5 February 2025, Preprint v1 · CC-BY 4.0

Research Article

AlignAb: Pareto-Optimal Energy Alignment for Designing Nature-Like Antibodies

Yibo Wen[1,2], Chenwei Xu[1,3], Jerry Yao-Chieh Hu[1,2], Han Liu[1,2,3]

1. Center for Foundation Models and Generative AI, Northwestern University, United States; 2. Department of Computer Science, Northwestern University, United States; 3. Department of Statistics and Data Science, Northwestern University, United States

We present a three-stage framework for training deep learning models specializing in antibody sequence-structure co-design. We first pre-train a language model using millions of antibody sequence data. Then, we employ the learned representations to guide the training of a diffusion model for joint optimization over both sequence and structure of antibodies. During the final alignment stage, we optimize the model to favor antibodies with low repulsion and high attraction to the antigen binding site, enhancing the rationality and functionality of the designs. To mitigate conflicting energy preferences, we extend AbDPO (Antibody Direct Preference Optimization) to guide the model towards Pareto optimality under multiple energy-based alignment objectives. Furthermore, we adopt an iterative learning paradigm with temperature scaling, enabling the model to benefit from diverse online datasets without requiring additional data. In practice, our proposed methods achieve high stability and efficiency in producing a better Pareto front of antibody designs compared to top samples generated by baselines and previous alignment techniques. Through extensive experiments, we showcase the superior performance of our methods in generating nature-like antibodies with high binding affinity consistently.

1. Introduction

Antibodies are large, Y-shaped proteins that play a crucial role in protecting the human body against various disease-causing antigens [1]. As shown in Figure 1, an antibody consists of two identical heavy chains and two identical light chains. Antibodies have remarkable abilities to bind a wide range of antigens, and the tips of the Y shape exhibit the most variability [2][3]. These critical regions, composed of specific arrangements of amino acids, are known as Complementarity Determining Regions (CDRs) since their shapes complement those of antigens. To a great extent, the CDRs at the tips of light and heavy chains determine an antibody's specificity to antigens [4]. Hence, the key challenge in antibody design is identifying and designing effective CDRs as part of the antibody framework that bind to specific antigens.

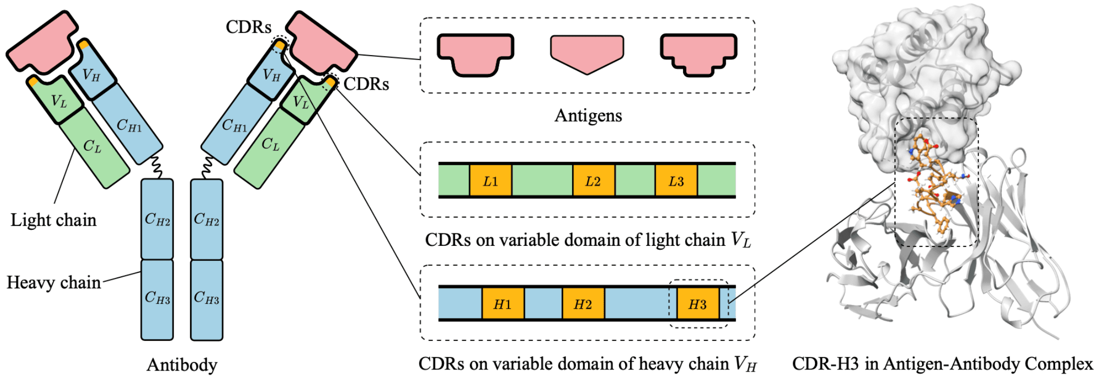

CDRs
CDRs
V_H
V_L
C_{H1}
C_L
C_{H2}
C_{H3}
Light chain
Heavy chain
Antibody
Antigens
L1
L2
L3
CDRs on variable domain of light chain V_L
H1
H2
H3
CDRs on variable domain of heavy chain V_H
CDR-H3 in Antigen-Antibody Complex

Figure 1. Illustration of an antibody binding to an antigen. The antibody's light and heavy chains are shown with their variable (V) and constant (C) regions. The third CDR in the heavy chain (CDR-H3), colored in orange, is critical for determining the binding affinity to the antigen.

Recently, various deep learning based methods achieve great success in the long-standing problem of antibody design and optimization. For example, [5] and [6] borrow ideas from language models and treat proteins as sequences to predict their structures, functions, and other important properties. These methods benefit from having access to large datasets with millions of protein sequences, but often lead to subpar results in generation tasks conditioned on protein structures [7][8]. Due to the determinant role of structure in protein function, co-designing sequences with structures emerges as a more promising approach [9][10][11][12]. Among all, diffusion-based methods stand out by learning the reverse process of transforming

doi.org/10.32388/OVBKGZ

desired protein structures from noise [13][14][8]. These methods achieve atomic-resolution antibody design and state-of-the-art results in various tasks, including sequence-structure co-design, fix-backbone CDR design, and antibody optimization [15][16].

Despite the prevalence of generative models, two key problems persist in effective antibody sequence-structure co-design. **First**, datasets containing complete 3D structures of antibodies are orders of magnitude smaller than sequence-only datasets. For example, the most common dataset for antibody design, SAbDab [17], only contains a few thousand antibody structures despite daily updates. The scarcity of high-quality antigen-antibody pairs, coupled with high variability of CDR structures [2], further constrains the performance of learning-based approaches. **Second**, existing methods overlook energy functions during supervised training and struggle to generate antibodies with low repulsion and high binding affinity. Contrary to traditional computational methods, recent efforts [15][11][12][18] shift their focus from searching for minimal energy states to optimizing metrics such as Amino Acid Recovery (AAR) and Root Mean Square Deviation (RMSD). However, these metrics are prone to manipulation, often fail to differentiate between different error types, and ignore important side chain structures in CDR-antigen interactions [16]. Overreliance on these metrics gives rise to irrationality in generated structures and widens the gap between *in silico* and *in vitro* antibody design.

To address the aforementioned challenges, we introduce a three-stage training pipeline focusing on rationality and functionality for antibody design. Inspired by the recent success of Large Language Models, we adopt a similar training paradigm comprising pre-training, transferring and alignment.

1. **Pre-training.** We first utilize a pre-trained antibody language model, trained on millions of amino acid sequences, to alleviate the shortage of structured antibody data. This approach enables the model to capture underlying relationships between proteins and internalize fundamental biological concepts such as structure and function [6][19].
2. **Transferring.** We then leverage the learned representations extracted from the language model to train a smaller model on a curated dataset of antibody-antigen pairs, allowing the model to adapt to the specific task of antigen-specific antibody design. The diffusion-based model is then able to recover not only sequences but also coordinates and side-chain orientations of each amino acid conditioned on the entire antigen-antibody framework [15].
3. **Alignment.** For the final stage, we conduct energy-based alignment of the diffusion model using Pareto-Optimal Energy Alignment as an extension of Direct Preference Optimization (DPO) [20]. By reusing designs generated by the model and labeling them with biophysical energy measurements, we compel the model to favor antibodies with lower repulsion and higher affinity in a data-free fashion. Additionally, we introduce an iterative version of the alignment algorithm in an online setting, allowing the model to benefit from online exploration. To balance exploration and exploitation during alignment, we propose decaying temperature scaling during the sampling process. Empirical results verify that our methods surpass existing alignment methods, consistently generating antibodies with energies closer to Pareto optimality.

In summary, our main contributions are:

- We devise the first three-stage training framework for antibody sequence-structure co-design, consisting of pre-training, transferring, and alignment.
- We propose an efficient multi-objective alignment algorithm with online exploration which consistently produces a better Pareto front of models in terms of energy without extra data.
- Our approach achieves state-of-the-art performance in generating more natural-like antibodies with better rationality and functionality.

2. Related Work

Computational Antibody Design. Deep learning based methods are now widely used for antibody design, with many latest work incorporating generative models [21][22][23][24]. [11] introduce HERN, which uses hierarchical message passing networks to encode both atoms and residues in an autoregressive manner. [25] propose MEAN, utilizing E(3)-equivariant graph networks to better capture the geometrical correlation between different components. Additionally, [26] propose dyMEAN, focusing on epitope-binding CDR-H3 design and modeling full-atom geometry. [15] propose a diffusion model that uses residue type, atom coordinates, and side-chain orientations to generate antigen-specific CDRs. [8] propose Ab-Diffuser, which incorporates more domain knowledge and physics-based constraints.

Diffusion-based Generative Models. Diffusion models are a type of generative model with an encoder-decoder structure. It involves a Markov-chain process with diffusion steps to add noise to data (encoder) and reverse steps to reconstruct desired data from noise (decoder) [27][28][29]. DDPM [30] is one of the most well-known diffusion models utilizing this process. [31] propose DDIM, which is an improved version of DDPM that reduces the number of steps in the generation process. Score-matching [32][33][34] is also a popular research area in diffusion models. The key idea of score-matching is to use

doi.org/10.32388/OVBKGZ

Langevin dynamics to generate samples and estimate the gradient of data distribution. Later, [35] propose a solver for faster sampling in the context of score-matching methods using stochastic differential equations.

Alignment of Generative Models. Preference alignment during fine-tuning improves the quality and usability of generated data. Reinforcement Learning (RL) is one popular approach to align models with human preferences, and RLHF [36] is an example of such algorithm. [37] propose DPO as an alternative approach to align with human preferences. Different from RL-based approaches, DPO achieves higher stability and efficiency as it does not require explicit reward modeling. Building upon DPO, recent work such as DDPO [38], DPOK [39], and DiffAC [40] demonstrate the possibility of adapting existing alignment techniques to various generative models. SimPO [41] improves DPO by using the average log probability of a sequence as the implicit reward.

3. Preliminaries

3.1. Problem Definition

Each amino acid is represented by its type $s_i \in \{\text{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}\}$, coordinate $\mathbf{x}_i \in \mathbb{R}^3$, and orientation $\mathbf{O}_i \in \mathrm{SO}(3)$, where $i \in \{1, \ldots, N\}$. Here, N is total number of amino acids in the protein complex which may contain multiple chains [15].

