 ± 0.042	0.004 ± 0.099	0.516 ± 0.063	0.079 ± 0.064	0.562 ± 0.052	-0.016 ± 0.028	0.494 ± 0.017	0.015 ± 0.088	0.492 ± 0.056	0.050 ± 0.065	0.531 ± 0.037
w/ LogiMLM†(35M) logit-level MLM	0.253 ± 0.079	0.652 ± 0.042	**0.317** ± 0.089	**0.670** ± 0.054	0.134 ± 0.015	0.585 ± 0.027	0.148 ± 0.083	0.574 ± 0.029	0.196 ± 0.062	0.594 ± 0.029	0.220 ± 0.081	0.615 ± 0.045
w/ LogiMLM†(150M) logit-level MLM	0.255 ± 0.078	0.651 ± 0.045	0.262 ± 0.034	0.630 ± 0.045	0.234 ± 0.075	0.650 ± 0.069	0.152 ± 0.083	0.583 ± 0.032	0.236 ± 0.056	0.627 ± 0.047	0.260 ± 0.057	0.632 ± 0.032
w/ LogiCA†(35M) **token-score contrastive**	0.301 ± 0.091	**0.676** ± 0.059	0.297 ± 0.104	0.658 ± 0.071	**0.316** ± 0.074	**0.701** ± 0.052	**0.210** ± 0.089	**0.610** ± 0.038	0.200 ± 0.082	0.592 ± 0.061	0.256 ± 0.051	0.636 ± 0.034
w/ LogiCA†(150M) **token-score contrastive**	**0.314** ± 0.085	0.674 ± 0.049	0.302 ± 0.057	0.660 ± 0.035	0.232 ± 0.066	0.640 ± 0.061	0.204 ± 0.103	0.595 ± 0.039	0.224 ± 0.064	0.641 ± 0.046	**0.276** ± 0.062	**0.645** ± 0.031
Unconditional baselines												
ESM-1v (Meier et al., 2021) masked LM	0.094 ± 0.034	0.548 ± 0.022	0.146 ± 0.041	0.577 ± 0.031	0.111 ± 0.062	0.558 ± 0.049	-0.009 ± 0.092	0.487 ± 0.041	-0.021 ± 0.026	0.483 ± 0.032	0.093 ± 0.094	0.549 ± 0.049
ESM-2 (Lin et al., 2023) (35M) masked LM	0.022 ± 0.047	0.513 ± 0.023	0.010 ± 0.062	0.509 ± 0.044	0.037 ± 0.068	0.547 ± 0.044	0.001 ± 0.074	0.488 ± 0.038	-0.034 ± 0.064	0.466 ± 0.057	0.040 ± 0.090	0.523 ± 0.054
ESM-2 (Lin et al., 2023) (150M) masked LM	0.062 ± 0.045	0.537 ± 0.019	0.029 ± 0.047	0.521 ± 0.039	0.041 ± 0.069	0.545 ± 0.041	-0.019 ± 0.128	0.476 ± 0.055	-0.009 ± 0.048	0.485 ± 0.056	0.064 ± 0.067	0.538 ± 0.040
EVE (Frazer et al., 2021) MSA VAE	0.031 ± 0.079	0.523 ± 0.045	0.272 ± 0.051	0.649 ± 0.038	0.114 ± 0.103	0.573 ± 0.047	-0.009 ± 0.074	0.505 ± 0.039	**0.502** ± 0.197	**0.788** ± 0.246	0.121 ± 0.149	0.572 ± 0.090
Tranception (Notin et al., 2022a) retrieval LM	0.061 ± 0.032	0.530 ± 0.021	0.032 ± 0.044	0.513 ± 0.036	0.056 ± 0.069	0.562 ± 0.054	0.027 ± 0.073	0.506 ± 0.042	-0.057 ± 0.086	0.465 ± 0.057	0.059 ± 0.078	0.535 ± 0.041
Contextualized baselines (fine-tuned, 10-gene LOPO)												
DrugBAN (Bai et al., 2023) DTI classifier + MLP	0.017 ± 0.038	0.513 ± 0.019	0.071 ± 0.057	0.540 ± 0.041	0.089 ± 0.038	0.595 ± 0.048	-0.048 ± 0.050	0.477 ± 0.021	-0.111 ± 0.038	0.443 ± 0.027	0.000 ± 0.048	0.511 ± 0.035
Boltz-2 (Passaro et al., 2025) structure features + MLP	0.007 ± 0.047	0.493 ± 0.047	-0.059 ± 0.051	0.479 ± 0.027	0.003 ± 0.040	0.484 ± 0.026	0.013 ± 0.063	0.515 ± 0.048	0.007 ± 0.116	0.499 ± 0.048	0.011 ± 0.034	0.504 ± 0.021
DrugCLIP (Jia et al., 2026) DTI contrastive + MLP	0.011 ± 0.083	0.505 ± 0.060	0.026 ± 0.073	0.506 ± 0.037	-0.072 ± 0.094	0.492 ± 0.059	-0.013 ± 0.048	0.496 ± 0.030	0.011 ± 0.096	0.511 ± 0.041	0.003 ± 0.033	0.506 ± 0.011

