AbSpecAlign: Specificity Reward Alignment for Antigen-Conditioned Antibody Generation

Jingbo Zeng[1,2*] , **Yang Xiao**[1,2*] , **Shenghua Gao**[1,2†]
[1]The University of Hong Kong, [2]Shenzhen Loop Area Institute
zengj1ngb@connect.hku.hk

Abstract

Designing antibodies that recognize a target antigen while avoiding non-cognate antigens remains a central challenge in computational antibody engineering. Existing antigen-conditioned generative models can jointly design antibody sequences and structures, but they are typically trained to match structural data distributions and lack an explicit alignment mechanism for antigen-specific preferences. We present AbSpecAlign, a plug-and-play specificity-reward alignment framework for antibody design. AbSpecAlign introduces a Bidirectional Multimodal Specificity Scorer (BiMSS), which integrates antibody and antigen sequence representations with SE(3)-equivariant structural embeddings to produce a contrastive reward over cognate versus non-cognate antigen pairs. We further introduce a GRPO-based post-training strategy that uses group-relative specificity rewards to align pre-trained antibody generators toward higher-scoring antigen-conditioned designs while maintaining structural regularity. To the best of our knowledge, AbSpecAlign is among the first frameworks to apply GRPO-style group-relative reward alignment to antigen-conditioned antibody sequence–structure co-design with an explicit off-target specificity reward. Experiments on RAbD CDR design benchmarks show that AbSpecAlign improves most sequence-recovery and structural-fidelity metrics when combined with diffusion-based antibody generators. Experiment results suggest that GRPO-based reward alignment with contrastive specificity signals provides an effective post-training mechanism for antigen-conditioned antibody generation, while prospective experimental validation remains necessary.

1 Introduction

Antibodies are essential components of the adaptive immune system and have become indispensable in modern therapeutics and diagnostics due to their ability to bind diverse disease-associated antigens with high specificity [20]. Structurally, a typical antibody molecule consists of two identical heavy chains V_H and two light chains V_L arranged in a Y-shaped conformation, where the antigen-binding site is formed by the variable domains of both chains. Within these variable regions lie six complementarity determining regions (CDRs) — three from each chain — that primarily determine antigen specificity through their highly diverse sequences and loop conformations. Among all CDRs, CDR-H3 exhibits the greatest sequence and structural variability and therefore contributes disproportionately to antigen recognition diversity and binding affinity, making it a central focus in antibody design research. The enormous combinatorial space of possible CDR sequences and structures creates both opportunities and challenges for computational design methodologies [25].

Traditional antibody discovery methods, such as hybridoma technology [30, 32], animal immunization [8], and phage display [9, 15], have been instrumental in the generation of therapeutic antibodies.

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, these methods are time-consuming, labor-intensive, and heavily reliant on the diversity of antibody libraries, limiting their scalability and the speed at which new antibodies can be developed. Furthermore, these techniques often fail to address the growing demand for optimized antibody affinity and specificity for novel or difficult targets. These challenges highlight the need for more efficient computational approaches to antibody design.

Recent advances in learning-based methods have revolutionized the field of antibody design. Protein language models (PLMs) such as ESM-2 [17] and AntiBERT [23] provide sequence embeddings that capture complex patterns in antibody sequences, enabling better predictions of binding affinity and functional properties. Additionally, learning-based structural models, such as IgFold [24], have made significant strides in accurately predicting three-dimensional structures of antibodies from sequence alone, facilitating better designs by aligning sequence, structure, and function. These models not only improve prediction accuracy, but also open new avenues for generative antibody design. However, these works have focused more on the sequence level and have not achieved a structure-to-structure design between antigens and antibodies.

Recently, diffusion models have shown strong performance in jointly generating antibody sequences and structures conditioned on antigen context [19, 18, 13]. Although these works provide an outstanding ability to condition antibody generation on antigen structure, improving design outcomes, existing approaches still exhibit some key limitations. Most prior diffusion-based co-design methods primarily optimize affinity for an antigen and antibody pair, but do not explicitly estimate specificity between generated antibodies and other antigens with similar structure.

In this paper, we introduce AbSpecAlign for high-specificity antibody generation. Our approach conditions antibody generation on both antigen sequence and structure, enabling better control over antigen-specific binding. By augmenting generative models with an RL-driven optimization process, we actively steer the design process toward favorable interactions, optimizing for binding specificity. We summarize our contributions as follows.

1. AbSpecAlign serves as a plug-and-play reward-alignment framework for improving antigen-conditioned antibody generation.
2. We propose a bidirectional feature fusion module (BiMSS) for combining antigen-centric and antibody-centric embeddings, attached with a reward function for determining the binding specificity of a given antigen-antibody pair.
3. Using this reward signal, we formulate a GRPO-based [26] post-training strategy to optimize the functional properties of antibody–antigen interactions during generation.
4. We evaluate our model on the CDR sequence-structure co-design task. Experiments show that integrating AbSpecAlign with pretrained generators improves sequence recovery and structural fidelity across most CDR design settings.

2 Related Work

2.1 Learning-Based Antibody Design

Learning-based antibody design builds on two pillars: accurate structure prediction and large-scale sequence modeling. Previous methods for general protein folding have been outperformed by AlphaFold-2 [12], which achieves near-experimental accuracy on monomeric structures and extends to protein complexes through the more recent AlphaFold-3 [1]. These models and associated databases make it possible to obtain high-quality structural priors for both antigens and antibodies directly from sequence, which in turn enables large-scale *in silico* screening and conditioning for generative models. Antibody-specific structure predictors such as IgFold [24] further specialize these ideas for variable domains, combining antibody language models with equivariant GNNs to rapidly predict variable fragment structures and CDR loop conformations from sequence alone.

On the sequence side, a series of antibody-specific language models have been trained on the Observed Antibody Space (OAS) and related large repertoires. AbLang [21] focuses on sequence restoration and germline-aware mutation modeling, AntiBERTa [16] and related BCR models learn contextual embeddings that capture paratope positions and other functional signals, and more recent foundation models such as BALM [11] provide general-purpose antibody representations for downstream prediction and design. These models are increasingly coupled with generative architectures: IgLM

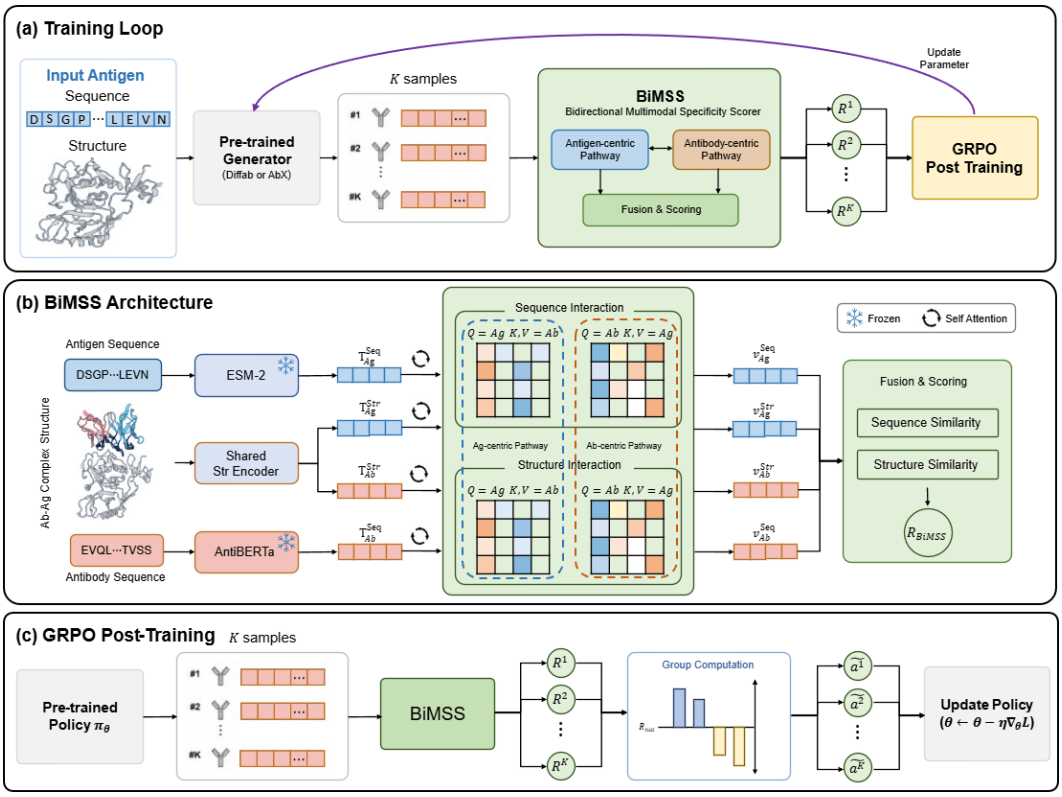


Figure 1: Overview of AbSpecAlign. (a) Architecture and workflow: AbSpecAlign leverages antigen sequence and structure as input, uses Bidirectional Multimodal Specificity Scorer (BiMSS) to estimate a specificity score for each antibody–antigen pair, and applies GRPO-based post-training to fine-tune a pre-trained antibody generator. (b) The BiMSS module fuses antigen-centric and antibody-centric embeddings and computes a score for the antigen-antibody complex. (c) GRPO-based post-training guides the model toward more specific generation.