In this work, we focus on the specific problem of designing CDR, a critical functioning component of the antibody, given the remaining antibody and antigen structure. Let the CDR of interest consists of m amino acids starting from index $l+1$ to $l+m$ on the entire antibody-antigen framework with a total of N amino acids. We denote the target CDR as $\mathcal{R} = \{(s_j, \mathbf{x}_j, \mathbf{O}_j) | j = l+1, \ldots, l+m\}$ and the given antibody-antigen framework as $\mathcal{F} = \{(s_i, \mathbf{x}_i, \mathbf{O}_i) | i \in \{1, \ldots, N\} \backslash \{l+1, \ldots, l+m\}\}$. Therefore, our objective is to model the conditional distribution $P(\mathcal{R}|\mathcal{F})$.

3.2. Direct Preference Optimization

To tackle the common issues of fine-tuning with Reinforcement Learning (RL), [42] propose DPO as an alternative for effective model alignment. In the setting of DPO, we have an input x and a pair of output (y_1, y_2) from dataset $\mathcal{D}$, and a corresponding preference denoted as $y_w \succ y_l | x$ where y_w and y_l are the "winning" and "losing" samples amongst (y_1, y_2) respectively. According to Bradley-Terry (BT) model [43], for a pair of output, the human preferences are governed by a ground truth reward model $r(x, y)$ such that BT preference model is

$$p(y_1 \succ y_2 | x) = \sigma(r(x, y_1) - r(x, y_2)), \tag{3.1}$$

where $\sigma(\cdot)$ is sigmoid. Then, the optimal policy π_r^* is defined by maximizing reward:

$$\pi_r^* = \operatorname{argmax}_\pi \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi(y|x)} \left[r(x, y) - \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} \right], \tag{3.2}$$

where β is the inverse temperature controlling the KL regularization. By solving (3.2) analytically, [42] give a relation between the ground-truth reward and optimal policy:

$$r(x, y) = \beta \log \frac{\pi_r^*(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x), \text{ where } Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp(r(x, y)/\beta). \tag{3.3}$$

This allows us to rewrite BT preference model (3.1) without reward model r (only in $\pi_r^*, \pi_{\text{ref}}$):

$$p(y_w \succ y_l \mid x) = \sigma\left(\beta \log \frac{\pi_r^*(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_r^*(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right). \tag{3.4}$$

In this way, the maximum likelihood reward objective for a parameterized policy π_θ becomes:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]. \tag{3.5}$$

This derived loss function bypasses the need for explicit reward modeling, enabling an RL-free approach for preference optimization. While DPO is first designed for language models, we can re-formulate it for diffusion models and arrive at a similar differentiable objective following [20], or see A.3 for details.

4. Methodology

In this section, we present our energy alignment method for designing nature-like antibodies, named **AlignAb**. We introduce Pareto-Optimal Energy Alignment to fine-tune the model under conflicting energy preferences in 4.1. Then, we present an iterative version of the algorithm and discuss how to mitigate mode collapse during sampling with temperature scaling in 4.2. Finally, we summarize the alignment algorithm and three-stage training framework in Section 4.3.

4.1. Pareto-Optimal Energy Alignment (POEA)

Pre-trained models often struggle to produce natural-like antibodies because they tend to ignore important physical properties during the optimization process. These physical properties manifest themselves as various energy measurements such as Lennard-Jones potentials (accounting for attractive and repulsive forces), Coulombic electrostatic potential and hydrogen bonding energies [44]. We aim to close this gap by aligning the pre-trained model to favor antibodies with low repulsion and high attraction energy configurations at the binding site. While AbDPO [16] demonstrates the potential of naïve DPO in antibody design, there are two primary distinctions in this context:

- (D1) The ground-truth reward model, given by energy measurements, is available.
- (D2) There are multiple, often conflicting, energy-based preferences.

Therefore, we propose Pareto-Optimal Energy Alignment to address (D1) by injecting ground-truth reward margin into the DPO loss, and (D2) by extending DPO to multiple preferences.

Incorporating Reward Model. Since we have access to the ground-truth reward model, it would be unwise to ignore this extra information and perform alignment with just the preference labels. We show how to extend DPO and incorporate the available reward values as part of the training objective. Let's consider a new reward function $r'(x,y) := r(x,y) + f(x)$ by adding the ground-truth reward model $r(x,y)$ and a random reward model $f(x)$ which depends only on the input. According to (3.3), we express $r'(x,y)$ in terms of its optimal policy under the KL constraint:

$$r'(x,y) = \beta \log \frac{\pi^*_{r'}(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta \log Z(x), \text{ where } Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x) \exp(r'(x,y)/\beta). \tag{4.1}$$

Note that $r'(x,y)$ and $r(x,y)$ induce the same optimal policy by construction (see Lemma A.2 and Appendix A.2 for details):

$$\pi^*_{r'} = \pi^*_r = \mathrm{argmax}_\pi \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(y|x)} \left[r(x,y) - \beta \log \frac{\pi(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} \right].$$

Then, we cast the random reward model $f(x)$ into a function of π^*_r and r:

$$f(x) = \beta \log \frac{\pi^*_r(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta \log Z(x) - r(x,y). \tag{4.2}$$

Finally, we replace $r(x,y)$ with $f(x)$ in the original preference model $p(y_1 \succ y_2 \mid x) = \sigma(r(x,y_1) - r(x,y_2))$ and hence DPO loss (3.5) becomes below loss over the parametrized model π_θ as

$$-\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)} - \Delta_r \right) \right], \tag{4.3}$$

where $\Delta_r := r(x,y_w) - r(x,y_l)$ is the positive reward margin between y_w and y_l. Notably, the obtained loss differs from the vanilla DPO loss (3.5) by including an additional reward margin Δ_r. To better understand how the derived loss facilitates the alignment process, we take the gradient of the loss and interpret each term individually:

$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\underbrace{\sigma(\tilde{r}_\theta(x,y_l) - \tilde{r}_\theta(x,y_w) + \Delta_r)}_{\text{(I): combined sample weight}} \left[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{(II): increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{(III): decrease likelihood of } y_l} \right] \right],$$

where $\tilde{r}_\theta(x,y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\mathrm{ref}}(y|x)}$ is the implicit reward defined by the models. Similar to the DPO gradient, term (II) and (III) aim to increase the likelihood of the preferred sample y_w and decrease that of the dispreferred sample y_l. However, the key distinction lies in the weighting of each sample pair in term (I). Our weighting term incorporates both the implicit reward margin, $\tilde{r}_\theta(x,y_w) - \tilde{r}_\theta(x,y_l)$, and the explicit ground-truth reward margin Δ_r. This meets our expectation as a larger reward gap between the sampled pair would result in a more pronounced adjustment in the model's weights.

Multi-Objective Alignment. Given n ground-truth reward models $\mathbf{r} = [r_1, \ldots, r_n]^\top$, we construct a dataset $\hat{\mathcal{D}} = \{(x_i, y_i, \mathbf{r}(x, y_i))\}$ that records the reward values for each input and its corresponding output. In practice, each reward value is an energy measurement associated with certain physical properties. Following [45], the goal for multi-objective preference alignment is not to learn a single optimal model but rather a Pareto front of models $\{\pi^*_{\hat{r}} \mid \hat{r} = \mathbf{w}^\top \mathbf{r}, \mathbf{w} \in \Omega\}$ and each solution optimizes for one specific collective reward model $\hat{r}$:

$$\pi^*_{\hat{r}} = \mathrm{argmax}_\pi \mathbb{E}_{x,y\sim\hat{\mathcal{D}}} \left[\hat{r}(x,y) - \beta \log \frac{\pi(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} \right], \tag{4.4}$$

where $\mathbf{w} = [w_1, \ldots, w_n]^\top$ s.t. $\sum_{i=1}^n w_i = 1$ is a weighting vector in the preference space Ω. To obtain a preference pair (x, y_w, y_l), we first select two random data points $(x, y_i, \mathbf{r}(x, y_i))$ and $(x, y_j, \mathbf{r}(x, y_j))$ from $\hat{\mathcal{D}}$ and then compute their collective rewards $\hat{r}(x, y_i)$ and $\hat{r}(x, y_j)$. Among (y_i, y_j), we assign $y_w \succ y_l \mid x$ which satisfies $\hat{r}(x, y_w) > \hat{r}(x, y_l)$.

To incorporate multiple preferences, we replace the original reward model r in (3.3) with the collective reward model $\hat{r} = \mathbf{w}^\top \mathbf{r}$ and arrive at a Pareto-Optimal-Energy-Alignment (**POEA**) loss:

$$\mathcal{L}_{\text{POEA}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\hat{\mathcal{D}}} \left[\log\sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} - \Delta_{\hat{r}} \right) \right], \tag{4.5}$$

where $\Delta_{\hat{r}} := \hat{r}(x, y_w) - \hat{r}(x, y_w)$. This simple formulation inherits the desired properties from its single-objective counterpart, ensuring that it produces the optimal model $\pi_{\hat{r}}$ for each specific $\mathbf{w}$. In practice, we calculate the reward margin with energy measurements following Equation (D.4).

4.2. Iterative Alignment with Temperature Scaling

Iterative Online Alignment. To further exploit the available reward model, we develop an iterative version of our alignment method as an analogy to online reinforcement learning (RL). Instead of relying on a large offline dataset collected prior to training as in AbDPO [16], our approach starts with an empty dataset and augments it with an online dataset constructed by querying the current model at the start of each iteration. This method mirrors how online RL agents gather data and learn by interacting with the environment, enabling continuous policy improvement. We present the detailed algorithm in Algorithm 1. Ideally, we are able to repeat the process until no further improvement is observed, and we select the best model based on validation metrics. Our experiments suggest that this online exploration leads to substantial performance gains, even when utilizing a much smaller dataset compared to offline learning, as shown in Section 5.3.

Temperature Scaling. While CDRs exhibit significant sequence variation within antibodies [2], parameterized neural networks often struggle to capture this diversity and suffer from mode collapse during training [46]. By measuring the entropy $H = -\sum p \log p$ of generated sequences, we observe a notable gap between the diversity of generated and natural CDR-H3 sequences as shown in Table 1. This implies possible model collapse during model training (see the comparison between 100k and 200k training steps in Table 1). To combat this, we apply temperature scaling to the pre-trained diffusion model during the inference process.

