AbFlowNet: Optimizing Antibody-Antigen Binding Energy via Diffusion-GFlowNet Fusion

Abrar Rahman Abir [1] * Haz Sameen Shahgir [2] * Md Rownok Zahan Ratul [3] Md Toki Tahmid [4] Greg Ver Steeg [2] Yue Dong [2]

Abstract

Complementarity Determining Regions (CDRs) are critical segments of an antibody that facilitate binding to specific antigens. Current computational methods for CDR design utilize reconstruction losses and do not jointly optimize binding energy, a crucial metric for antibody efficacy. Rather, binding energy optimization is done through computationally expensive Online Reinforcement Learning (RL) pipelines rely heavily on unreliable binding energy estimators. In this paper, we propose AbFlowNet, a novel generative framework that integrates GFlowNet with Diffusion models. By framing each diffusion step as a state in the GFlowNet framework, AbFlowNet jointly optimizes standard diffusion losses and binding energy by directly incorporating energy signals into the training process, thereby unifying diffusion and reward optimization in a single procedure. Experimental results show that AbFlowNet outperforms the base diffusion model by 3.06% in amino acid recovery, 20.40% in geometric reconstruction (RMSD), and 3.60% in binding energy improvement ratio. ABFlowNet also decreases Top-1 total energy and binding energy errors by 24.8% and 38.1% without pseudo-labeling the test dataset or using computationally expensive online RL regimes. [1]

1. Introduction

Antibodies are essential molecules of the adaptive immune system, with their complementarity-determining regions (CDRs) serving as the primary determinants of antigen recognition and binding specificity. Compared to the rest of the antibody, CDRs exhibit remarkable variability, enabling the immune system to recognize diverse antigens (Polonelli et al., 2008). CDRs are crucial for therapeutic antibody development, particularly in *humanization* where CDRs from non-human antibodies are transferred onto human antibodies to help it target new antigens, using techniques like CDR grafting (Jones et al., 1986) and shuffling (Jirholt et al., 1998). Daclizumab, the first FDA-approved humanized drug, was developed in 1997 by humanizing a mouse antibody to treat multiple sclerosis (Tsurushita et al., 2005). Since then, thousands of other drugs have been developed using CDR-based antibody modifications (Lu et al., 2020).

While transferring known animal CDRs has proven effective, there has been immense research interest into designing *de novo* CDRs to target novel antigens (Tang et al., 2024) and neoantigens (Zhang et al., 2021). The computational (*in silico*) design of CDRs presents a significant challenge due to the vast search space - a CDR sequence with L amino acids has 20^L possible combinations, not accounting for structural variations. Traditional Monte Carlo search-based approaches use biophysical energy functions (Adolf-Bryfogle et al., 2018a; Lapidoth et al., 2015; Adolf-Bryfogle et al., 2018b) to guide the search process but are computationally intensive and often get trapped in local optima (Luo et al., 2022; Jin et al., 2022). Deep Learning (DL) approaches using either Graph Neural Networks (Gao et al., 2023; Jin et al., 2022; Kong et al., 2023; 2022) or Diffusion models (Luo et al., 2022; Peng et al., 2023; Zhu et al., 2024) can learn the distribution of existing CDRs and sample new ones. However, unlike search-based approaches, DL methods do not explicitly optimize biophysical energy functions which has led to new research on online Reinforcement Learning (RL) (Sutton et al., 1998) post-training using these functions as rewards (Zhou et al., 2024; Ren et al., 2025; Wen et al., 2024). However these online RL regimes are extremely computationally expensive and highly dependent on the programs used to estimate said energy functions which are not always reliable (Vreven et al., 2012). Some approaches (Zhou et al., 2024; Wen et al., 2024) use the test dataset itself during RL which raises concerns about dataset bias

* Equal contribution [1] Bangladesh University of Engineering and Technology [2] University of California Riverside [3] University of Southern California [4] Princeton University. Correspondence to: Haz Sameen Shahgir <hshah057@ucr.edu>.

Proceedings of the Workshop on Generative AI for Biology at the 42^{nd} *International Conference on Machine Learning*, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

[1] The code and model weights are available at https://github.com/Patchwork53/abflownet.

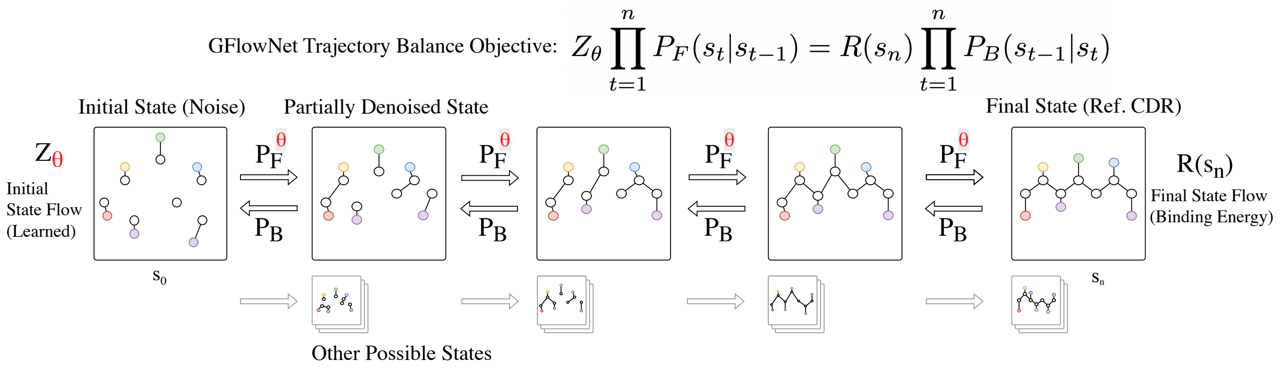

GFlowNet Trajectory Balance Objective: $Z_\theta \prod_{t=1}^{n} P_F(s_t|s_{t-1}) = R(s_n) \prod_{t=1}^{n} P_B(s_{t-1}|s_t)$

Figure 1. AbFlowNet reframes the diffusion process as a GFlowNet where each partially denoised CDR is a *state* and the transition probabilities are *flows through edges*. The initial state's flow is learned and the final state's flow is the binding energy of the reference CDR. To train, we simply enforce forward and backward flow parity, in addition to the diffusion losses.