[27] and PALM-H3 [10] use Transformer-style language models to generate antigen-conditioned CDR-H3 sequences, demonstrating that sequence-only models can already produce functional and antigen-specific loops.

Building on these foundations, diffusion-based and score-based generative models have been introduced for antibody sequence–structure co-design. DiffAb [18] first demonstrated SE(3)-equivariant diffusion on antibody–antigen complexes, jointly generating CDR structures and antigen-conditioned sequences. AbX [33], IgDiff [5], and many other methods extend this line by incorporating evolutionary constraints, multi-scale equivariant graph networks, and explicit complex-level objectives to improve structural fidelity and developability. More broadly, RFdiffusion [31] and other protein-level diffusion frameworks provide powerful priors for de novo binder design that can be adapted to antibodies.

2.2 Specificity-Aware Antibody Optimization

Beyond de novo generation, a separate line of work explicitly targets antigen specificity and binding affinity. Many methods combine generative models with supervised predictors that estimate the binding or developability of a sequence or structure. Examples include PALM-H3 [10], which pairs an antigen-conditioned CDR-H3 generator with a binder network trained to predict antibody–antigen binding affinity, and IgLM [27] presents an infilling language model that generates CDR-H3 variants and ranks them using structural similarity and docking-derived metrics.

Structure-based optimization pipelines further integrate docking or physics-inspired scores. AbDesign and AbDock [22] combine a diffusion-based design model with a paratope–epitope docking network

to screen and refine CDR designs, showing that learned docking scores can substantially enrich high-affinity binders relative to Rosetta-based baselines. More generally, deep docking models such as EquiBind [28] and DiffDock [4] predict ligand or peptide binding poses with SE(3)-equivariant GNNs, while HADDOCK [7] leverages ML-generated antibody structures and AlphaFold-predicted antigens to model antibody–antigen complexes with high precision. Trained scoring networks such as SAGE-Rank [29] have also been developed specifically for ranking antibody–antigen decoys, highlighting the importance of discriminative models for specificity.

3 Method

3.1 Overview

Figure 1 provides an overview of the AbSpecAlign model. We address antibody design by combining a diffusion-based pre-trained backbone model with a specificity scorer that is sensitive to geometry and a group-wise reinforcement learning (RL) objective.

First, we pretrain an SE(3)-equivariant structure encoder on antibody–antigen complexes from SAbDab dataset. The encoder operates in atomic coordinates using an EGNN-style network and produces residue-level embeddings for both antibody and antigen.

Second, we train a bidirectional multimodal specificity scorer that takes paired antibody–antigen complexes as input. The heavy and light chain sequences of an antibody are encoded with AntiBERTa, while the antigen sequence is encoded with ESM-2. In parallel, the pre-trained EGNN provides structural embeddings at the residue level. These sequence and structure features are fused with cross-attention and pooled to obtain a pairwise representation, which is assigned to a scalar specificity score $R_{\mathrm{BiMSS}} \in (0, 1)$. The scorer is trained using native and off-target antibody–antigen pairs.

Finally, we freeze the encoder and scorer and use the resulting scalar score as a fixed reward to post-train a pre-trained antibody generation model. For each antigen, we sample a group of candidate antibodies, evaluate their rewards with the fixed scorer, and compute group-wise normalized advantages in a GRPO-style objective. The reward is treated as a stop-gradient coefficient in a score-function update: the generator is updated through the diffusion trajectory log-probability, while the frozen scorer is not optimized during post-training. This distinction is important because the generator contains both discrete amino-acid choices and continuous coordinates. The RL loss, combined with a small structural regularizer in the CDR geometry, updates the diffusion generator parameters without requiring direct backpropagation through the discrete sequence samples.

3.2 Pre-train the Structure Encoder

EGNN-based atomic encoder Let $\{x_i \in \mathbb{R}^3, f_i\}_{i=1}^N$ denote atomic coordinates and initial atom-level features for an antibody–antigen complex. We construct a radius graph and encode the complex with an EGNN-style message passing network. The message function depends on invariant pairwise distances, while the coordinate update is expressed as a weighted sum of relative coordinate vectors; therefore, the encoder preserves SE(3)-equivariance and can model antibody–antigen interface geometry without depending on a fixed global frame. The detailed layer-wise update equations are given in Appendix A.1.

Contact-based pre-train objective The structure encoder is pre-trained with supervised antibody–antigen contact prediction. For each antibody residue and antigen residue, the contact label is determined by the minimum inter-atomic distance under a threshold d_c. A shallow contact head maps the paired residue embeddings to a contact probability, and the encoder is optimized with binary cross-entropy over sampled residue pairs. This stage provides residue-level structural embeddings for both antibody and antigen, which are reused by the BiMSS scorer and the post-training reward. The explicit contact objective is provided in Appendix A.1.

3.3 Bi-directional Multimodal Specificity Scorer

Given an antibody–antigen complex $(\mathrm{Ab}, \mathrm{Ag})$, our scorer produces a scalar score $R_{\mathrm{BiMSS}} \in (0, 1)$ that estimates how specifically the antibody recognizes the antigen. The scorer is multimodal: it leverages both sequence information from pre-trained protein language models and 3D structural

information from the EGNN encoder. The overall pipeline is: (i) encode antibody and antigen sequences, as well as their 3D structures, into four token sequences; (ii) refine each sequence independently with self-attention; (iii) perform cross-attention between antibody and antigen in both modalities to capture pairwise interactions; and (iv) predict a specificity score and train it contrastively with positive and negative pairs.

Sequence encoders For each complex, we first encode the antibody and antigen sequences.

- **Antibody encoder** The heavy-chain and light-chain sequences are tokenized and passed through AntiBERTa. Let $\mathbf{H}_{\mathrm{H}} \in \mathbb{R}^{L_{\mathrm{H}} \times d_s}$ and $\mathbf{H}_{\mathrm{L}} \in \mathbb{R}^{L_{\mathrm{L}} \times d_s}$ denote the final hidden embeddings of the heavy and light chains, respectively. Here, L_{H} and L_{L} are the length of heavy and light chains, and d_s is the hidden dimension of AntiBERTa. We concatenate the two chains along the sequence dimension and apply a linear projection to obtain antibody sequence tokens
$$\mathbf{T}_{\mathrm{ab}}^{\mathrm{seq}} = \mathrm{Proj}_{\mathrm{ab}}\big([\mathbf{H}_{\mathrm{H}}; \mathbf{H}_{\mathrm{L}}]\big) \in \mathbb{R}^{L_{\mathrm{ab}} \times d}, \tag{1}$$
where $L_{\mathrm{ab}} = L_{\mathrm{H}} + L_{\mathrm{L}}$.
- **Antigen encoder** The antigen protein sequence is encoded with ESM-2. Let $\mathbf{H}_{\mathrm{ag}} \in \mathbb{R}^{L_{\mathrm{ag}} \times d_s}$ be the final hidden states; we apply a linear projection to match the model dimension,
$$\mathbf{T}_{\mathrm{ag}}^{\mathrm{seq}} = \mathrm{Proj}_{\mathrm{ag}}(\mathbf{H}_{\mathrm{ag}}) \in \mathbb{R}^{L_{\mathrm{ag}} \times d}. \tag{2}$$

Structural encoders In parallel, we apply the EGNN encoder to the antibody–antigen complex, obtaining residue-level structural embeddings for both partners. Let $\mathbf{T}_{\mathrm{ab}}^{\mathrm{str}} \in \mathbb{R}^{R_{\mathrm{ab}} \times d}$ and $\mathbf{T}_{\mathrm{ag}}^{\mathrm{str}} \in \mathbb{R}^{R_{\mathrm{ag}} \times d}$ denote the resulting residue tokens for the antibody and antigen, respectively. We refer to
$$\mathbf{T}_{\mathrm{ab}}^{\mathrm{seq}},\ \mathbf{T}_{\mathrm{ag}}^{\mathrm{seq}},\ \mathbf{T}_{\mathrm{ab}}^{\mathrm{str}},\ \mathbf{T}_{\mathrm{ag}}^{\mathrm{str}}$$
collectively as the four input token sequences to the fusion process.