Method	**Entropy** (↑)
Reference	3.95
MEAN	2.18
DiffAb (100k step)	3.57
DiffAb (200k step)	3.29
DiffAb-TS	**3.84**

Table 1. CDR-H3 entropy

Temperature scaling adjusts the logits before applying the softmax function to control the randomness (i.e., entropy) of generated sequences. The scaled softmax is given by: $\text{Softmax}(z_i/T) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$ where T is the temperature. Higher temperatures encourage diversity, while lower temperatures encourage predictability. Since our diffusion model uses multinomial distribution to model antibody sequences (as described in Appendix A.1), we inject a small temperature scale to enhance the sample diversity at inference time. Inspired by epsilon-greedy learning from RL, we adopt a decaying temperature schedule, achieving a balance between exploration and exploitation.

We validate this approach by applying a small temperature scale ($T = 1.5$) to the pre-trained diffusion model DiffAb [15]. The resulting model, DiffAb-TS, produces sequences that match the diversity of natural CDR-H3 sequences, as shown in Table 1. Through ablation studies in Section 5.3, we further demonstrate the effect of temperature scaling during alignment.

Algorithm 1 Iterative Pareto-Optimal Energy Alignment

```
Input: Initial dataset $\hat{\mathcal{D}}_0 = \emptyset$, KL regularization $\beta$, online iterations $T$, batch size $m$, reference model $\pi_{\text{ref}}$, initial model $\pi_0 = \pi_{\text{ref}}$, and reward model $\hat{r}$.
for $t = 0, 1, 2, \cdots, T$ do
    Observe $x_i \sim \mathcal{X}$, and sample $y_i^1, y_i^2 \sim \pi_t(\cdot \mid x)$ for all $i \in [m]$.
    Calculate rewards $\hat{r}(x_i, y_i^1)$ and $\hat{r}(x_i, y_i^2)$ for all $i \in [m]$, and collect them as $\hat{\mathcal{D}}_t$.
    Optimize $\pi_{t+1}$ with $\hat{\mathcal{D}}_{0:t}$ according to (4.5):
        $$\pi_{t+1} \leftarrow \underset{\pi}{\operatorname{argmin}} \, \mathbb{E}_{(x, y_w, y_l) \sim \hat{\mathcal{D}}_{0:t}} \left[ \log \sigma \left( \beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} - \Delta_{\hat{r}} \right) \right].$$
end for
Output: Choose the best model in $\pi_{0:T}$ by a validation set.
```

4.3. Three-Stage Training Framework

Inspired by the recent success of large language models, we adapt the widely used 3-stage training framework to the task of antibody design in combination with our devised alignment method.

- **Pre-training.** Due to the limited availability of structured antibody data, we leverage the abundant online antibody sequences for pre-training using a BERT-based model [47]. Following[7], we employ a masked language modeling objective, where we mask all residues within CDRs and aim to recover them. This approach enables the antibody language model to learn expressive representations that capture the underlying relationships between proteins and internalize fundamental biological concepts such as structure and function.
- **Transferring.** We use the pretrained BERT model as a frozen encoder to train a downstream diffusion model. Specifically, this transfers learned representations to the diffusion model for antibody generation (see details of embedding fusion in Appendix E.1). Crucially, this representation enhancement addresses the challenge of antigen-specific antibody design: datasets are limited and curated by human experts. The diffusion-based model recovers sequences, coordinates, and orientations of each amino acid, conditioned on the entire antigen-antibody framework. For detailed formulation on diffusion models for antibody generation, see Appendix A.1.
- **Alignment.** Lastly, we align the trained diffusion model via Pareto-Optimal-Energy-Alignment (POEA) from (3.2), an extended version of multi-objective DPO-diffusion for antibody design. Importantly, the Pareto weight $\mathbf{w}$ allows us to incorporate designers' preferences, enabling balanced control over multiple objectives (physical, chemical, and biological properties) by domain experts. In summary, we propose POEA (3.2) to address issues of conflicting energy preferences and potential mode collapse during the alignment stage. We take advantage of ground-truth reward models (see detailed reward calculations in Appendix D) by incorporating reward margin in the loss function and utilizing online exploration datasets.

5. Experimental Studies

We evaluate our proposed framework, named **AlignAb**, for the task of designing antigen-binding CDR-H3 regions. We first present the general experiment setup for the three training stages, then describe the evaluation metrics and discuss the final results in this section.

5.1. Experiment Setup

Energy Definitions. We introduce four key energy measurements where we use the first two to evaluate the rationality and functionality of antibodies and use the rest to generate preferences during alignment. To determine the rationality and functionality of different CDR designs, we identify two key energy measurements CDR E_{total} and CDR-Ag ΔG.

1. CDR E_{total} represents the combined energy of all amino acids within the CDR, calculated using the default score function in Rosetta [48]. This energy is a strong indicator of structural rationality, as a higher E_{total} suggests large clashes between amino acids.
2. CDR-Ag ΔG represents the binding energy between the CDR and the antigen, determined using the protein interface analyzer in Rosetta [48]. This measurement reflects the difference in total energy when antibody is separated from antigen. Lower ΔG corresponds to higher binding affinity, serving as a strong indicator of structural functionality.

To generate energy-based preferences during model alignment, we use two fine-grained energy measurements: CDR-Ag E_{rep} and CDR-Ag E_{att}.

3. CDR-Ag E_{att} captures the attraction forces between the designed CDR and the antigen.
4. CDR-Ag E_{rep} captures the repulsion forces between the designed CDR and the antigen.

As suggested by [16], we further decompose E_{att} and E_{rep} at the amino acid level to provide more explicit and intuitive gradients. We include detailed calculation formulas for the energy measurements and their corresponding reward functions in Appendix D. We exclude CDR E_{total} and CDR-Ag ΔG measurements when determining the preference pairs because our experiments demonstrate that CDR-Ag E_{att} and CDR-Ag E_{rep} are sufficient for effective model alignment. This simplification reduces the computational cost associated with tuning multiple weights for different reward models, resulting in a more efficient and stable alignment process.

Datasets. For pre-training, we utilize the antibody sequence data from the Observed Antibody Space database [49]. Following [7], we adopt the same preprocessing steps including sequence filtering and clustering. Since we focus on CDR-H3 design, we select 50 million heavy chain sequences to pre-train the model.

To transfer the knowledge, we use the antibody-antigen data with structural information from SAbDab database [50]. Following [25], we first remove complexes with a resolution worse than 4Å and renumber the sequences under the Chothia scheme [51]. Then, we identify and collect structures with valid heavy chains and protein antigens. We also discard duplicate data with the same CDR-H3 and CDR-L3. We use MMseqs2 [52] to cluster the remaining complexes with a threshold of 40% sequence similarity based on the CDR-H3 sequence of each complex. During training, we split the clusters into a training set of 2,340 clusters and a validation set of 233 clusters. For testing, we borrow the RAbD benchmark [44] and select 42 legal complexes not used during training.

For alignment, we avoid using additional datasets and only draw samples from the trained diffusion model. During each iteration, we first generate 1,280 unique CDR-H3 designs and collect them as the online dataset. Then, we reconstruct the full CDR structure including side chains at the atomic level using PyRosetta [48], and record the predefined energies for each CDR at residue level. We repeat this iterative process 3 times for each antibody-antigen complex in the test set.

Baselines. We compare AlignAb with 5 recent state-of-the-art antibody sequence-structure co-design baselines. **MEAN**[25] generates sequences and structures using a progressive full-shot approach. **HERN** [11] generates sequences autoregressively and refines structures iteratively. **dyMEAN** [26] generates designs with full-atom modeling. **ABGNN** [7] introduces a pre-trained antibody language model combined with graph neural networks for one-shot sequence-structure generation. **DiffAb** [15] utilizes diffusion models to model type, position and orientation of each amino acid. All methods except for MEAN is capable of generating multiple antibodies for a specific antigen. To ensure a fair comparison, we implement a random version of MEAN by adding a small amount of random noise to the input structure.

5.2. Antigen-binding CDR-H3 Design

Evaluation Metrics. To better measure the gap between designs generated by different models and natural antibodies, we use CDR E_{total} and CDR-Ag ΔG as defined above, rather than commonly used metrics such as AAR and RMSD. Additionally, we include CDR-Ag E_{att} and CDR-Ag E_{rep} used during model alignment. [16] argue these physics-based measurements are indispensable in designing nature-like antibodies and act as better indicators of the rationality and functionality of antibodies. Based on energy measurements, we compute energy gap as the mean absolute error relative to natural antibodies. We sample 1,280 antibodies using each method and perform structure refinement with the relax protocol in Rosetta [48]. To select the best sample from each test case, we aggregate rankings of CDR E_{total} and CDR-Ag ΔG.

Method	CDR E_{total}		CDR-Ag ΔG		CDR-Ag E_{att}		CDR-Ag E_{rep}		Gap	
	Top	Avg.	Top	Avg.	Top	Avg.	Top	Avg.	Top	Avg.
Reference	-19.33	-	-16.00	-	-18.34	-	18.05	-	-	-
MEAN	46.27	186.05	**-19.94**	26.14	-5.13	-5.16	7.77	29.21	31.16	73.14
HERN	7,345.11	10,599.92	640.50	2,795.15	-6.64	-1.98	**1.67**	36.88	1453.75	2416.97
dyMEAN	5,074.11	12,311.15	4,452.26	10,881.22	-12.62	-5.06	139.42	1,762.59	2422.10	6183.425
ABGNN	1315.34	3022.88	-11.52	**16.08**	-1.63	-0.48	22.15	**8.84**	354.38	778.54
DiffAb	-1.50	158.90	-6.18	260.30	-12.30	**-15.71**	18.63	603.58	19.74	263.44
AlignAb	**-6.37**	**30.45**	-8.81	25.16	**-14.89**	-14.81	15.52	56.22	**17.91**	**39.00**

Table 2. Summary of CDR E_{total}, CDR-Ag ΔG, CDR-Ag E_{att}, and CDR E_{rep} (kcal/mol) of reference antibodies, ranked top-1 antibodies and total antibodies designed by our model and other baselines (MEAN, HERN, dyMEAN, ABGNN, DiffAb). We compute the generation gap as the mean absolute error relative to reference. Lower values are better in all measurements. Our results show that our generated antibodies are closer to references compared to all baseline methods.