Table S13. Models performance comparison on the IMMREP25 unseen epitopes setting (Richardson et al., 2026).

Method	IMMREP25	
	AUC	APS
Contextualized backbone		
w/ LogiMLM† continued MLM	0.499 ± 0.073	0.116 ± 0.046
w/ LogiCA-TCR† **TCR-anchored token scoring**	0.498 ± 0.065	0.110 ± 0.017
w/ LogiCA-Pep† **peptide-anchored token scoring**	0.500 ± 0.073	0.111 ± 0.026
w/ LogiCA-Dual† **dual-anchor token scoring**	0.499 ± 0.062	0.111 ± 0.019
External baselines		
EPACT embedding contrastive	0.492 ± 0.057	0.111 ± 0.022
ERGO-II classifier head	0.498 ± 0.039	0.105 ± 0.010
ImRex classifier head	0.503 ± 0.064	0.110 ± 0.023
iTCep classifier head	0.505 ± 0.058	0.113 ± 0.019
TCR-T5† generative MLM	0.501 ± 0.084	0.121 ± 0.036
TEIM classifier head	0.513 ± 0.072	0.113 ± 0.021
TULIP-TCR† generative MLM	0.506 ± 0.060	0.114 ± 0.019

Table S14. Dependency-map prediction performance for inter-chain interactions, evaluated against ground-truth contact maps. Best values are shown in **bold**, and second-best values are underlined.

Region	Model	AUC	AUPR
CDR3α–CDR3β	ESM-2 (Lin et al., 2023)	0.558 ± 0.227	0.027 ± 0.027
	TULIP-TCR (Meynard-Piganeau et al., 2024)	0.449 ± 0.190	0.017 ± 0.009
	TCRlang (Olsen et al., 2024)	0.502 ± 0.220	0.035 ± 0.114
	w/ LogiMLM	**0.667 ± 0.214**	**0.062 ± 0.088**
	w/ LogiCA-TCR	0.633 ± 0.190	0.054 ± 0.083
	w/ LogiCA-Pep	0.600 ± 0.195	0.031 ± 0.028
	w/ LogiCA-Dual	0.578 ± 0.196	0.034 ± 0.060
Peptide–CDR3α	ESM-2 (Lin et al., 2023)	0.493 ± 0.212	0.039 ± 0.055
	TULIP-TCR (Meynard-Piganeau et al., 2024)	0.531 ± 0.243	0.041 ± 0.051
	w/ LogiMLM	0.553 ± 0.193	0.039 ± 0.054
	w/ LogiCA-TCR	**0.592 ± 0.178**	0.047 ± 0.114
	w/ LogiCA-Pep	0.544 ± 0.239	0.046 ± 0.113
	w/ LogiCA-Dual	0.503 ± 0.250	**0.049 ± 0.126**
Peptide–CDR3β	ESM-2 (Lin et al., 2023)	0.451 ± 0.224	0.031 ± 0.046
	TULIP-TCR (Meynard-Piganeau et al., 2024)	0.455 ± 0.189	0.024 ± 0.021
	TCR-T5 (Karthikeyan et al., 2025)	0.648 ± 0.243	0.077 ± 0.108
	w/ LogiMLM	0.717 ± 0.153	0.058 ± 0.065
	w/ LogiCA-TCR	**0.743 ± 0.131**	0.075 ± 0.145
	w/ LogiCA-Pep	0.712 ± 0.169	0.063 ± 0.073
	w/ LogiCA-Dual	0.733 ± 0.166	**0.094 ± 0.162**