Per-stream self-attention Before modeling antibody–antigen interactions, we refine each token stream. This step provides contextualized representations within each sequence or structure stream, without mixing antibody and antigen information.

Cross-attention and multimodal fusion To explicitly model pairwise interactions, we then apply bi-directional cross-attention between antibody and antigen embeddings, obtaining two pairwise representations: an antigen-centric view and an antibody-centric view. For example, in the sequence modality, we let antibody tokens attend to antigen tokens via
$$\tilde{\mathbf{H}}_{\mathrm{ab}}^{\mathrm{seq}} = \mathrm{crossattn}\left(\mathbf{T}_{\mathrm{ab}}^{\mathrm{seq}},\ \mathbf{T}_{\mathrm{ag}}^{\mathrm{seq}}\right). \tag{3}$$
Similarly, by reversing the query–key/value roles in the sequence modality and applying the same bidirectional cross-attention to the structural tokens, we obtain the remaining three cross-contextualized token embeddings $\tilde{\mathbf{H}}_{\mathrm{ag}}^{\mathrm{seq}}$, $\tilde{\mathbf{H}}_{\mathrm{ab}}^{\mathrm{str}}$, and $\tilde{\mathbf{H}}_{\mathrm{ag}}^{\mathrm{str}}$. We then pool these cross-contextualized tokens to obtain four global vectors
$$\mathbf{v}_{\mathrm{ab}}^{\mathrm{seq}},\ \mathbf{v}_{\mathrm{ag}}^{\mathrm{seq}},\ \mathbf{v}_{\mathrm{ab}}^{\mathrm{str}},\ \mathbf{v}_{\mathrm{ag}}^{\mathrm{str}} \in \mathbb{R}^d. \tag{4}$$
We compute cosine similarities in each modality,
$$s_{\mathrm{seq}}(\mathrm{Ab}, \mathrm{Ag}) = \frac{\langle \mathbf{v}_{\mathrm{ab}}^{\mathrm{seq}}, \mathbf{v}_{\mathrm{ag}}^{\mathrm{seq}} \rangle}{\|\mathbf{v}_{\mathrm{ab}}^{\mathrm{seq}}\|_2 \|\mathbf{v}_{\mathrm{ag}}^{\mathrm{seq}}\|_2}, \tag{5}$$
$$s_{\mathrm{str}}(\mathrm{Ab}, \mathrm{Ag}) = \frac{\langle \mathbf{v}_{\mathrm{ab}}^{\mathrm{str}}, \mathbf{v}_{\mathrm{ag}}^{\mathrm{str}} \rangle}{\|\mathbf{v}_{\mathrm{ab}}^{\mathrm{str}}\|_2 \|\mathbf{v}_{\mathrm{ag}}^{\mathrm{str}}\|_2}, \tag{6}$$
and fuse them with a learnable convex combination,
$$s = \alpha\, s_{\mathrm{seq}} + (1 - \alpha)\, s_{\mathrm{str}}, \qquad \alpha \in [0, 1]. \tag{7}$$
Finally, a sharpness-controlled sigmoid maps s to a scalar specificity score,
$$R_{\mathrm{BiMSS}}(Ab, Ag) = \sigma(\beta\, [\alpha\, s_{\mathrm{seq}} + (1 - \alpha)\, s_{\mathrm{str}}]), \tag{8}$$
where $\beta > 0$ controls the sharpness and gradient magnitude. This score $R \in (0, 1)$ is a learned scalar reward used in post-training.

Contrastive training with positive and negative pairs To train the scorer, we construct mini-batches where each antibody Ab is paired with its cognate antigen Ag^+ (positive pair) and multiple non-cognate antigens $\{\mathrm{Ag}_n^-\}_{n=1}^N$ sampled from other SAbDab entries. For each modality $m \in \{\mathrm{seq}, \mathrm{str}\}$ we have

$$s_{\mathrm{m}}^+ = \frac{s_{\mathrm{m}}(\mathrm{Ab}, \mathrm{Ag}^+)}{\tau}, \tag{9}$$

$$s_{\mathrm{m,n}}^- = \frac{s_{\mathrm{m}}(\mathrm{Ab}, \mathrm{Ag}_{\mathrm{n}}^-)}{\tau}, \quad n = 1, \ldots, N, \tag{10}$$

where s_m denotes s_{seq} or s_{str}, τ is the temperature, and N denotes the number of non-cognate antigens. We then apply an InfoNCE-style contrastive loss for each modality:

$$\mathcal{L}_{\mathrm{m}} = -\log \frac{\exp(s_{\mathrm{m}}^+)}{\exp(s_{\mathrm{m}}^+) + \sum_{n=1}^N \exp(s_{\mathrm{m,n}}^-)}. \tag{11}$$

Despite optimizing sequence and structure separately, we also apply the same function for the combination s mentioned before.

$$\mathcal{L}_{\mathrm{com}} = -\log \frac{\exp s^+}{\exp(s^+) + \sum_{n=1}^N \exp(s_n^-)}, \tag{12}$$

where s^+ and s_n^- are defined similarly based on Equation 9 and 10. The overall scorer loss is

$$\mathcal{L}_{\mathrm{BiMSS}} = \mathcal{L}_{\mathrm{com}} + \delta_{\mathrm{seq}}\mathcal{L}_{\mathrm{seq}} + \delta_{\mathrm{str}}\mathcal{L}_{\mathrm{str}}, \tag{13}$$

where both δ_{seq} and δ_{str} are equal to 0.1.

After training, we freeze all parameters of the sequence encoders, EGNN encoder, and fusion layers, and only use the resulting scorer as a fixed reward function R_{BiMSS} in the post-training stage.

3.4 Post-training with GRPO

Antigen-conditioned generation We start from a pre-trained diffusion-based antibody design model, for example, DiffAb, viewed as a conditional policy π_θ that generates antibody sequence–structure pairs conditioned on a target antigen Ag. Let c denote the conditioning context constructed from the antigen. For each antigen, we draw K independent noise seeds and generate a group of candidates:

$$\hat{Ab}^{(k)} = \pi_\theta(c, \epsilon^{(k)}), \quad \epsilon^{(k)} \sim p(\epsilon), \quad k = 1, \ldots, K. \tag{14}$$

In our experiments, K is set to 5. Each generated antibody is paired with the fixed antigen and scored by the frozen specificity scorer:

$$R_{\mathrm{gen}}^{(k)} = R_{\mathrm{BiMSS}}(\hat{Ab}^{(k)}, Ag) \in (0, 1). \tag{15}$$

We additionally compute a per-antigen baseline using the native complex,

$$R_{\mathrm{nat}} = R_{\mathrm{BiMSS}}(Ab_{\mathrm{nat}}, Ag), \tag{16}$$

and define the raw advantage $a^{(k)} = R_{\mathrm{gen}}^{(k)} - R_{\mathrm{nat}}$.

Group-wise normalization To reduce variance and make updates comparable across antigens, we normalize advantages within each group:

$$\bar{a} = \frac{1}{K}\sum_{k=1}^K a^{(k)}, \tag{17}$$

$$s_a = \sqrt{\frac{1}{K}\sum_{k=1}^K \left(a^{(k)} - \bar{a}\right)^2}, \tag{18}$$

and form standardized advantage:

$$\tilde{a}^{(k)} = \begin{cases} 0, & s_a < \varepsilon, \\ \dfrac{a^{(k)} - \bar{a}}{s_a + \varepsilon}, & \text{otherwise}, \end{cases} \tag{19}$$

where $\varepsilon > 0$ is a small constant for numerical stability.

Table 1: Evaluation of de novo CDR design on the RAbD test set. The best results are shown in **bold**.