Results. We report the main evaluation results in Table 2. For the sake of completeness, we include additional metrics for RMSD and AAR in Table 3. We also provide additional visualization examples in Figure 4. Overall, AlignAb outperforms baseline methods and narrows the gap between generated and natural antibodies. Furthermore, AlignAb demonstrates the smallest difference between top samples and average samples, suggesting a higher consistency in the generated antibody quality.

While baseline methods possess lower values for certain energy measurements, the generated antibodies are often far from ideal. For instance, MEAN, despite achieving a low CDR-Ag ΔG, exhibits significantly higher CDR E_{total}, indicating less favorable overall interactions and potential structural clashes. HERN, dyMEAN and ABGNN show poor performance across most metrics, with high CDR E_{total} values, suggesting strong repulsion due to close antigen-antibody proximities. Comparatively, DiffAb demonstrates a more balanced approach. It benefits from the theoretically guaranteed diversity of diffusion models and produces a higher variance in the quality of the designed CDRs. This provides DiffAb a higher probability of generating high-quality top-1 designs compared to other baselines. Compared with DiffAb, AlignAb achieves better results in all but one energy measurement. Thanks to the proposed energy alignment, AlignAb reduces average CDR E_{total}, CDR-Ag ΔG and CDR-Ag E_{rep} by a large margin, while maintaining reasonable CDR-Ag E_{att} values. This indicates antibodies generated by AlignAb have fewer clashes and exhibit strong binding affinity to target antigens.

We anticipate further performance gains beyond current results with some simple modifications. Due to limited computational resources, we assign the same weight to the reward models across all test data (see Appendix E.2). By tuning the reward weightings, we can optimize the energy trade-offs between multiple conflicting objectives for each antigen-antibody complex, potentially resulting in a Pareto front of models. Additionally, increasing the sample size and number of iterations for alignment will likely enhance the overall performance and reliability of the generated antibodies. These preliminary results underscore the potential of AlignAb in generating nature-like antibodies. We include the full evaluation results in Table 4.

5.3. Ablation Studies

Our approach introduces three main novel designs for creating nature-like antibodies: Pareto-optimal energy alignment, iterative online exploration, and temperature scaling. To validate the effectiveness of each component, we conduct comprehensive ablation studies to demonstrate how these design elements contribute to the overall performance of our model. As an example, we apply our method to an antigen with PDB ID: 5nuz to illustrate the impact of each component. We provide additional examples of our ablation studies in Figure 3.

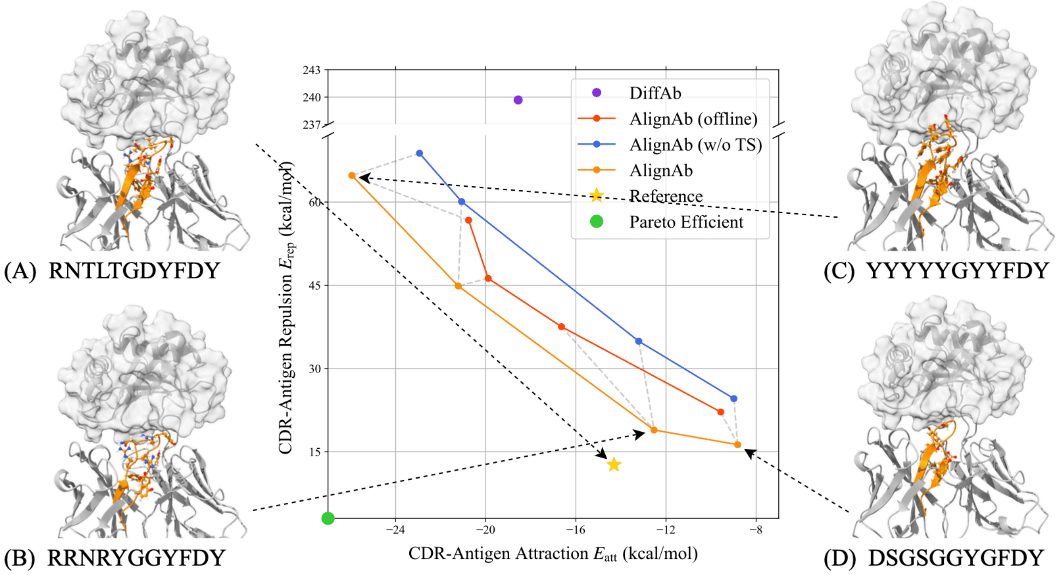


Figure 2. Frontiers of CDR-Ag E_{att} and CDR-Ag E_{rep} alignment and typical samples produced by different reward weightings in POEA. (A) is the reference CDR-H3 (colored in orange) from PDB ID 5nuz. (B) is the best CDR-H3 design generated by AlignAb with low overall energy and high similarity with the reference structure. (C) is the typical type of design when E_{att} reward dominates, and often consists of large side chains and contains structural collisions. (D) is the typical type of design when E_{rep} reward dominates, and often lack of side chains with weak binding with the antigen.

Pareto-Optimal Energy Alignment. To illustrate how our proposed algorithm resolves conflicting alignment objectives, we train a front of models by selecting different weightings of the reward models defined in Equation (4.4). For each set of reward weightings, we train the model for 2,000 steps and record the average CDR-Ag E_{att} and CDR-Ag E_{rep} by sampling 128 designs. To better understand the effects of different reward weightings between CDR-Ag E_{att} and CDR-Ag E_{rep}, we analyze two typical categories of inferior results caused by unbalanced reward models, as shown in Figure 2 (C) and (D). Specifically, when the weight for CDR-Ag E_{att} is too high, the model tends to generate sequences with large amino acids such as Tyrosine (Y) and Tryptophan (W), resulting in massive structural collisions. Conversely, when the weight for CDR-Ag E_{rep} is too high, the model tends to generate sequences with small amino acids, like Serine (S) and Glycine (G), resulting in low binding affinities with the target antigen. These examples highlight the importance of balancing reward weightings during alignment to design nature-like antibodies.

Online Exploration with Temperature Scaling. To show the effectiveness of online learning and temperature scaling, we compare the full AlignAb framework and its counterparts without the two modules. For AlignAb, we collect 1,280 samples at each iteration and repeat the alignment process for 3 times, each for 2k steps. For offline alignment with a fixed dataset (AlignAb offline), we collect 3,840 samples to match the total size of the dataset observed during iterative alignment. We also test the performance of AlignAb without temperature scaling during sampling (AlignAb w/o TS). As shown in Figure 2, the full training framework produces a better front of models in terms of CDR-Ag E_{att} and CDR-Ag E_{rep} and proves the necessity of both online exploration and temperature scaling. This matches our expectation as we show in Section 4.1 that using POEA loss in Equation (4.5) the model converges to optimality under different collective reward models.

6. Conclusion

In this work, we adapt the successful paradigm of training large language models to the field of antibody sequence-structure co-design. Our three-stage training pipeline addresses the key challenges posed by limited structural antibody-antigen data and the common oversight of energy considerations during optimization. During alignment, we optimize the model to favor antibodies with low repulsion and high attraction to the antigen binding site, enhancing the rationality and functionality of the designs. To mitigate conflicting energy preferences, we extend AbDPO in combination with iterative online exploration and temperature scaling to achieve Pareto optimality under multiple alignment objectives. Our proposed methods demonstrate high stability and efficiency, producing a superior Pareto front of antibody designs compared to top samples generated by baselines and previous alignment techniques. Future work includes further investigating the performance of the framework using larger fine-tuning datasets and extending our method to other structures such as small molecules.

Appendix A. Supplementary Backgrounds

A.1. Diffusion Processes for Antibody Generation

A diffusion probabilistic model consists of two processes: the forward diffusion process and the reverse generative process. Let T denote the terminal time, and $t \in [T]$ denote the diffusion time step. Let $\mathcal{R}^t = \{(s_j^t, \mathbf{x}_j^t, \mathbf{O}_j^t) | j = l+1, \ldots, l+m\}$ denote a sequence of latent variables sampled during the diffusion process, where $(s_j^t, \mathbf{x}_j^t, \mathbf{O}_j^t)$ is the intermediate state for amino acid j at diffusion step t. Intuitively, the forward diffusion process injects noises to the original data $\mathcal{R}^0$, while the reverse generative process learns to recover ground truth by removing noise from $\mathcal{R}^T$. To model both the sequence and structure of antibodies, [15] defines three separate diffusion processes for $q(\mathcal{R}^t | \mathcal{R}^0)$ as follows:

$$
\begin{aligned}
q(s_j^t \mid s_j^0) &= \mathcal{C}(1(s_j^t) \mid \overline{\alpha}^t \cdot 1(s_j^0) + (1 - \overline{\alpha}^t) \cdot \frac{1}{20} \cdot \mathbf{1}), \\
q(\mathbf{x}_j^t \mid \mathbf{x}_j^0) &= \mathcal{N}(\mathbf{x}_j^t \mid \sqrt{\overline{\alpha}^0} \cdot \mathbf{x}_j^0, (1 - \overline{\alpha}^0)\mathbf{I}), \\
q(\mathbf{O}_j^t \mid \mathbf{O}_j^0) &= \mathcal{IG}_{\mathrm{SO(3)}}(\mathbf{O}_j^t \mid \mathrm{ScaleRot}(\sqrt{\overline{\alpha}^t}, \mathbf{O}_j^0), 1 - \overline{\alpha}^t),
\end{aligned}
$$

where $\overline{\alpha}^t = \prod_{\tau=1}^{t}(1 - \beta^\tau)$ and $\{\beta^t\}_{t=1}^{T}$ is the predetermined noise schedule. Here, $\mathcal{C}$ denotes the categorical distribution defined on 20 types of amino acids; $\mathcal{N}$ denotes the Gaussian distribution on $\mathbb{R}^3$; $\mathcal{IG}_{\mathrm{SO(3)}}$ denotes the isotropic Gaussian distribution on SO(3). We use 1 to represent one-hot encoding function and $\mathrm{ScaleRot}$ to represent rotation angle scaling under a fixed axis.