Table S15. Binary classification performance on the ePytope benchmark. The blue block compares contextualized sequence-model variants, including LogiMLM and LogiCA variants; the gray block lists external baselines. The † marker denotes methods that retain a native-vocabulary generative interface. Performance is reported as mean ± standard deviation across ePytope mutation sets. Best values are shown in bold, and second-best values are underlined.

Method	ePytope binary classification				
	AUC	AUPR	AUC0.1	BEDROC	logAUC
Contextualized backbone					
w/ LogiMLM† continued MLM	0.524 ± 0.134	0.341 ± 0.123	0.531 ± 0.045	0.345 ± 0.181	0.024 ± 0.022
w/ LogiCA-TCR† **TCR-anchored token scoring**	**0.672** ± 0.158	**0.485** ± 0.151	**0.586** ± 0.083	**0.541** ± 0.250	<u>0.080</u> ± 0.056
w/ LogiCA-Pep† **peptide-anchored token scoring**	0.586 ± 0.169	0.389 ± 0.133	0.540 ± 0.047	0.416 ± 0.205	0.037 ± 0.027
w/ LogiCA-Dual† **dual-anchor token scoring**	<u>0.633</u> ± 0.189	<u>0.466</u> ± 0.154	<u>0.574</u> ± 0.070	<u>0.523</u> ± 0.283	**0.093** ± 0.068
External baselines					
EPACT (Zhang et al., 2024a) embedding contrastive	0.524 ± 0.140	0.309 ± 0.185	0.498 ± 0.030	0.245 ± 0.229	0.019 ± 0.021
ERGO-II (Springer et al., 2021) classifier head	0.506 ± 0.089	0.317 ± 0.191	0.509 ± 0.031	0.311 ± 0.272	0.020 ± 0.020
ImRex (Moris et al., 2021) classifier head	0.563 ± 0.135	0.335 ± 0.200	0.507 ± 0.050	0.272 ± 0.250	0.019 ± 0.027
iTCep (Zhang et al., 2023) classifier head	0.612 ± 0.148	0.392 ± 0.194	0.543 ± 0.046	0.428 ± 0.227	0.041 ± 0.037
TCR-T5† (Karthikeyan et al., 2025) generative MLM	0.502 ± 0.121	0.312 ± 0.180	0.507 ± 0.032	0.302 ± 0.216	0.024 ± 0.022
TEIM (Peng et al., 2023) classifier head	0.554 ± 0.092	0.332 ± 0.177	0.516 ± 0.037	0.329 ± 0.265	0.031 ± 0.025
TITAN (Weber et al., 2021) classifier head	0.567 ± 0.102	0.327 ± 0.162	0.502 ± 0.028	0.279 ± 0.214	0.017 ± 0.019
TULIP-TCR† (Meynard-Piganeau et al., 2024) generative MLM	0.591 ± 0.113	0.403 ± 0.138	0.551 ± 0.062	0.431 ± 0.227	0.056 ± 0.043

Table S16. Core notation for LOGICA scoring and ranking.