CDR	Method	AAR(%)↑	RMSD(Å) ↓	CDR	Method	AAR(%)↑	RMSD(Å) ↓
H1	DiffAb	70.01	0.88	L1	DiffAb	61.07	0.85
	DiffAb+AbSpecAlign	72.14	0.86		DiffAb+AbSpecAlign	63.28	0.83
	AbX	80.72	0.85		AbX	79.37	0.78
	AbX+AbSpecAlign	**81.93**	**0.85**		AbX+AbSpecAlign	**80.81**	**0.76**
H2	DiffAb	38.52	0.78	L2	DiffAb	58.58	0.55
	DiffAb+AbSpecAlign	41.27	0.74		DiffAb+AbSpecAlign	55.39	0.51
	AbX	70.73	0.76		AbX	84.53	0.45
	AbX+AbSpecAlign	**72.25**	**0.74**		AbX+AbSpecAlign	**85.97**	**0.44**
H3	DiffAb	28.05	2.86	L3	DiffAb	47.57	1.39
	DiffAb+AbSpecAlign	30.83	2.67		DiffAb+AbSpecAlign	50.02	1.22
	AbX	45.18	2.50		AbX	65.92	1.18
	AbX+AbSpecAlign	**46.09**	**2.48**		AbX+AbSpecAlign	**67.13**	**1.17**

GRPO objective for diffusion policies GRPO updates the generator by increasing the likelihood of higher-advantage samples and decreasing that of lower-advantage samples:

$$\mathcal{L}_{\mathrm{RL}}(\theta) = -\frac{1}{K}\sum_{k=1}^{K} \mathrm{sg}\Big(\tilde{a}^{(k)}\Big)\, \log \pi_\theta\Big(\hat{Ab}^{(k)} \mid c\Big), \tag{20}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. For diffusion generators, the policy log-probability is evaluated through the reverse diffusion trajectory; the corresponding factorization is shown in Appendix A.5.

To mitigate distribution drift and potential reward hacking, we keep a frozen reference model π_{ref} initialized from the pre-trained generator. The reference anchor penalizes deviation in trajectory likelihood:

$$\mathcal{L}_{\mathrm{ref}} = \frac{1}{K}\sum_{k=1}^{K}\Big(\log \pi_\theta(\hat{Ab}^{(k)} \mid c) - \log \pi_{\mathrm{ref}}(\hat{Ab}^{(k)} \mid c)\Big)^2. \tag{21}$$

We also use a weak structural regularizer $\mathcal{L}_{\mathrm{struct}}$ computed as the group-averaged RMSD between generated and native aligned CDR coordinates. The final objective is

$$\mathcal{L}_{\mathrm{post}} = \mathcal{L}_{\mathrm{RL}} + \mathcal{L}_{\mathrm{ref}} + \lambda_{\mathrm{struct}}\mathcal{L}_{\mathrm{struct}}, \tag{22}$$

where $\lambda_{\mathrm{struct}} \ll 1$ ensures the specificity-driven update remains dominant while the structural term provides a gentle constraint.

4 Experiments

4.1 Experimental Setup

We evaluate the performance of AbSpecAlign in the design of antibodies. In this section we will introduce the dataset, the baselines, and the evaluation metrics in our experiments. Section 4.2 presents the performance compared with other antibody generation models, and Section 4.3 illustrates the impact of the proposed method and post-training strategies. All models are trained on NVIDIA H200 GPUs.

Dataset We use the SAbDab dataset [6] to train and validate our model. We preprocess the dataset by removing samples with resolution worse than 4Å. Furthermore, samples targeting non-protein antigens are excluded. To ensure evaluation on truly unseen targets, we split the data at the antigen level using clustering rather than randomly splitting complexes. For evaluation, we follow the experimental protocols in DiffAb [18] on RAbD benchmark [2].

Baselines For evaluation, several baselines are implemented: DiffAb [18] is a diffusion-based generative model and AbX [33] is a score-based diffusion model guided by multiple constraints.

Table 2: Ablation studies for reward signal and post-training strategy.

Reward	Post-training	AAR(%)↑	TM-score↑	RMSD(Å)↓	DockQ↑
BiMSS	GRPO	**46.09**±1.28	**0.9917**	**2.48**±0.16	**0.461**±0.01
HADDOCK	GRPO	45.73±1.53	0.9904	2.59±0.24	0.447±0.02
BiMSS	PPO	45.15±1.46	0.9911	2.63±0.15	0.442±0.03
-	SFT	44.37±1.60	0.9907	2.72±0.16	0.435±0.02
-	-	43.09±1.41	0.9906	2.88±0.21	0.429±0.04

Evaluation metrics Our model is evaluated by the following metrics: 1) Amino Acid Recovery (AAR), an overlapping ratio compared between generated sequence and ground truth; 2) Root Mean Square Deviation (RMSD), a deviation between the generated structure and ground truth; 3) TM-Score, a global similarity between the generated structure and ground truth; 4) DockQ [3], a comprehensive score for the docking quality. For Dockq computation, the antigen, antibody framework regions, and non-designed CDRs are kept fixed in the original complex coordinate frame. The residues in the masked CDR regions are replaced by the generated structure.

4.2 Sequence and Structure Co-design

Table 1 evaluates AbSpecAlign across all six CDRs under two antibody generation backbones. Overall, adding AbSpecAlign improves most sequence-recovery entries and consistently improves or preserves structural accuracy, indicating that the learned reward can provide useful interface-aware guidance beyond the original generative objectives.

For the DiffAb backbone, AbSpecAlign improves AAR on five out of six CDRs. The only exception is L2, where AAR decreases from 58.58% to 55.39%. Nevertheless, the structural metric still improves on L2, with RMSD decreasing from 0.55Å to 0.51Å.

For the stronger AbX backbone, the improvements are smaller but more consistent. AbSpecAlign improves AAR on all six CDRs: H1 improves from 80.72% to 81.93%, H2 from 70.73% to 72.25%, H3 from 45.18% to 46.09%, L1 from 79.37% to 80.81%, L2 from 84.53% to 85.97%, and L3 from 65.92% to 67.13%. RMSD is unchanged on H1 and decreases on the other five CDRs, including H3 from 2.50Å to 2.48Å and L3 from 1.18Å to 1.17Å. Averaged over all CDRs, AbSpecAlign improves AbX from 71.08% to 72.36% AAR and reduces RMSD from 1.09Å to 1.07Å.

These CDR-wise results support the interpretation that AbSpecAlign functions as a plug-in specificity refinement module. It provides larger gains when attached to the weaker DiffAb backbone and still yields consistent improvements when attached to the stronger AbX backbone. The improvements on H3 are especially relevant because CDR-H3 is the most variable and structurally challenging loop, while the gains on light-chain CDRs indicate that the learned reward is not restricted to a single redesign region.

4.3 Ablation Study

We conduct ablation studies to isolate the effects of the reward signal and the post-training strategy under the same generator, data split, sampling protocol, and update budget. As shown in Table 2, the full BiMSS+GRPO variant achieves the best performance among the compared settings, with $46.09 \pm 1.28\%$ AAR, 0.9917 TM-score, 2.48 ± 0.16Å RMSD, and 0.461 ± 0.01 DockQ. Compared with the generator without post-training, BiMSS+GRPO improves AAR from $43.09 \pm 1.41\%$ to $46.09 \pm 1.28\%$, reduces RMSD from 2.88 ± 0.21Å to 2.48 ± 0.16Å, and increases DockQ from 0.429 ± 0.04 to 0.461 ± 0.01.

Replacing the learned BiMSS reward with HADDOCK under the same GRPO procedure decreases AAR, increases RMSD, and reduces DockQ. This indicates that the learned antigen-aware specificity reward provides a more suitable optimization signal for antibody generation than a generic docking score. Replacing GRPO with PPO also lowers performance. The SFT variant performs worse than both RL-based variants, with $44.37 \pm 1.60\%$ AAR and 2.72 ± 0.16Å RMSD. These results suggest that group-wise reward normalization is important for stable candidate-level optimization and that

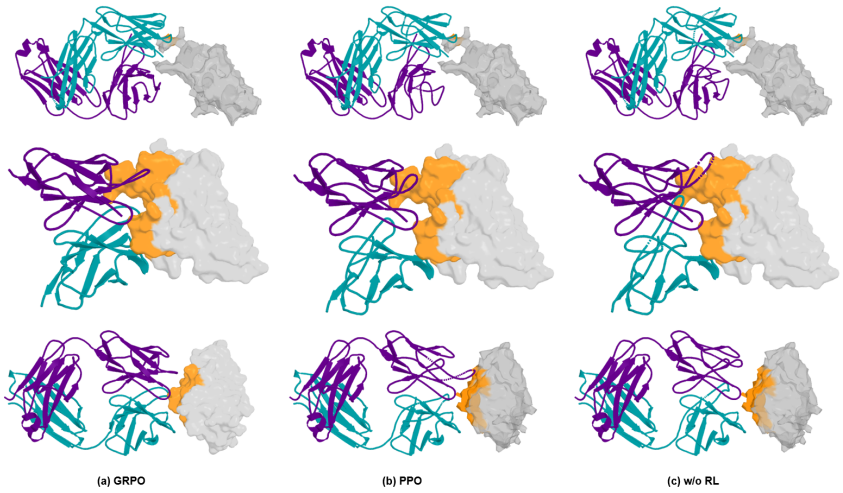


Figure 2: Visualization results for the usage of different post-training strategies.

reward-driven post-training is more effective than supervised fine-tuning alone. To more intuitively illustrate the effect of GRPO-based post-training, we additionally provide qualitative visualization results in Figure 2.