To recover $\mathcal{R}^0$ from $\mathcal{R}^T$ given specified antibody-antigen framework $\mathcal{F}$, [15] defines the reverse generation process $p(\mathcal{R}^{t-1} \mid \mathcal{R}^t, \mathcal{F})$ at each time step as follows:

$$
\begin{aligned}
p(s_j^{t-1} \mid \mathcal{R}^t, \mathcal{F}) &= \mathcal{C}(s_j^{t-1} \mid f_{\theta_s}(\mathcal{R}^t, \mathcal{F})[j]), \\
p(\mathbf{x}_j^{t-1} \mid \mathcal{R}^t, \mathcal{F}) &= \mathcal{N}(\mathbf{x}_j^{t-1} \mid f_{\theta_{\mathbf{x}}}(\mathcal{R}^t, \mathcal{F})[j], \beta^t \mathbf{I}), \\
p(\mathbf{O}_j^{t-1} \mid \mathcal{R}^t, \mathcal{F}) &= \mathcal{IG}_{\mathrm{SO(3)}}(\mathbf{O}_j^{t-1} \mid f_{\theta_{\mathbf{O}}}(\mathcal{R}^t, \mathcal{F})[j], \beta^t),
\end{aligned}
$$

where all three f_θ are parameterized by SE(3)-equivariant neural networks and $f(\cdot)[j]$ denotes the output for amino acid j. Therefore, the training objective consists of three parts:

$$
\mathcal{L}_s^t = \mathbb{E}_{\mathcal{R}^t \sim p}\left[\frac{1}{m}\sum_{j=l+1}^{l+m} \mathbb{D}_{\mathrm{KL}}(q(s_j^{t-1} \mid s_j^t, s_j^0) \parallel p(s_j^{t-1} \mid \mathcal{R}^t, \mathcal{F}))\right], \tag{A.1}
$$

$$
\mathcal{L}_{\mathbf{x}}^t = \mathbb{E}_{\mathcal{R}^t \sim p}\left[\frac{1}{m}\sum_{j=l+1}^{l+m} \|\mathbf{x}_j^0 - f_{\theta_{\mathbf{x}}}(\mathcal{R}^t, \mathcal{F})[j]\|^2\right], \tag{A.2}
$$

$$
\mathcal{L}_{\mathbf{O}}^t = \mathbb{E}_{\mathcal{R}^t \sim p}\left[\frac{1}{m}\sum_{j=l+1}^{l+m} \|(\mathbf{O}_j^0)^\top f_{\theta_{\mathbf{O}}}(\mathcal{R}^t, \mathcal{F})[j] - \mathbf{I}\|_F^2\right]. \tag{A.3}
$$

Finally, the overall loss function is $\mathcal{L} = \mathbb{E}_{t \sim \mathrm{Uniform}(1, \cdots, T)}[\mathcal{L}_s^t + \mathcal{L}_{\mathbf{x}}^t + \mathcal{L}_{\mathbf{O}}^t]$. After training the model, we can use the reverse generation process to design CDRs given the antibody-antigen framework.

A.2. Optimal Policy of Equivalent Reward Functions

We cite the following definition and lemmas from DPO [42]:

Definition A.1. We say that two reward functions $r(x, y)$ and $r'(x, y)$ are equivalent iff $r(x, y) - r'(x, y) = f(x)$ for some function f.

Lemma A.1. Under the Plackett-Luce, and in particular the Bradley-Terry, preference framework, two reward functions from the same class induce the same preference distribution.

Lemma A.2. Two reward functions from the same equivalence class induce the same optimal policy under the constrained RL problem.

A.3. DPO for Diffusion Model Alignment

Here we review DPO for diffusion model alignment [20]. By alignment, we mean to align the diffusion models with users' preferences.

Let $\mathcal{D} := \{(x, y_w, y_l)\}$ be a dataset consisting an input/prompt x and a pair of output from a preference model p_{ref} with preference $y_w \succ y_l$. Our goal is to learn a diffusion model $p_\theta(y \mid x)$ aligning with such preference associated with p_{ref}. Let T denote the diffusion terminal time, and t denote the diffusion time step. Let $y^{1:T}$ be the intermediate latent variables and $R(y, y^{0:T})$ be the commutative reward of the whole markov chain such that

$$r(x, y^0) := \mathbb{E}_{p_\theta(y^{1:T}|x,y^0)}\left[R(y, y^{0:T})\right].$$

Aligning p_θ to p_{ref} needs

$$\max_{p_\theta}\left\{\mathbb{E}_{x\sim\mathcal{D}y^{0:T}\sim p_\theta(y^{0:T}|x)}\left[r(x, y^0)\right] - D_{\text{KL}}[p_\theta(y^{0:T} \mid x) \| p_{\text{ref}}(y^{0:T} \mid x)]\right\}.$$

Mirroring DPO (3.2), we arrive a ELBO-simplified DPO objective for diffusion model Appendix S.2 [20]:

$$\mathcal{L}_{\text{DPO-Diffusion}}(p_\theta, p_{\text{ref}}) \leq -\mathbb{E}_{\substack{(x_0^w, x_0^l)\sim\mathcal{D},\\ t\sim\mathcal{U}(0,T),\\ x_{t-1}^w, t\sim p_\theta(x_{t-1}^w|x_t^w),\\ x_{t-1}^l, t\sim p_\theta(x_{t-1}^l|x_t^l)}} \log\sigma\left(\beta T \log\frac{p_\theta(x_w^{t-1} \mid x_w^t)}{p_{\text{ref}}(x_w^{t-1} \mid x_w^t)} - \beta T \log\frac{p_\theta(x_l^{t-1} \mid x_l^t)}{p_{\text{ref}}(x_l^{t-1} \mid x_l^t)}\right),$$

where $\mathcal{U}$ denotes uniform distribution, β is KL regularization temperature. We remark this objective has a simpler form for empirical usage, see Eqn. 14 [20].

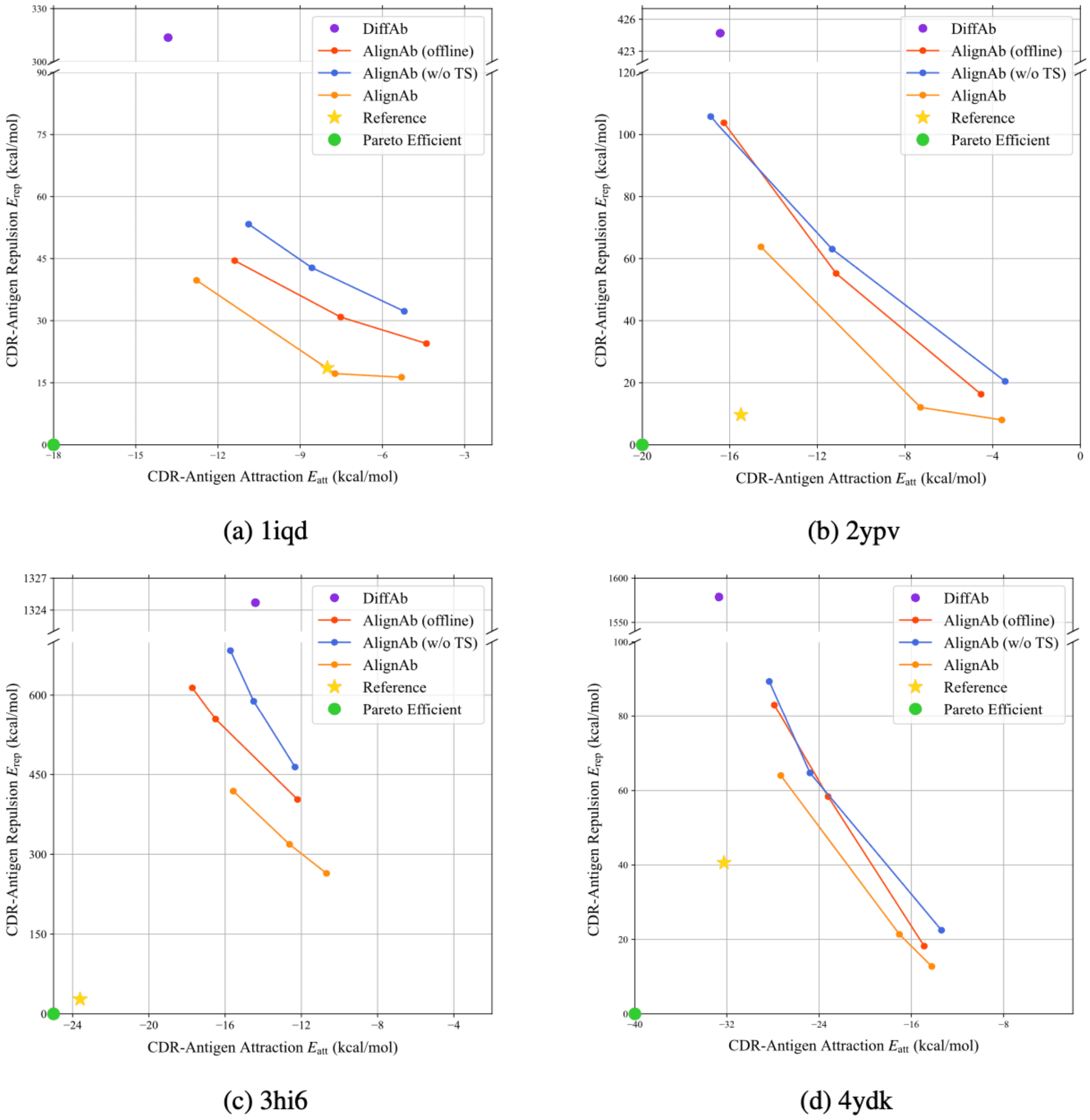
Appendix B. Additional Numerical Experiments

B.1. Additional Evaluation Metrics

Metrics	HERN	MEAN	dyMEAN	ABGNN	DiffAb	AlignAb*	AlignAb
AAR ↑	33.17	33.47	**40.95**	38.3	36.42	37.65	35.34
RMSD ↓	9.86	1.82	7.24	2.02	2.48	2.25	**1.51**

Table 3. Summary of AAR and RMSD metrics by our method and other baselines. We follow the default sampling settings from all baselines and use ranked top-1 samples generated by our method. AlignAb* indicates the AlignAb framework without the alignment stage.

B.2. Additional Ablation Examples



(a) 1iqd

(b) 2ypv

(c) 3hi6

(d) 4ydk

Figure 3. Frontiers of CDR-Ag E_{att} and CDR-Ag E_{rep} alignment produced by different reward weightings in POEA with four PDB examples.