Symbol	Description
Sequences and contexts	
x	Query sequence in its native vocabulary, such as a protein, TCR, or peptide.
y	External context, such as a binding partner, ligand, drug, therapy, or tokenized condition.
x^{wt}	Wild-type reference sequence for variant ranking.
L	Sequence length of x; $[L] = \{1, \dots, L\}$.
a, $\mathcal{C}$	Anchor and candidate set in the contextual ranking template (Eq. 1); a may be a sequence or context, $\mathcal{C}$ is the set of competing candidates.
Token likelihoods and site sets	
$\pi_\theta(x_i \mid x_{\setminus i}, y)$	Contextualized probability assigned to the observed token x_i when site i is masked, under context y.
$\pi_{\theta,i}(\cdot \mid x_{\setminus i}, y)$	Full categorical distribution at site i over the native vocabulary; used in the cross-modal dependency map (Eq. 6).
$A \subseteq [L]$	Scored site set. For full-sequence interaction scoring, $A = [L]$; for variant scoring, $A = \mathcal{M}$.
A_x, A_y	Per-modality scored token positions used in the bidirectional score s_α (Eq. 4).
$\ell_A(x \mid y)$	Site-averaged log-likelihood over A: $\frac{1}{\lvert A\rvert}\sum_{i\in A} \log \pi_\theta(x_i \mid x_{\setminus i}, y)$.
$\mathcal{M}(x, x^{\mathrm{wt}})$	Mutation set $\{i : x_i \neq x_i^{\mathrm{wt}}\}$ for substitution-only variants (x and x^{wt} share length); throughout the proofs, $m = \lvert\mathcal{M}\rvert$.
$\mathcal{A}_j^y$	Substitution alphabet at context position j used to perturb the context token y_j in the dependency map.
Task scores and diagnostics	
$s(a, c)$	Generic compatibility score in the contextual ranking template; instantiated by s_α for interaction scoring or $s_{\mathcal{M}}$ for variant scoring.
$s_{\text{LogiCA}}(x, y; A)$	Site-averaged LOGICA score $\ell_A(x \mid y)$ (Eq. 2); reduces to the directional log-likelihood under the chosen site set.
$s_\alpha(x, y)$	Bidirectional interaction score (Eq. 4): $\alpha\, \ell_{A_x}(x \mid y) + (1-\alpha)\, \ell_{A_y}(y \mid x)$.
$s_{\mathcal{M}}(x, y; x^{\mathrm{wt}})$	Mutation-local variant score (Eq. 3): $\ell_{\mathcal{M}}(x \mid y) - \ell_{\mathcal{M}}(x^{\mathrm{wt}} \mid y)$.
$\bar{\gamma}$	Symmetric contextualized likelihood margin between matched and sampled-mismatched contexts.
$\gamma_{x\mid y}, \gamma_{y\mid x}$	Directional likelihood margins under the negative-sampling distributions.
α	Learned directional weight in s_α.
τ	Temperature in candidate-choice, contrastive, and pairwise preference losses.
$D_{ij}^{y\to x}$	Cross-modal dependency map (Eq. 6) measuring how perturbing context token j shifts the predicted distribution at sequence position i.
Proof notation	
ℓ_i^V	Site log-likelihood $\log \pi_\theta(x_{V,i} \mid x_{V,\setminus i}, y)$ for $V \in \{A, B, \mathrm{wt}\}$.
d_i	Per-site log-likelihood gap $\ell_i^A - \ell_i^B$.
Δ	Exact score gap $\frac{1}{m}\sum_{i\in\mathcal{M}} d_i$ for variants with the same mutation set.
x_A, x_B	Two variants compared under the same context and mutation set.
$\mu_i, \bar{\mu}_{\mathcal{M}}$	Site-level mean advantage and its average over the mutation set.
$\sigma_i^2, \nu_{\mathcal{M}}^2$	Site-level sub-Gaussian parameters and the averaged parameter $\frac{1}{m^2}\sum_{i\in\mathcal{M}} \sigma_i^2$ used in the misranking bound.