Figure 3 further ablates two key GRPO post-training hyperparameters. For the candidate group size K, the best setting among the tested values is $K = 5$. A smaller group size, $K = 3$, gives lower AAR and higher RMSD, suggesting that too few candidates provide noisier relative rewards. Increasing the group size to $K = 7$ does not further improve performance, which suggests that larger candidate groups do not necessarily provide a proportional benefit under the fixed update budget.

For the structural regularization coefficient, $\lambda_{\text{struct}} = 0.01$ gives the strongest result among the tested settings. By increasing the coefficient from 0.01 to 1, the performance of the model is decreased. This trend indicates that an overly large structural coefficient over-anchors the generator to coordinate reconstruction, whereas a small coefficient preserves the structural prior while allowing the learned specificity reward to dominate the alignment signal.

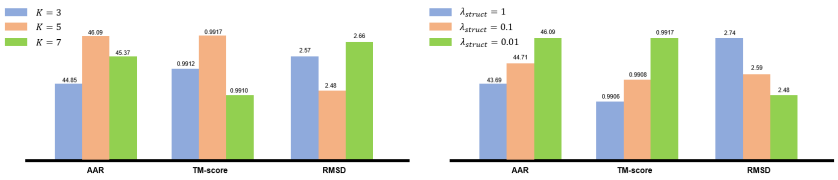


Figure 3: Parameter ablations for GRPO post-training. Left: varying the candidate count K. Right: varying the structural regularization coefficient λ_{struct}.

5 Conclusion

In this work, we introduced AbSpecAlign, a plug-and-play discriminative reward model for antigen-specific antibody design that complements existing sequence–structure co-design generators. AbSpecAlign integrates protein language encoders with an SE(3)-equivariant structural encoder and a Bidirectional Multimodal Specificity Scorer (BiMSS) module to score antibody–antigen pairs with a scalar specificity reward, enabling both lightweight post hoc refinement and reward-driven alignment of a pre-trained generator. Our experiments demonstrate that attaching AbSpecAlign yields consistent improvements in sequence recovery, structural fidelity, and interface quality. While our model demonstrates promising results, it is only evaluated by *in silico* experiments. Moreover, prospective wet-lab validation remains an important direction for future work.

Acknowledgement

This work is partially supported by funding and computing resource provided by HKU-Shanghai Intelligent Computing Research Center. This work was supported by the General Research Fund of the Research Grants Council of Hong Kong under Grant No. 17200725 and the JC STEM Lab funded by The Hong Kong Jockey Club Charities Trust.

References

[1] Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A.J., Bambrick, J., et al. (2024) Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature* **630**(8016):493–500.

[2] Adolf-Bryfogle, J., Kalyuzhniy, O., Kubitz, M., Weitzner, B.D., Hu, X., Adachi, Y., Schief, W.R., and Dunbrack Jr, R.L. (2018) Rosettaantibodydesign (rabd): A general framework for computational antibody design. *PLoS computational biology* **14**(4):e1006112.

[3] Basu, S. and Wallner, B. (2016) Dockq: a quality measure for protein–protein docking models. *PloS one* **11**(8):e0161879.

[4] Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T. (2022) Diffdock: Diffusion steps, twists, and turns for molecular docking. arXiv preprint arXiv:2210.01776.

[5] Cutting, D., Dreyer, F.A., Errington, D., Schneider, C., and Deane, C.M. (2025) De novo antibody design with se (3) diffusion. *Journal of Computational Biology* **32**(4):351–361.

[6] Dunbar, J., Krawczyk, K., Leem, J., Baker, T., Fuchs, A., Georges, G., Shi, J., and Deane, C.M. (2014) Sabdab: the structural antibody database. *Nucleic acids research* **42**(D1):D1140–D1146.

[7] Giulini, M., Schneider, C., Cutting, D., Desai, N., Deane, C.M., and Bonvin, A.M. (2024) Towards the accurate modelling of antigen-antibody complexes from sequence using machine learning and information-driven docking. *Bioinformatics* **40**(10):btae583.

[8] Gray, A.C., Bradbury, A.R., Knappik, A., Plückthun, A., Borrebaeck, C.A., and Dübel, S. (2020) Animal-derived-antibody generation faces strict reform in accordance with european union policy on animal use. *Nature methods* **17**(8):755–756.

[9] Hammers, C.M. and Stanley, J.R. (2014) Antibody phage display: technique and applications. *Journal of Investigative Dermatology* **134**(2):1–5.

[10] He, H., He, B., Guan, L., Zhao, Y., Jiang, F., Chen, G., Zhu, Q., Chen, C.Y.-C., Li, T., and Yao, J. (2024) De novo generation of sars-cov-2 antibody cdrh3 with a pre-trained generative large language model. *Nature Communications* **15**(1):6867.

[11] Jing, H., Gao, Z., Xu, S., Shen, T., Peng, Z., He, S., You, T., Ye, S., Lin, W., and Sun, S. (2024) Accurate prediction of antibody function and structure using bio-inspired antibody language model. *Briefings in Bioinformatics* **25**(4).

[12] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021) Highly accurate protein structure prediction with alphafold. *nature* **596**(7873):583–589.

[13] Kulytė, P., Vargas, F., Mathis, S.V., Wang, Y.G., Hernández-Lobato, J.M., and Liò, P. (2024) Improving antibody design with force-guided sampling in diffusion models. arXiv preprint arXiv:2406.05832.

[14] Langley, P. (2000) Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA. Morgan Kaufmann.

[15] Ledsgaard, L., Kilstrup, M., Karatt-Vellatt, A., McCafferty, J., and Laustsen, A.H. (2018) Basics of antibody phage display technology. *Toxins* **10**(6):236.

[16] Leem, J., Mitchell, L.S., Farmery, J.H., Barton, J., and Galson, J.D. (2022) Deciphering the language of antibodies using self-supervised learning. *Patterns* **3**(7).

[17] Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. (2022) Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv* 2022:500902.

[18] Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., and Ma, J. (2022) Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems* **35**:9754–9767.

[19] Martinkus, K., Ludwiczak, J., Cho, K., Liang, W., Lafrance-Vanasse, J., Hotzel, I., Rajpal, A., Wu, Y., Bonneau, R., Gligorijevic, V., et al. (2024) Abdiffuser: full-atom generation of in vitro functioning antibodies.

[20] Murphy, K., Travers, P., Walport, M., and Janeway, C. (2012) *Immunobiology*. Garland Science New York, NY.

[21] Olsen, T.H., Moal, I.H., and Deane, C.M. (2022) Ablang: an antibody language model for completing antibody sequences. *Bioinformatics Advances* **2**(1):vbac046.

[22] Peng, Z., Han, C., Wang, X., Li, D., and Yuan, F. (2023) Generative diffusion models for antibody design, docking, and optimization. *bioRxiv*, pp. 2023–09.

[23] Ruffolo, J.A., Gray, J.J., and Sulam, J. (2021) Deciphering antibody affinity maturation with language models and weakly supervised learning. arXiv preprint arXiv:2112.07782.

[24] Ruffolo, J.A., Chu, L.-S., Mahajan, S.P., and Gray, J.J. (2023) Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Nature communications* **14**(1):2389.

[25] Sela-Culang, I., Kunik, V., and Ofran, Y. (2013) The structural basis of antigen-antibody recognition. *Frontiers in Immunology*, Volume 4 - 2013. ISSN 1664-3224.

[26] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. (2024) Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300.

[27] Shuai, R.W., Ruffolo, J.A., and Gray, J.J. (2023) Iglm: Infilling language modeling for antibody sequence design. *Cell Systems* **14**(11):979–989.