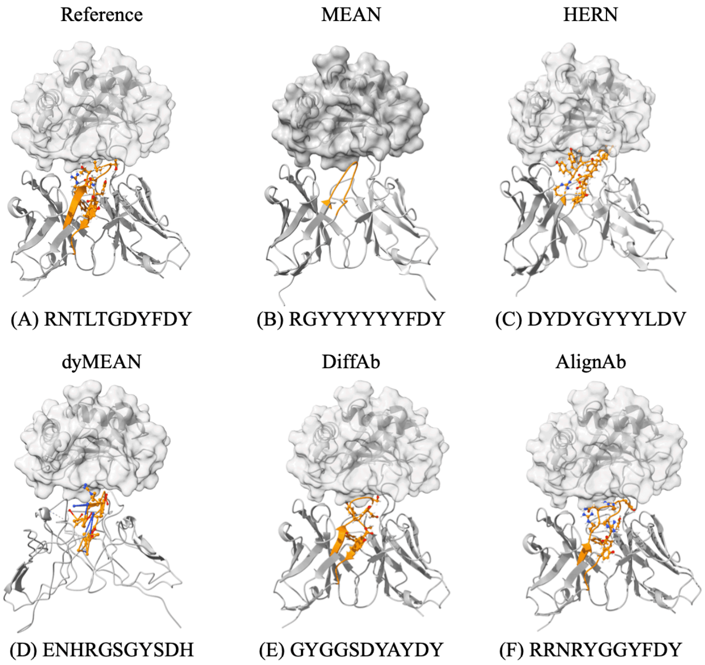
B.3. Detailed Evaluation Results

PDB ID	DiffAb				AlignAb				MEAN				ABGNN				reference		
	Total	Nonrep	Rep	dG	Total	Nonrep	Rep	dG	Total	Nonrep	Rep	dG	Total	Nonrep	Rep	dG	Total	Nonrep	Rep
1a14	-3.38	-13.60	21.18	3.71	1.18	-24.32	18.02	-0.73	51.17	-4.41	1.12	-15.65	1,251.38	-1.29	1.47	-15.12	-9.60	-16.04	16.12
1a2y	-9.43	-8.14	9.02	-20.42	-10.25	-18.94	14.94	-24.30	25.32	-0.79	0.62	-7.95	1,633.60	-3.25	2.54	-12.01	-21.18	-6.20	5.30
1fe8	2.15	-5.78	9.33	-6.52	0.37	-14.42	9.66	20.40	1.74	0.00	0.44	-9.59	1,054.35	0.00	0.00	-9.92	-18.24	-18.11	15.38
1ic7	1.75	-2.98	5.26	104.22	3.11	-1.42	4.22	101.33	9.07	-0.01	0.15	-4.83	843.27	0.00	0.00	-15.09	-7.06	-3.66	4.00
1iqd	5.10	-6.10	17.79	-5.85	3.98	-7.70	7.65	-21.13	16.70	-0.52	4.01	-28.84	951.63	-1.05	0.00	-19.38	-6.66	-8.01	18.59
1n8z	2.20	-17.08	30.05	32.23	6.24	-23.09	17.58	21.00	44.01	-4.83	5.05	-9.70	992.59	0.00	0.00	12.11	3.47	-20.73	15.93
1ncb	9.64	-12.98	30.98	73.88	1.66	-7.10	13.61	54.52	42.37	-1.43	8.18	-29.56	656.04	-4.36	4.78	-7.83	-4.68	-12.68	20.43
1osp	-0.97	-11.40	18.56	-10.47	-7.06	-17.39	26.50	-8.22	69.34	-9.59	11.47	-80.86	1,044.68	-2.54	2.51	-6.51	-23.03	-14.41	19.38
1uj3	-4.79	-9.68	23.43	21.18	-5.29	-12.96	17.67	13.95	-2.67	0.15	3.42	-23.86	1,024.76	-2.90	1.15	3.71	-12.27	-12.22	26.79
2adf	-5.15	-8.72	25.37	-15.88	-1.59	-10.66	20.15	-25.67	14.58	-1.92	8.87	-14.37	1,216.99	0.00	0.00	-20.54	-26.27	-25.54	16.20
2b2x	-3.00	-12.04	43.50	2.58	-2.62	-23.63	27.30	3.18	23.52	0.84	6.67	-10.04	1,216.00	-0.91	9.73	-13.26	-16.57	-12.61	20.90
2cmr	-9.02	-21.28	21.21	-11.00	-10.12	-19.82	22.29	-13.98	32.28	-8.56	4.04	-30.25	-	-	-	-	-28.71	-15.28	15.31
2dd8	-10.27	-9.62	10.67	22.36	-11.44	-9.66	13.31	17.80	16.99	-0.82	2.49	-14.16	562.62	0.00	0.00	-23.93	-11.28	-9.11	5.40
2vxt	8.56	-5.69	4.05	-31.77	4.48	-8.96	9.53	-31.72	4.16	3.74	2.42	-27.26	719.57	-1.58	0.58	14.50	-7.46	-4.87	7.08
2xqy	-8.62	-19.81	11.44	-40.41	-18.32	-22.06	19.19	-23.68	20.31	-4.93	8.54	-61.68	1,587.96	-8.25	3.57	-7.07	-14.51	-22.49	11.96
2xwt	2.28	-5.69	18.77	-54.84	-7.68	-9.46	11.64	-29.53	35.98	-1.65	5.21	-24.75	1,025.24	-4.23	4.75	-9.79	-20.15	-18.06	24.70
2ypv	5.07	-17.99	18.35	-12.20	1.74	-18.77	10.59	-20.24	71.25	-7.09	9.15	1.46	1,182.78	-10.14	6.91	-15.87	-20.15	-15.48	9.67
3hi6	1.04	-2.12	7.37	34.10	-6.46	-15.12	10.50	12.76	49.90	-8.21	8.38	-5.44	1,235.52	-2.38	0.56	-11.22	-18.35	-23.61	27.54
3k2u	4.81	-5.71	32.82	39.42	-1.55	-6.28	17.17	33.81	2.73	-0.82	0.58	-31.30	1,024.09	0.00	0.00	-21.06	3.70	-23.11	49.67
3mxw	-2.61	-6.74	5.77	-6.01	-7.36	-8.00	11.15	-10.50	32.38	-6.52	7.30	-6.42	1,597.12	0.74	2.98	-10.47	-13.94	-6.21	14.25
3s35	-7.41	-2.85	10.14	15.79	-4.80	-1.38	2.20	12.01	11.89	-0.91	4.62	-25.42	1,040.04	0.00	0.00	-19.82	-17.63	-5.99	5.68
4dvr	-11.21	-15.90	4.42	-13.20	-8.25	-17.83	6.92	-20.41	51.42	-5.04	3.63	10.37	1,176.08	0.00	0.00	-21.90	-25.13	-12.64	6.76
4g6j	-5.27	-9.85	26.69	-18.91	-9.29	-7.98	8.44	-18.32	14.18	-0.29	7.58	-4.91	899.87	0.00	0.00	-1.05	-10.34	-11.76	17.83
4g6m	0.76	-20.55	17.45	-22.17	1.02	-17.70	17.99	-22.13	30.97	-4.17	6.67	-15.02	1,788.05	0.00	0.00	-9.14	-14.16	-25.05	19.82
4h8w	0.57	-11.11	17.42	-19.39	-3.59	-15.08	13.11	-14.90	27.28	1.51	7.53	-19.58	1,737.00	0.46	0.58	-14.34	-17.13	-13.80	26.20
4ki5	-2.59	-7.89	25.40	-34.05	-7.42	-20.19	22.26	-32.08	115.73	-14.27	25.20	-101.61	1,101.30	0.00	0.00	-17.16	-58.90	-32.24	40.59
4lvn	5.51	-7.97	9.40	23.62	3.91	-10.40	11.47	17.08	52.74	-2.90	4.82	1.33	1,573.42	0.00	0.00	-16.69	-16.80	-24.52	20.92
4ot1	6.86	-17.43	24.19	-35.54	-18.19	-26.24	18.46	-70.11	168.30	-6.16	5.82	4.23	1,937.80	0.06	2.23	-9.41	-59.10	-27.58	19.46
4qci	-9.82	-12.69	8.85	-20.54	-14.19	-10.84	11.68	-24.92	32.22	-1.41	4.10	5.41	1,238.45	-0.10	0.00	-4.93	-17.78	-20.36	14.30
4xnq	-8.07	-9.36	16.43	-19.98	-14.48	-25.19	20.69	-42.07	82.24	-8.46	11.12	-4.47	1,657.84	0.00	0.00	-9.07	-26.21	-41.62	27.27
4ydk	-8.56	-34.89	48.45	-49.58	-19.03	-22.18	32.60	-91.78	164.79	-13.06	20.89	-52.66	2,078.24	0.00	0.00	-21.11	-58.90	-32.24	40.59
5b8c	1.79	-13.54	16.50	-5.04	-3.00	-4.77	19.13	12.11	43.88	-3.14	4.03	-32.62	1,973.32	-0.07	0.00	-14.98	-16.16	-22.64	25.06
5bv7	-9.80	-28.35	19.92	-86.88	-14.73	-21.44	21.01	-30.26	107.85	-21.52	28.63	-17.58	1,885.45	-2.79	0.84	-5.94	-32.05	-16.66	10.44
5d93	2.43	-11.97	5.37	54.56	-6.70	-12.74	8.06	46.18	9.57	0.00	4.64	-17.40	931.27	0.00	0.00	-14.07	-5.68	-13.48	11.39
5en2	0.88	-18.47	19.38	-40.37	-8.18	-20.55	17.19	-42.05	107.25	-18.51	16.29	-16.98	1,470.04	1.46	0.00	-15.06	-42.87	-25.64	11.91

PDB ID	DiffAb				AlignAb				MEAN				ABGNN				reference		
	Total	Nonrep	Rep	dG	Total	Nonrep	Rep	dG	Total	Nonrep	Rep	dG	Total	Nonrep	Rep	dG	Total	Nonrep	Rep
5f90	-3.31	-14.59	10.80	19.61	-10.45	-20.54	16.57	27.75	102.81	-17.36	28.27	-22.90	1,449.84	-3.62	3.39	-25.94	-15.83	-17.66	16.28
5ggs	-2.20	-14.62	36.29	-16.02	-10.87	-15.57	16.80	-21.65	22.19	-10.58	8.92	-12.95	1,813.13	0.00	0.00	-3.88	-26.34	-21.97	18.72
5hi4	8.84	-8.85	26.02	-20.39	3.73	-5.52	6.49	-21.09	10.03	0.03	1.92	-5.07	831.34	-4.94	1.21	-10.54	-17.45	-24.16	25.53
5j13	-4.91	-15.90	14.88	-41.02	-15.48	-23.48	17.11	-41.87	75.82	-12.28	12.14	-12.22	1,503.71	-0.04	0.00	-16.48	-15.85	-21.72	11.62
5l6y	-3.70	-20.66	23.83	-32.30	-12.39	-19.64	24.22	-27.57	78.93	2.35	6.54	-8.02	2,620.12	0.00	0.00	-4.56	-24.07	-28.83	14.06
5mes	0.73	-9.85	12.47	11.27	-7.95	-10.62	9.69	1.96	26.99	-3.57	5.08	-19.04	1,167.09	-5.76	0.64	-12.15	-20.00	-17.20	10.25
5nuz	0.18	-6.00	23.26	-27.38	-20.09	-15.74	27.22	-35.10	45.09	-18.48	9.93	-35.39	1,235.44	-9.42	857.92	-15.24	-31.07	-14.31	12.65

Table 4. Detailed evaluation results for various metrics for 42 antigens. The data source is the same as that in Table 2.