I. Supplementary Figures

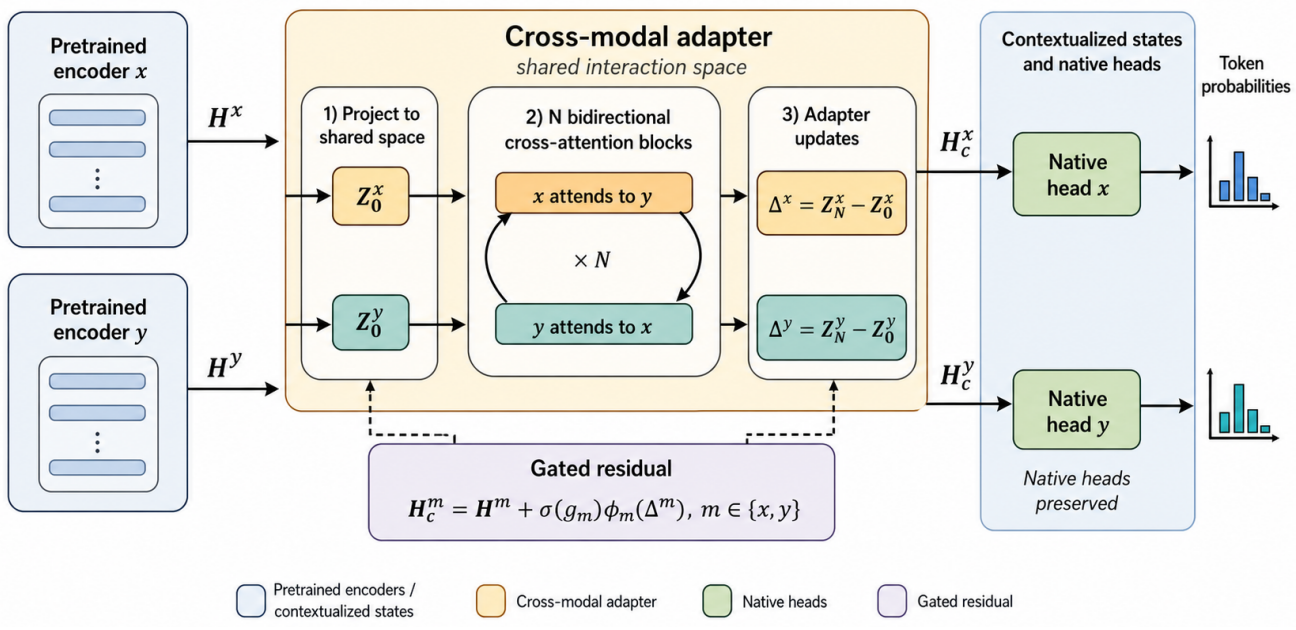


Figure S4. The LOGICA architecture with native head-preserving cross-modal adapters Pretrained hidden states from two biological foundation models (H^x, H^y) are projected into a shared interaction space. A stack of N bidirectional cross-attention blocks computes contextual updates. These updates ($Z_N - Z_0$) are then mapped back to the native dimensions via ϕ and integrated through a gated residual connection. This mechanism ensures the contextualized states (H_c^m) remain compatible with the original, native task heads, preserving the models' pretrained token probabilities and specialized functions.