[28] Stärk, H., Ganea, O., Pattanaik, L., Barzilay, R., and Jaakkola, T. (2022) Equibind: Geometric deep learning for drug binding structure prediction. In *International conference on machine learning*, pp. 20503–20521. PMLR.

[29] Sun, C., Li, X., Xu, H., Tang, Y., Bai, G., Wang, Y., and Ma, B. (2025) Sagerank: inductive learning of protein–protein interaction from antibody–antigen recognition. *Chemical Science* **16**(38):17885–17899.

[30] Tomita, M. and Tsumoto, K. (2011) Hybridoma technologies for antibody production. *Immunotherapy* **3**(3):371–380.

[31] Watson, J.L., Juergens, D., Bennett, N.R., Trippe, B.L., Yim, J., Eisenach, H.E., Ahern, W., Borst, A.J., Ragotte, R.J., Milles, L.F., et al. (2023) De novo design of protein structure and function with rfdiffusion. *Nature* **620**(7976):1089–1100.

[32] Zhang, C. (2012) Hybridoma technology for the generation of monoclonal antibodies. In *Antibody methods and protocols*, pp. 117–135. Springer.

[33] Zhu, T., Ren, M., and Zhang, H. (2024) Antibody design using a score-based diffusion model guided by evolutionary, physical and geometric constraints. In *Forty-first International Conference on Machine Learning*.

[34] Satorras, V. G., Hoogeboom, E., and Welling, M. (2021). E (n) equivariant graph neural networks. In *International conference on machine learning*. pp. 9323-9332. PMLR.

A Formula Derivations and Objective Details

A.1 EGNN structure encoder pre-train

The structure encoder is designed based on the EGNN network [34]. Let $\{x_i \in \mathbb{R}^3, f_i\}_{i=1}^N$ denote the atom-level input graph. We connect atoms within a radius threshold r and apply L EGNN layers. At layer ℓ, the hidden features and coordinates are updated by

$$d_{ij}^2 = \|x_i^\ell - x_j^\ell\|_2^2, \tag{23}$$

$$e_{ij}^\ell = \phi_e^\ell\big(h_i^\ell, h_j^\ell, d_{ij}^2\big)\,, \tag{24}$$

$$m_i^\ell = \sum_{j \in \mathcal{N}(i)} e_{ij}^\ell, \tag{25}$$

$$h_i^{\ell+1} = h_i^\ell + \phi_h^\ell(m_i^\ell), \tag{26}$$

$$x_i^{\ell+1} = x_i^\ell + \frac{1}{C} \sum_{j \in \mathcal{N}(i)} (x_i^\ell - x_j^\ell)\phi_x^\ell(e_{ij}^\ell). \tag{27}$$

Because the scalar messages depend on invariant distances and the coordinate displacement is a linear combination of relative vectors, the feature stream is SE(3)-invariant and the coordinate stream is SE(3)-equivariant.

For contact pretraining, an antibody residue r and antigen residue s are labeled as an interface contact if

$$y_{rs} = \mathbb{I}\left[\min_{a \in A(r),\, b \in A(s)} \|x_a - x_b\|_2 \le d_c\right]. \tag{28}$$

A contact head predicts $p_{rs} = \sigma(\phi_c([z_r; z_s]))$, where z_r and z_s are residue-level embeddings obtained by pooling atom embeddings within each residue. The pretraining loss is

$$\mathcal{L}_{\mathrm{EGNN}} = -\frac{1}{|\mathcal{P}|} \sum_{(r,s) \in \mathcal{P}} \left[y_{rs} \log p_{rs} + (1 - y_{rs}) \log(1 - p_{rs})\right]. \tag{29}$$

This objective encourages the structural encoder to represent antibody–antigen interface geometry before it is used by BiMSS.

A.2 Dimension bookkeeping for AbSpecAlign

Table 3 summarizes the tensor shapes used by AbSpecAlign. L_{H}, L_{L}, and L_{ag} denote the heavy-chain, light-chain, and antigen sequence lengths. R_{ab} and R_{ag} denote residue counts after structure preprocessing. All streams are projected to the same fusion width d.

Table 3: Input and output dimensions of the main AbSpecAlign components.

Component	Input	Output
AntiBERTa heavy-chain encoder	tokens in $\mathbb{N}^{L_{\mathrm{H}}}$	$\mathbf{H}_{\mathrm{H}} \in \mathbb{R}^{L_{\mathrm{H}} \times d_s}$
AntiBERTa light-chain encoder	tokens in $\mathbb{N}^{L_{\mathrm{L}}}$	$\mathbf{H}_{\mathrm{L}} \in \mathbb{R}^{L_{\mathrm{L}} \times d_s}$
Antibody sequence projection	$[\mathbf{H}_{\mathrm{H}}; \mathbf{H}_{\mathrm{L}}]$	$\mathbf{T}_{\mathrm{ab}}^{\mathrm{seq}} \in \mathbb{R}^{L_{\mathrm{ab}} \times d}$
ESM-2 antigen encoder	tokens in $\mathbb{N}^{L_{\mathrm{ag}}}$	$\mathbf{H}_{\mathrm{ag}} \in \mathbb{R}^{L_{\mathrm{ag}} \times d_s}$
Antigen sequence projection	$\mathbf{H}_{\mathrm{ag}}$	$\mathbf{T}_{\mathrm{ag}}^{\mathrm{seq}} \in \mathbb{R}^{L_{\mathrm{ag}} \times d}$
EGNN structural encoder	atom graph $(\mathbf{x}, \mathbf{f}, \mathcal{E})$	residue tokens $\mathbf{T}_{\mathrm{ab}}^{\mathrm{str}}$ and $\mathbf{T}_{\mathrm{ag}}^{\mathrm{str}}$
Self-attention block	one token stream in $\mathbb{R}^{n \times d}$	contextualized stream in $\mathbb{R}^{n \times d}$
Cross-attention block	query $\mathbb{R}^{n_q \times d}$; key/value $\mathbb{R}^{n_k \times d}$	cross-contextualized query in $\mathbb{R}^{n_q \times d}$
Pooling and scorer	token streams in $\mathbb{R}^{n \times d}$	scalar $R_{\mathrm{BiMSS}} \in (0, 1)$

A.3 Reward mapping and gradients

The fused similarity is

$$s = \alpha s_{\mathrm{seq}} + (1 - \alpha) s_{\mathrm{str}}, \qquad \alpha \in [0, 1], \tag{30}$$

and the specificity reward is parameterized as

$$R_{\mathrm{BiMSS}}(\mathrm{Ab}, \mathrm{Ag}) = \sigma(\beta s), \qquad \beta > 0. \tag{31}$$

The derivative with respect to the fused similarity is

$$\frac{\partial R_{\mathrm{BiMSS}}}{\partial s} = \beta R_{\mathrm{BiMSS}}(1 - R_{\mathrm{BiMSS}}), \tag{32}$$

which shows that β controls both reward sharpness and gradient magnitude. For the modality weight,

$$\frac{\partial R_{\mathrm{BiMSS}}}{\partial \alpha} = \beta R_{\mathrm{BiMSS}}(1 - R_{\mathrm{BiMSS}})(s_{\mathrm{seq}} - s_{\mathrm{str}}). \tag{33}$$

Thus, when sequence and structure similarities disagree, the fusion coefficient receives a learning signal that adapts the relative contribution of the two modalities.

A.4 InfoNCE objective used for positive and off-target pairs

For each antibody, the scorer receives one cognate antigen and N non-cognate antigens. Let q denote the antibody representation and let p^+ and p_k^- denote the positive and negative antigen representations in a given modality. Using cosine similarity $\mathrm{sim}(\cdot, \cdot)$ and temperature τ, the modality-level loss can be written as

$$\mathcal{L}_m = -\log \frac{\exp(\mathrm{sim}(q, p^+)/\tau)}{\exp(\mathrm{sim}(q, p^+)/\tau) + \sum_{n=1}^{N} \exp(\mathrm{sim}(q, p_n^-)/\tau)}. \tag{34}$$

Differentiating with respect to the positive logit ℓ^+ gives

$$\frac{\partial \mathcal{L}_m}{\partial \ell^+} = P^+ - 1, \qquad P^+ = \frac{\exp(\ell^+)}{\exp(\ell^+) + \sum_{n=1}^{N} \exp(\ell_n^-)}, \tag{35}$$

whereas each negative logit has derivative $\partial \mathcal{L}_m / \partial \ell_n^- = P_n^-$. The objective therefore increases the cognate antibody–antigen similarity while reducing similarities to off-target antigens.