Appendix C. Additional Visualization



(A) RNTLTGDYFDY (B) RGYYYYYYFDY (C) DYDYGYYYLDV

(D) ENHRGSGYSDH (E) GYGGSDYAYDY (F) RRNRYGGYFDY

Figure 4. Visualization of reference antibody (PDB ID 5nuz) and different antibodies designed by our method and other baselines. The designed CDR-H3 structures are colored in orange and the designed CDR-H3 sequences are recorded accordingly.

Appendix D. Energy Calculation and Reward Models

In Section 5, we introduce the calculation of two functionality-associated energies, CDR-Ag E_{att} and CDR-Ag E_{rep}. Following [16], we denote the residue with the index i in the antibody-antigen complex as A_i. We then represent the **side chain** of the residue as A_i^{sc} and **backbone** of the residue as A_i^{bb}, respectively.

We define the interaction energies between a pair of amino acids as EP, with the default weights defined by REF15 [44]. EP consists of six different energy types: EP_{hbond}, EP_{att}, EP_{rep}, EP_{sol}, EP_{elec}, and EP_{lk}. Following the settings from Section 3, we define the indices of residues within the CDR-H3 range from $l+1$ to $l+m$, and the indices of residues within the antigen range from $g+1$ to $g+n$. Thus, for the CDR residue with the index j, the CDR-Ag E_{att} and CDR-Ag E_{rep} are defined as:

$$\text{CDR-Ag}E^{j}_{\text{att}} = \sum_{i=g+1}^{g+n} \sum_{\text{e}\in\{\text{hbond,att,sol,elec,lk}\}} \Big(\text{EP}_{\text{e}}(A^{sc}_{j}, A^{sc}_{i}) + \text{EP}_{\text{e}}(A^{sc}_{j}, A^{bb}_{i})\Big)\,, \qquad \text{(D.1)}$$

$$\text{CDR-Ag}E^{j}_{\text{rep}} = \sum_{i=g+1}^{g+n} \Big(\text{EP}_{\text{rep}}(A^{sc}_{j}, A^{sc}_{i}) + \text{EP}_{\text{rep}}(A^{sc}_{j}, A^{bb}_{i}) + 2\times\text{EP}_{\text{rep}}(A^{bb}_{j}, A^{sc}_{i}) + 2\times\text{EP}_{\text{rep}}(A^{bb}_{j}, A^{bb}_{i})\Big)\,. \qquad \text{(D.2)}$$

From Equations (D.1) and (D.2), we conclude that the interaction energy between the CDR and the antigen is determined by both side-chain and backbone interactions. The CDR-Ag E_{att} considers interactions primarily from side-chain atoms in the CDR-H3 region. In contrast, CDR-Ag E_{rep} assigns higher costs to repulsions from backbone atoms in the CDR-H3 region. This reason for the different is that side-chain atoms contribute most of the interaction energy between CDR-H3 and the antigen, as shown in Figure 1. Therefore, CDR-Ag E_{att} exhibits a benefit in interactions, while CDR-Ag E_{rep} represents repulsive costs.

To guide the model alignment process, we utilize the above two energy definitions to compute the final rewards as follows:

$$r_{\text{att}}(x,y) = -\sum_{i=l+1}^{l+m} \text{CDR-Ag}E^{j}_{\text{att}}, \qquad r_{\text{rep}}(x,y) = -\sum_{i=l+1}^{l+m} \text{CDR-Ag}E^{j}_{\text{rep}}, \qquad \text{(D.3)}$$

where lower energy corresponds to a higher reward. Therefore, we compute the final collective reward with predetermined weights as $\hat{r}(x,y) = w_{\text{att}} r_{\text{att}}(x,y) + w_{\text{rep}} r_{\text{rep}}(x,y)$. We observe the repulsion reward is often several orders of magnitude bigger than the attraction reward. Therefore, we utilize the following reward margin in our actual experiments:

$$\Delta_{\hat{r}} = \log(\hat{r}(x,y_w) - \hat{r}(x,y_l)). \qquad \text{(D.4)}$$

Appendix E. Implementation Details

E.1. Model Details

AlignAb consists of two parts: a pre-trained BERT model from AbGNN [7], and a pre-trained diffusion model from DiffAb [15]. For the pre-trained BERT model, our model uses a 12-layer Transformer model with a BERT$_{\text{base}}$configuration. We set the embedding size to 768 and the number of heads to 12. For the pre-trained diffusion model, our model takes the perturbed CDR-H3 and its surrounding context as input. For example, 128 nearest residues of the antigen or the antibody framework around the residues of CDR-H3. The input consists of both single and pairwise residue embeddings. The number of features with single residue embedding is 128. It consists of the encoded information of its amino acid types, torsional angles, and 3D coordinates of all heavy atoms. The number of features with pairwise residue embedding is 64. It consists of the encoded information of the Euclidean distances and dihedral angles between the two residues. To combine the feature embeddings with the hidden representations learned from the pre-trained BERT model, we extract the embedding for each residue from the final layer of the BERT model and concatenate it with the single and pairwise residue embeddings. We then utilize multi-layer perception (MLP) neural networks to process the concatenated embeddings. The MLP has 6 layers. In each layer, the hidden state 128. The output of this neural network is the predicted categorical distribution of amino acid types, C_α coordinates, a $so(3)$ vector for the rotation matrix.

The diffusion models aim to generate amino acid types, C_α coordinates, and orientations. Hence, for training the diffusion models, we take the output of MLP as input for diffusion models. We set the number of diffusion (forward) stets to be 100. For the noise schedules, we apply the same setting of DDPM [30], utilizing a β schedule with $s=0.01$. The noises are gradually added to amino acid types, C_α coordinates, and orientations.

E.2. Training Details

Transferring. We train the diffusion model part of AlignAb following the same procedure as [15]. The optimization goal is to minimize the rotation, position, and sequence loss. We apply the same weight to each loss during training. We utilize the Adam [53] optimizer with `init_learning_rate=1e-4, betas=(0.9,0.999), batch_size=16`, and `clip_gradient_norm=100`. We also utilize a learning rate scheduler, with `factor=0.8, min_lr=5e-6`, and `patience=10`. We perform evaluation for every 1000 training steps and train the model on one NVIDIA GeForce GTX A100 GPU, and it can converge within 36 hours and 200k steps.

Alignment. After obtaining the diffusion model, we further align it with energy-based preferences provided by domain experts. We utilize the Adam [53] optimizer with `init_learning_rate=2e-7, betas=(0.9,0.999), batch_size=8, clip_gradient_norm=100`. We set the KL regularization term $\beta=100.0$. In each batch, we select 8 pairs of energy-based preference data with labeled rewards. We do not use learning rate scheduling during alignment stage. For rewards, we set the w_{att} and w_{rep} with a fixed ratio 1:3. In each alignment iteration, we fine-tune the diffusion model for 4k steps. We repeat this process 3 times for each antigen in the test set.