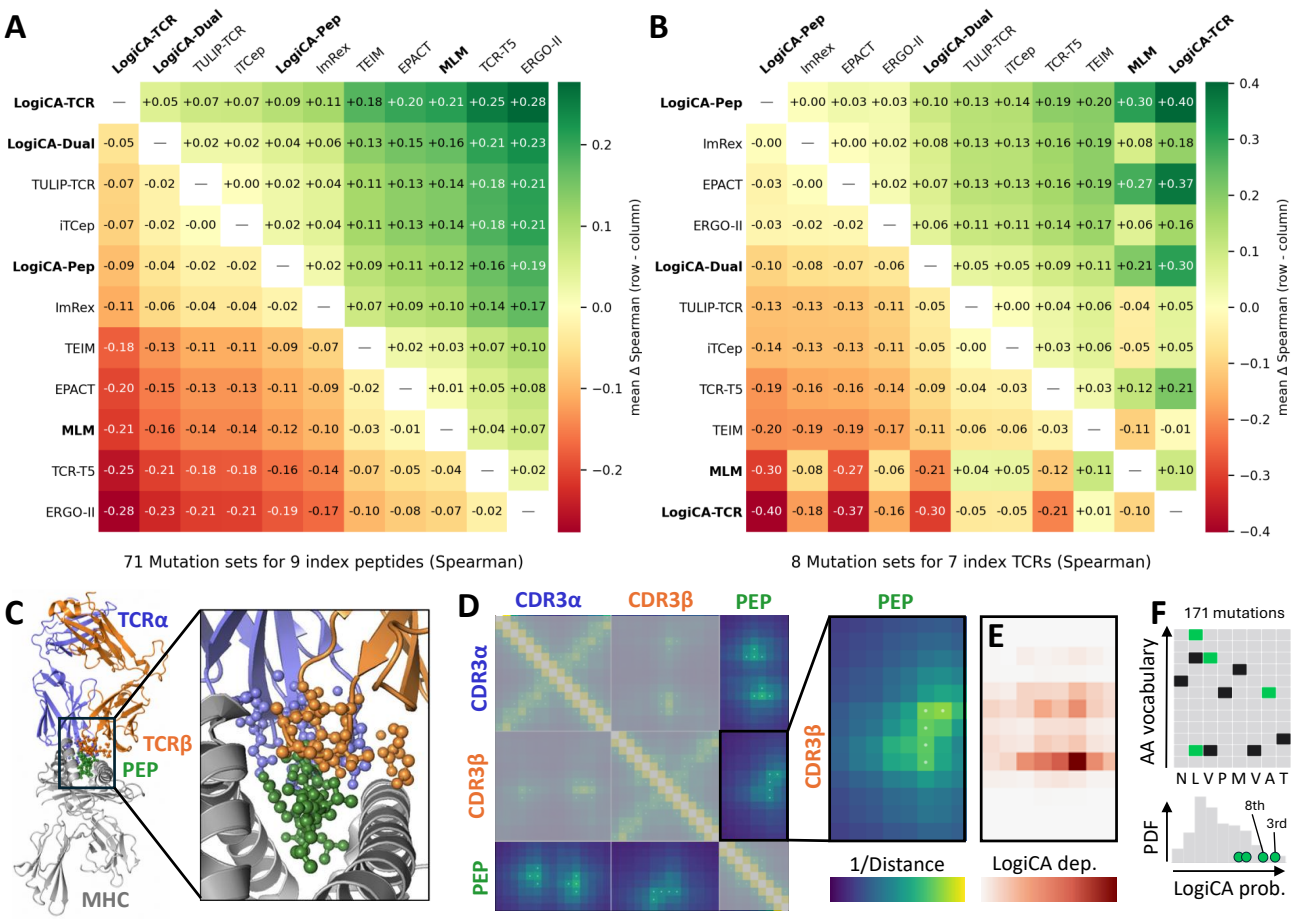


Figure S5. LogiCA performs zero-shot TCR–peptide variant ranking and identifies cross-modal dependencies. (A, B) Pairwise win-margin heatmap for peptide variant ranking and TCR variant ranking. Each cell reports the mean difference in Spearman correlation between the row model and the column model across mutation sets, with positive values indicating that the row model performs better. Individual scores are provided in Table 3 and Figure S6. (C) Representative TCR–pMHC complex structure (PDB 5TEZ), with peptide residues and TCR CDR regions shown as spheres to highlight the primary interaction interface. (D) Ground-truth residue-proximity map. Residue pairs within 5 Å are marked as contacts, while intra-modality pairs are grayed out because the analysis focuses on inter-molecular dependencies. (E) Zero-shot LogiCA-predicted dependency scores between CDR3β and the peptide. (F) Zero-shot prioritization of mutations for the NLVPMVATV peptide (Kula et al., 2019). Among 171 screened single mutants, four showed improved activity over the wild-type peptide. LogiCA ranks these activity-enhancing variants highly: L2I at 3/171, L2V at 8/171, A7P at 29/171, and V3L at 48/171.