A.5 Diffusion-policy log probability and structural regularization

When the generator is a diffusion model, the policy probability of a generated antibody is represented by the reverse trajectory. For a trajectory $\mathbf{z}_{0:T}$ with $\mathbf{z}_0 = \widehat{\mathrm{Ab}}$, the conditional probability factorizes as

$$p_\theta(\mathbf{z}_{0:T} \mid c) = p(\mathbf{z}_T) \prod_{t=1}^{T} p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t, c). \tag{36}$$

Since $p(\mathbf{z}_T)$ is independent of θ, the reward-weighted policy update uses

$$\log \pi_\theta(\widehat{\mathrm{Ab}} \mid c) \simeq \sum_{t=1}^{T} \log p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t, c). \tag{37}$$

In sequence–structure co-design, $\mathbf{z}_t$ contains both continuous coordinate variables and residue-identity variables. The coordinate contribution is the log-density of the Gaussian reverse transition, while the sequence contribution is the categorical or discretized denoising likelihood used by the backbone generator. We sum both contributions when available in the backbone implementation. The standardized group advantage is treated as a stop-gradient coefficient, so the update changes the generator rather than the frozen reward model. Consequently, BiMSS provides a sample-level ranking reward; it is not used as a direct differentiable loss through amino-acid identities.

For the structural regularizer, let $X_{\mathrm{nat}} \in \mathbb{R}^{N \times 3}$ and $X_{\mathrm{gen}}^{(k)} \in \mathbb{R}^{N \times 3}$ denote aligned coordinates of the selected CDR residues. We define

$$d^{(k)} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \| X_{\mathrm{gen},i}^{(k)} - X_{\mathrm{nat},i} \|_2^2}, \qquad \mathcal{L}_{\mathrm{struct}} = \frac{1}{K} \sum_{k=1}^{K} d^{(k)}. \tag{38}$$

This term constrains the post-trained generator to remain close to the pre-trained structural prior while the BiMSS reward improves antigen-conditioned specificity.

Algorithm 1 GRPO post-training

Require: Generator π_θ, frozen reference π_{ref}, frozen scorer R_{BiMSS}, dataset $\mathcal{D} = \{(Ag, Ab_{\mathrm{nat}})\}$, group size K, ε, $\lambda_{\mathrm{struct}}$, lr η

for each iteration **do**

  Sample minibatch $\mathcal{B} \subset \mathcal{D}$

  $\mathcal{L} \leftarrow 0$

  for all $(Ag, Ab_{\mathrm{nat}}) \in \mathcal{B}$ **do**

    $c \leftarrow \mathrm{BuildContext}(Ag)$; $\quad R_{\mathrm{nat}} \leftarrow R_{\mathrm{BiMSS}}(Ab_{\mathrm{nat}}, Ag)$

    for $k = 1$ to K **do**

      $(\hat{Ab}^{(k)}, \log p_\theta^{(k)}) \leftarrow \mathrm{LogProb}(\pi_\theta, c)$

      $\log p_{\mathrm{ref}}^{(k)} \leftarrow \mathrm{LogProbRef}(\pi_{\mathrm{ref}}, \hat{Ab}^{(k)}, c)$

      $a^{(k)} \leftarrow R_{\mathrm{BiMSS}}(\hat{Ab}^{(k)}, Ag) - R_{\mathrm{nat}}$

      $d^{(k)} \leftarrow \mathrm{RMSD}(\hat{Ab}^{(k)}, Ab_{\mathrm{nat}})$

    end for

    $\tilde{\mathbf{a}} \leftarrow (\mathbf{a}, \varepsilon)\ \{(a - \bar{a})/(s_a + \varepsilon)\}$

    $\mathcal{L} \leftarrow -\frac{1}{K} \sum_k sg(\tilde{a}^{(k)})\ \log p_\theta^{(k)} + \frac{1}{K} \sum_k \left(\log p_\theta^{(k)} - \log p_{\mathrm{ref}}^{(k)} \right)^2 + \lambda_{\mathrm{struct}} \frac{1}{K} \sum_k d^{(k)}$

  end for

  $\theta \leftarrow \theta - \eta\, \nabla_\theta \mathcal{L}$

end for

A.6 Details of GRPO Post-Training

Details of the overall sampling algorithm are provided in Algorithm 1.

B Experimental Implementation Details

B.1 Complex preprocessing

SAbDab complexes are filtered by structural resolution, retaining protein-antigen complexes with resolution better than 4 Å. Non-protein antigens are removed to keep the antigen encoder and the structural interface definition consistent with protein sequence–structure modeling. Antibody and antigen chains are parsed into sequence tokens for the language-model encoders and atomic graphs for EGNN encoding.

B.2 Negative-pair construction

For BiMSS training, each native antibody–antigen pair is treated as a positive example. Negative examples are constructed by pairing the same antibody with non-cognate antigens sampled from other SAbDab entries. We use $K = 5$ off-target antigens per positive pair in the scorer loss. This design matches the intended specificity criterion: a high reward should indicate plausible binding to the cognate antigen and discriminate against unrelated or potentially confusable antigens.

B.3 Post-training protocol

For each antigen conditioning context, the generator samples K antibodies by drawing independent diffusion noise seeds. The frozen BiMSS module scores each generated antibody–antigen pair, and the native complex score is used as a per-antigen reference baseline. Group-wise normalization is then performed within candidates generated for the same antigen, avoiding the need to learn an additional value function. The reference model π_{ref} is initialized from the pre-trained generator and remains frozen. The structural regularizer is evaluated on aligned CDR coordinates, with $\lambda_{\mathrm{struct}} = 0.01$ in the reported configuration.

B.4 Statistical reporting

For the main CDR-H3 and ablation tables, we report error bars estimation. Because full retraining of all generator backbones is computationally expensive, this estimation is constructed by antigen-level bootstrap resampling of the RAbD test set and repeated stochastic generation seeds. Unless otherwise stated, the reported error bars are bootstrap standard deviations of the mean over 1,000 resamples, rounded to two decimal places. This protocol captures variability from antigen selection and sampling stochasticity while keeping the backbone training budget fixed.

B.5 Reproducibility

All result tables use the same antigen-level split, candidate-generation budget, and post-training update budget for the corresponding backbone. To make the uncertainty estimates reproducible without requiring independent retraining of every backbone, we first generate candidates with fixed model checkpoints and multiple diffusion seeds, then compute antigen-level metrics, and finally bootstrap the test antigens with replacement for 1,000 resamples. The standard deviation of the bootstrap mean is reported as the error bar. This procedure is applied consistently to Table **??** and Table 2; CDR-wise results in Table 1 are kept as point estimates for compactness.

Table 4: Reproducibility-relevant experimental settings used in the reported configuration.

Item	Setting
Dataset source	SAbDab antibody–antigen complexes; protein-antigen complexes only
Structure filter	Retain complexes with crystallographic resolution better than or equal to 4.0 Å; remove non-protein antigens
Split protocol	Antigen-level clustering split following the RAbD/DiffAb protocol; no antigen cluster is shared between train and test
Designed region	Main task: CDR-H3 sequence–structure co-design; additional CDR-wise evaluation follows the same preprocessing
Scorer negatives	$N = 5$ non-cognate antigens per cognate antibody–antigen pair during BiMSS contrastive training
Generator group size	$K = 5$ candidates per antigen for GRPO post-training unless varied in the parameter ablation
Structural regularization	Aligned CDR RMSD regularizer with $\lambda_{\mathrm{struct}} = 0.01$ in the default configuration
Compute	NVIDIA H200 GPUs; the same sampling budget, split, and update budget are used for all reward and optimizer ablations

C Additional Ablation Discussion

The reward ablation tests whether an external docking score can replace BiMSS. We keep the post-training algorithm fixed as GRPO and replace the learned reward with the HADDOCK docking score. This comparison isolates the reward source. HADDOCK can evaluate geometric compatibility and complex stability, but it is not trained to distinguish a cognate antigen from non-cognate antigens sampled from the antibody–antigen complex distribution. BiMSS is therefore better aligned with the specificity objective used in this paper.

The optimizer ablation keeps the learned BiMSS reward fixed and compares GRPO with PPO and SFT. PPO introduces additional sensitivity to reward scale and policy-ratio clipping, whereas GRPO uses group-wise normalization among candidates generated for the same antigen. This normalization is suitable for antigen-conditioned generation because the absolute reward scale may vary across antigens. SFT improves imitation of the training distribution but does not directly optimize the candidate-level specificity reward; hence it is weaker than reward-driven post-training for interface refinement.