References

1. ^Scott AM, Wolchok JD, Old LJ (2012). "Antibody therapy of cancer". Nature Reviews Cancer. **12**: 278-287. S2CID 205469234.
2. a, b, cCollis AVJ, Brouwer AP, Martin ACR (2003). "Analysis of the antigen combining site: Correlations between length and sequence composition of the hypervariable loops and the nature of the antigen". Journal of Molecular Biology. **325** (2): 337–354. doi:10.1016/S0022-2836(02)01222-6. Link.
3. ^Chiu ML, Goulet DR, Teplyakov A, Gilliland GL (2019). "Antibody Structure and Function: The Basis for Engineering Therapeutics". Antibodies. **8** (4): 55. doi:10.3390/antib8040055. PMID 31816964.
4. ^Akbar R, Robert PA, Pavlović M, Jeliazkov JR, Snapkov I, Slabodkin A, Weber CR, Scheffer L, Miho E, Haff IH, Haug DTT, Lund-Johansen F, Safonova Y, Sandve GK, Greiff V (2021). "A compact vocabulary of paratope-epitope interactions enables predictability of antibody-antigen binding". Cell Reports. **34** (11): 108856. doi:10.1016/j.celrep.2021.108856. ISSN 2211-1247. Link to article.
5. ^Madani A, Krause B, Greene ER, Subramanian S, Mohr BP, Holton JM, Olmos JL, Xiong C, Sun ZZ, Socher R, Fraser JS, Naik NV (2023). "Large language models generate functional protein sequences across diverse families". Nature Biotechnology. pages 1-8. S2CID 256304602.
6. a, bRives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, Guo D, Ott M, Zitnick CL, Ma J, Fergus R (2019). "Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences". PNAS. doi:10.1101/622803. Available from: https://www.biorxiv.org/content/10.1101/622803v4.
7. a, b, c, d, eGao K, Wu L, Zhu J, Peng T, Xia Y, He L, Xie S, Qin T, Liu H, He K, et al. Pre-training antibody language models for antigen-specific computational antibody design. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023. p. 506–517.
8. a, b, cMartinkus K, Ludwiczak J, Cho K, Liang WC, Lafrance-Vanasse J, Hotzel I, Rajpal A, Wu Y, Bonneau R, Gligorijevic V, Loukas A. "AbDiffuser: Full-Atom Generation of in vitro Functioning Antibodies". 2024. arXiv. arXiv:2308.05027.
9. ^Anishchenko I, Chidyausiku TM, Ovchinnikov S, Pellock SJ, Baker D (2020). "De novo protein design by deep network hallucination". bioRxiv. doi:10.1101/2020.07.22.211482. Available from: https://www.biorxiv.org/content/early/2020/07/23/2020.07.22.211482.
10. ^Harteveld Z, Southern J, Defferrard M, Loukas A, Vandergheynst P, Bronstein M, Correia B. Deep sharpening of topological features for de novo protein design. In: ICLR2022 Machine Learning for Drug Discovery; 2022. Available from: https://openreview.net/forum?id=DwN81YIXGQP.
11. a, b, c, dJin W, Barzilay R, Jaakkola T (2022). "Antibody-Antigen Docking and Design via Hierarchical Equivariant Refinement". arXiv. arXiv:2207.06616 [q-bio.BM].
12. a, bJin W, Wohlwend J, Barzilay R, Jaakkola T (2022). "Iterative Refinement Graph Neural Network for Antibody Sequence-Structure Co-design". arXiv. arXiv:2110.04624 [q-bio.BM].
13. ^Vinod R. Joint protein sequence-structure co-design via equivariant diffusion. In: NeurIPS 2022 Workshop on Learning Meaningful Representations of Life; 2022. Available from: https://openreview.net/forum?id=dq3g7Bl9of.
14. ^Lisanza SL, Gershon JM, Tipps S, Arnoldt L, Hendel S, Sims JN, Li X, Baker D (2023). "Joint Generation of Protein Sequence and Structure with RoseTTAFold Sequence Space Diffusion". bioRxiv. doi:10.1101/2023.05.08.539766. Available from: https://www.biorxiv.org/content/early/2023/05/10/2023.05.08.539766.
15. a, b, c, d, e, f, g, h, i, j, kLuo S, Su Y, Peng X, Wang S, Peng J, 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.
16. a, b, c, d, e, f, gZhou X, Xue D, Chen R, Zheng Z, Wang L, Gu Q (2024). "Antigen-Specific Antibody Design via Direct Energy-based Preference Optimization". arXiv preprint arXiv:2403.16576. Available from: https://arxiv.org/abs/2403.16576.
17. ^Dunbar J, Krawczyk K, Leem J, Baker T, Fuchs A, Georges G, Shi J, Deane CM (2013). "SAbDab: the structural antibody database". Nucleic Acids Research. **42** (D1): D1140–D1146. doi:10.1093/nar/gkt1043.
18. ^Kong X, Huang W, Liu Y (2023). "Conditional Antibody Design as 3D Equivariant Graph Translation". arXiv. arXiv:2208.06073 [q-bio.BM].
19. ^Chowdhury R, Bouatta N, Biswas S, Rochereau C, Church GM, Sorger PK, AlQuraishi M. "Single-sequence protein structure prediction using language models from deep learning." bioRxiv. 2021. doi:10.1101/2021.08.02.454840. Available from: https://www.biorxiv.org/content/early/2021/08/04/2021.08.02.454840.
20. a, b, c, d, eWallace B, Dang M, Rafailov R, Zhou L, Lou A, Purushwalkam S, Ermon S, Xiong C, Joty S, Naik N (2023). "Diffusion Model Alignment Using Direct Preference Optimization". arXiv. **cs.CV**. Available from: https://arxiv.org/abs/2311.12908.
21. ^Alley EC, Khimulya G, Biswas S, AlQuraishi M, Church GM (2019). "Unified rational protein engineering with sequence-based deep representation learning". Nature methods. **16** (12): 1315–1322.

22. ^Saka K, Kakuzaki T, Metsugi S, Kashiwagi D, Yoshida K, Wada M, Tsunoda H, Teramoto R (2021). "Antibody design using LSTM based deep generative model from phage display library for affinity maturation". Scientific reports. **11** (1): 5852.
23. ^Shin JE, Riesselman AJ, Kollasch AW, McMahon C, Simon E, Sander C, Manglik A, Kruse AC, Marks DS (2021). "Protein design and variant prediction using autoregressive generative models". Nature communications. **12** (1): 2403.
24. ^Akbar R, Robert PA, Weber CR, Widrich M, Frank R, Pavlović M, Scheffer L, Chernigovskaya M, Snapkov I, Slabodkin A, et al. In silico proof of principle of machine learning-based antibody design at unconstrained scale. MAbs. 2022; **14**(1): 2031482. Taylor & Francis.
25. a, b, cKong X, Huang W, Liu Y (2022). "Conditional antibody design as 3d equivariant graph translation". arXiv preprint arXiv:2208.06073. Available from: https://arxiv.org/abs/2208.06073.
26. a, bKong X, Huang W, Liu Y (2023). "End-to-end full-atom antibody design". arXiv preprint arXiv:2302.00203. Available from: https://arxiv.org/abs/2302.00203.
27. ^Weng L. "What are diffusion models?" lilianweng.github.io. Jul 2021. Available from: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/.
28. ^Luo C (2022). "Understanding diffusion models: A unified perspective". arXiv preprint arXiv:2208.11970. Available from: https://arxiv.org/abs/2208.11970.
29. ^Chan SH (2024). "Tutorial on Diffusion Models for Imaging and Vision". arXiv. arXiv:2403.18103.
30. a, bHo J, Jain A, Abbeel P (2020). "Denoising diffusion probabilistic models". Advances in neural information processing systems. **33**: 6840–6851.
31. ^Song J, Meng C, Ermon S. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502. 2020.
32. ^Hyvärinen A, Dayan P (2005). "Estimation of non-normalized statistical models by score matching." Journal of Machine Learning Research. **6** (4).
33. ^Vincent P (2011). "A connection between score matching and denoising autoencoders". Neural computation. **23** (7): 1661–1674.
34. ^Song Y, Garg S, Shi J, Ermon S. "Sliced score matching: A scalable approach to density and score estimation." In: Uncertainty in Artificial Intelligence. PMLR; 2020. p. 574-584.
35. ^Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, Poole B (2021). "Score-Based Generative Modeling through Stochastic Differential Equations". arXiv. arXiv:2011.13456.
36. ^Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C, Mishkin P, Zhang C, Agarwal S, Slama K, Ray A, et al. Training language models to follow instructions with human feedback. Advances in neural information processing systems. **35**:27730–27744, 2022.
37. ^Rafailov R, Sharma A, Mitchell E, Manning CD, Ermon S, Finn C (2024). "Direct preference optimization: Your language model is secretly a reward model". Advances in Neural Information Processing Systems. **36**.
38. ^Black K, Janner M, Du Y, Kostrikov I, Levine S (2023). "Training diffusion models with reinforcement learning". arXiv preprint arXiv:2305.13301.
39. ^Fan Y, Watkins O, Du Y, Liu H, Ryu M, Boutilier C, Abbeel P, Ghavamzadeh M, Lee K, Lee K (2024). "Reinforcement learning for fine-tuning text-to-image diffusion models". Advances in Neural Information Processing Systems. **36**.
40. ^Zhou X, Wang L, Zhou Y (2024). "Stabilizing Policy Gradients for Stochastic Differential Equations via Consistency with Perturbation Process". arXiv preprint arXiv:2403.04154. Available from: https://arxiv.org/abs/2403.04154.
41. ^Meng Y, Xia M, Chen D (2024). "Simpo: Simple preference optimization with a reference-free reward". arXiv preprint arXiv:2405.14734. Available from: https://arxiv.org/abs/2405.14734.
42. a, b, cRafailov R, Sharma A, Mitchell E, Ermon S, Manning CD, Finn C (2023). "Direct Preference Optimization: Your Language Model is Secretly a Reward Model". arXiv. arXiv:2305.18290 [cs.LG].
43. ^Bradley RA, Terry ME (1952). "Rank analysis of incomplete block designs: I. the method of paired comparisons". Biometrika. **39** (3/4): 324–345. Available from: http://www.jstor.org/stable/2334029.
44. a, b, cAdolf-Bryfogle J, Kalyuzhniy O, Kubitz M, Weitzner BD, Hu X, Adachi Y, Schief WR, Dunbrack RL. RosettaAntibodyDesign (RAbD): A general framework for computational antibody design. PLoS Computational Biology. **14**; 2017. S2CID 13658767.
45. ^Zhou Z, Liu J, Yang C, Shao J, Liu Y, Yue X, Ouyang W, Qiao Y (2023). "Beyond One-Preference-Fits-All Alignment: Multi-Objective Direct Preference Optimization". arXiv. arXiv:2310.03708.
46. ^Bayat R (2023). "A Study on Sample Diversity in Generative Models: GANs vs. Diffusion Models". Available from: https://openreview.net/forum?id=BQpCuJoMykZ.
47. ^Devlin J, Chang MW, Lee K, Toutanova K (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". arXiv. arXiv:1810.04805.
48. a, b, c, dChaudhury S, Lyskov S, Gray JJ. "PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta". Bioinformatics. **26** (5): 689–691, 01 2010. doi:10.1093/bioinformatics/btq007.

49. ^Olsen TH, Boyles F, Deane CM (2022). "Observed Antibody Space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences". Protein Science. **31** (1): 141–146. doi:10.1002/pro.4205. https://onlinelibrary.wiley.com/doi/abs/10.1002/pro.4205.
50. ^Dunbar J, Krawczyk K, Leem J, Baker T, Fuchs A, Georges G, Shi J, Deane CM (2014). "SAbDab: the structural antibody database". Nucleic acids research. **42** (D1): D1140--D1146.
51. ^Chothia C, Lesk AM (1987). "Canonical structures for the hypervariable regions of immunoglobulins". Journal of Molecular Biology. **196** (4): 901–917. doi:10.1016/0022-2836(87)90412-8. Link.
52. ^Steinegger M, S\u00f6ding J (2017). "MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets". Nature biotechnology. **35** (11): 1026--1028.
53. a, bKingma DP, Ba J (2014). "Adam: A method for stochastic optimization". arXiv preprint arXiv:1412.6980. Available from: https://arxiv.org/abs/1412.6980.

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.