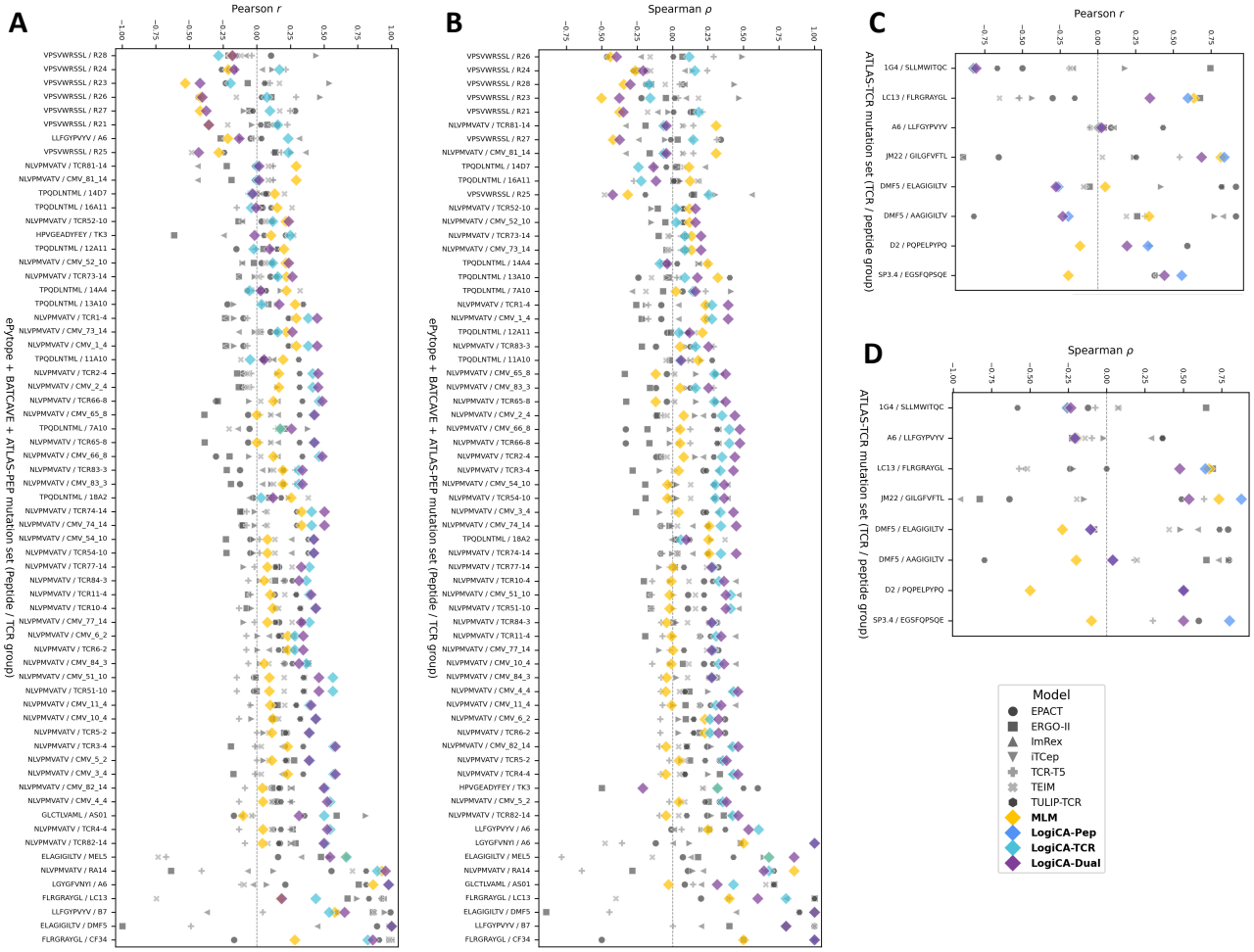


Figure S6. Per-mutation-set variant-ranking performance across TCR–epitope benchmarks. Each point represents one model's correlation between predicted variant scores and experimental readouts within a mutation set. Panels A and B show peptide-mutation sets from ePytope, BATCAVE, and ATLAS-PEP; panels C and D show TCR-mutation sets from ATLAS-TCR. Panels A–C report Pearson correlation, and panels B–D report Spearman correlation. The dashed horizontal line indicates zero correlation. In-family model variants are highlighted with diamond markers.