We further evaluate the sensitivity of AbSpecAlign to two reward-related hyperparameters: the InfoNCE temperature τ and the BiMSS sharpness coefficient β in Eq. 8. The temperature τ controls the softness of the contrastive distribution used to train the specificity scorer. The results are presented in Figure 4. We evaluate $\tau \in \{0.01, 0.05, 0.1, 0.5, 1\}$ and observe the best performance at $\tau = 0.1$, where AbSpecAlign achieves an AAR of 46.09% and an RMSD of 2.48 Å. When τ is too small, the

contrastive objective becomes overly sharp and the scorer is dominated by a small number of hard negative antigen pairs, which can make the learned reward less stable during post-training. In contrast, when τ is too large, the contrastive logits are over-smoothed, reducing the separation between cognate and non-cognate antigen pairs and weakening the specificity signal. The intermediate value $\tau = 0.1$ therefore provides the best trade-off between discriminative negative-pair learning and stable reward calibration.

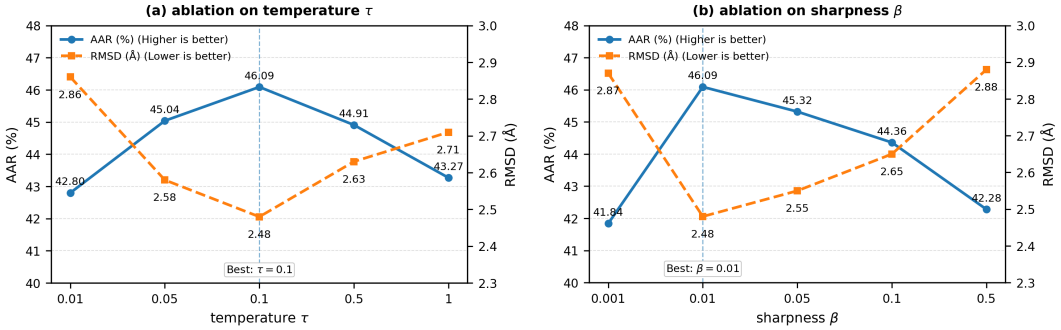


Figure 4: Ablation study on InfoNCE temperature τ, and β in Equation 8

D Visualization of reward-based candidate selection.

To further illustrate how the proposed specificity reward is used for candidate-level selection, we visualize representative generation groups in Figure 5. For each target antigen, the generator samples multiple candidates, and the frozen reward assigns a specificity score to each candidate. The candidate with the highest reward is selected as the final design. In each row of Figure 5, the leftmost column shows the highest-reward candidate selected, while the remaining columns show alternative generated candidates from the same group. This visualization highlights that AbSpecAlign does not rely on a single sampled structure, but instead ranks diverse antigen-conditioned candidates and selects the one with more favorable reward-estimated interface compatibility.

E Broader Impact Analysis

This work has potential positive social impacts by improving the efficiency of antigen-conditioned antibody design. By providing a plug-and-play specificity reward that can be attached to existing sequence–structure generators, AbSpecAlign can help prioritize antibody candidates with better antigen-specific structural fidelity and interface quality, thereby reducing the cost and iteration time of early-stage therapeutic and diagnostic antibody discovery. Such computational refinement could also reduce the reliance on large-scale wet-lab screening and animal-based discovery pipelines when used as an upstream filtering tool. However, the same capability also has potential negative social impacts. More effective antibody-generation systems may reduce the barrier to design biologically active binding molecules, including candidates against sensitive or dual-use targets, if used without appropriate oversight. Furthermore, because our current evaluation is entirely in silico and lacks wet-lab validation, overinterpreting computational scores could lead to premature advancement of unsafe or ineffective candidates. Dataset bias, incomplete modeling of off-target binding, immunogenicity, manufacturability, and toxicity may further limit real-world reliability. Therefore, AbSpecAlign should be used only as a computational prioritization component within regulated biomedical workflows, accompanied by experimental validation, off-target assessment, biosafety review, and responsible access controls for high-risk applications.

F Limitations and Future Work

AbSpecAlign has three main limitations. First, all reported evaluations are conducted *in silico*. The reward model and the post-training procedure improve sequence recovery, structural fidelity, and docking quality on benchmark complexes, but these metrics do not replace experimental validation

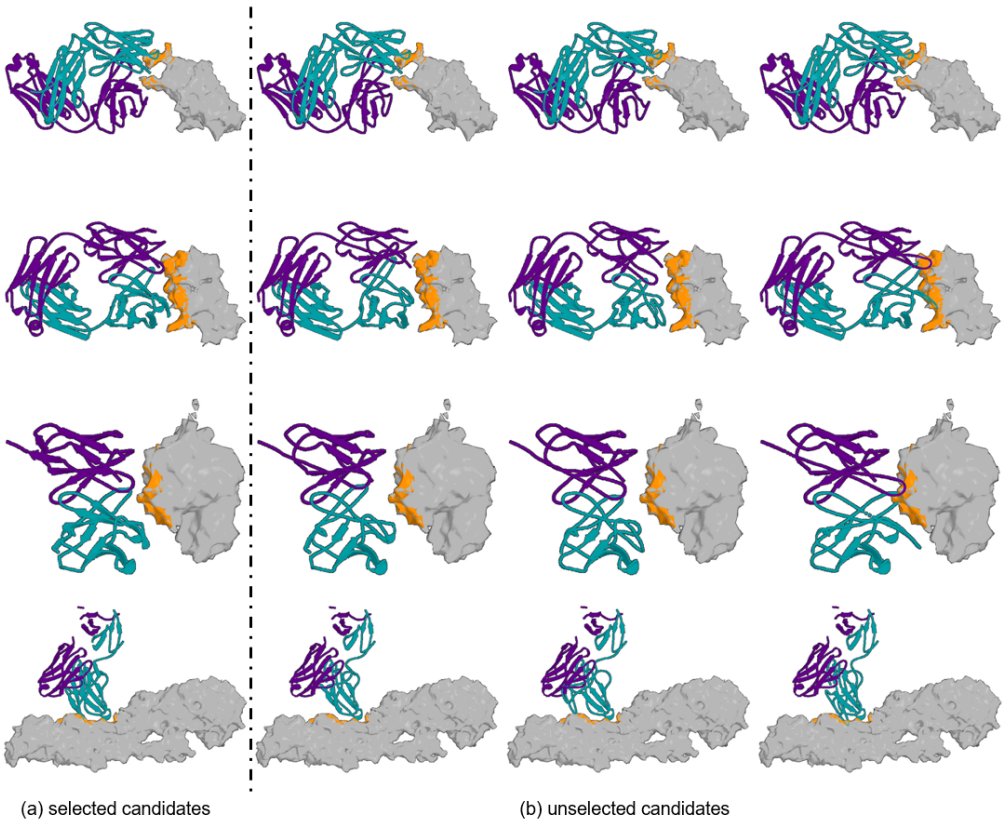


Figure 5: Visualization of candidate selection. Each row corresponds to one target antigen. For each antigen, multiple antibody candidates are generated and scored by the frozen reward. The leftmost column shows the selected highest-reward candidate, while the remaining columns show alternative generated candidates from the same group.

of binding affinity, specificity, and developability. Second, the specificity reward is trained from available structural complexes and sampled non-cognate antigens. Its discrimination ability is therefore constrained by the diversity and coverage of SAbDab and by the negative-sampling strategy. Closely related antigens that are absent from the training distribution may still be challenging. Third, post-training depends on the quality of the pre-trained generator. When the generator produces unrealistic structures or samples outside the support of the training data, the frozen scorer may provide unreliable gradients or rankings.

Future work should combine AbSpecAlign with prospective wet-lab screening, harder negative mining based on structurally similar antigens, and developability-aware rewards. These extensions would allow the framework to optimize specificity while controlling additional properties such as solubility, aggregation risk, and manufacturability.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The claims in the abstract and introduction strictly follow the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: We discuss the limitations in the Appendix Section E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes].

Justification: Please see Section 3 and Appendix Section A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes].

   Justification: We provide the experimental settings in Section 4.1 and Appendix Section B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No].

Justification: We will make the data and code publicly available upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Please see Section 4.1 and Appendix Section B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: Based on the computational cost, we only eported error bars in ablation study shown in Table 2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: We reported the GPU usage in Section 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: Our work conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: Please see Appendix Section D.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: All third-party codebases and datasets used in this work are open-source and publicly available.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: New assets will be made available upon acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: This paper does not contain any crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.