(Deng et al., 2023). Furthermore, Zhou et al. (2024) has shown that RL improves binding energy but reduces two key structural metrics: Amino Acid Recovery and Root Mean Square Deviation with respect to the reference CDR.

Given the limitations of RL, GFlowNets have emerged as a promising alternative for optimizing Image Diffusion models (Venkatraman et al., 2024; Zhang et al., 2024a;b). In this work, we introduce AbFlowNet to address these concerns in biophysical energy optimization. As shown in Figure 1, we reframe the denoising diffusion process (Ho et al., 2020; Leach et al., 2022) in a GFlowNet (Bengio et al., 2021) framework where each partially denoised structure is a GFlowNet *state* and the forward and backwards transition probabilities are *flows through edges*. The *flow of a state* is the sum of all flow's of all trajectories through that state. The final fully-denoised state's flow is the binding energy *reward*. We use the Trajectory Balance objective (Bengio et al., 2021) to enforce that the forward and backward flow for a trajectory (a full denoising sequence starting from random noise and ending at a CDR structure) must be equal. As a result, the diffusion model implicitly learns better state transitions that lead to higher rewards.

Practically, AbFlowNet can be implemented by adding a single learned parameter and adding a TB loss term to the original loss terms. AbFlowNet shows convincing improvements over the base diffusion model, DiffAb (Luo et al., 2022), for the same number of gradient updates. Concretely, when averaged over all six CDR regions, AbFlowNet improves amino acid recovery (AAR) by 3.06%, root mean square deviation (RMSD) by 20.40% and samples 3.60% more CDRs that have better binding energy than the reference CDR. AbFlowNet also improves over DiffAb in Top-1 total energy and binding energy by 24.8% and 38.1%. Unlike online RL approaches such as AbDPO (Zhou et al., 2024), AbFlowNet is orders-of-magnitude less expensive, does not need repeated use of unreliable energy estimators and does not rely on pseudo-labeling the test set. Our key contributions are:

1. We present AbFlowNet, the first application of the GFlowNet framework for direct binding energy optimization in *de novo* diffusion-based CDR design. AbFlowNet improves over the base diffusion model in all metrics.
2. AbFlowNet is competitive with RL-based methods (Zhou et al., 2024) without using the test set complexes to generate synthetic CDR data for training, thereby mitigating data leakage concerns.
3. Unlike existing RL-based approaches (Zhou et al., 2024; Wen et al., 2024) which reduce AAR and RMSD, AbFlowNet improves AAR by $+3.06\%$ and RMSD by $+20.40\%$.

2. Related Works

Computational CDR Design Classical approaches to CDR design, such as RAbD (Adolf-Bryfogle et al., 2018a) and AbDesign (Lapidoth et al., 2015), rely on Monte Carlo algorithms that sample and optimize antibody structures based on biophysical energy functions. These methods, while effective in certain contexts, are computationally expensive and often get trapped in local optima due to the rugged energy landscape (Luo et al., 2022; Kong et al., 2023). In recent years, deep learning methods have emerged as promising alternatives. Notable Graph Neural Network-based models include HERN (Jin et al., 2022), MEAN (Kong et al., 2022), AbGNN (Gao et al., 2023), and dyMEAN (Kong et al., 2023). This methods have shown high AAR and RMSD but are limited in generation diversity due to their GNN structure. Diffusion-based models can generate multiple CDR given a complex which can be later ranked heuristically. Notable Diffusion-based models include AbDiffuser (Martinkus et al., 2024), DiffAb (Luo et al., 2022), AbDesign (Peng et al., 2023), RFanti-

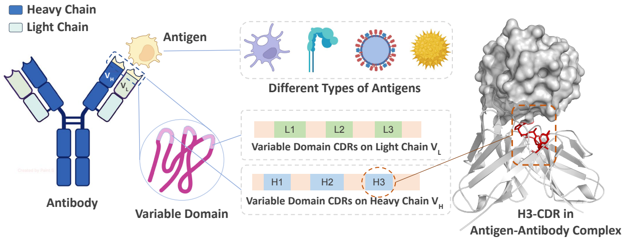

Heavy Chain
Light Chain
Antigen
V_H
V_L
Different Types of Antigens
L1
L2
L3
Variable Domain CDRs on Light Chain V_L
H1
H2
H3
Variable Domain CDRs on Heavy Chain V_H
Antibody
Variable Domain
H3-CDR in
Antigen-Antibody Complex

Figure 2. Antibody-antigen complex.

body (Bennett et al., 2025), and AbX (Zhu et al., 2024). AbX is the current state-of-the-art CDR design model and uses a large Protein Language Model (Lin et al., 2022) to enforce evolutionary plausibility of the generated CDRs.

Binding Energy Optimization For CDR Design A key metric for CDR effectiveness is binding affinity. One commonly used energy metric is binding energy ΔG. Since binding energy is a singular value for the entire complex, it is a sparse training signal which is often optimized via Reinforcement Learning (Sutton et al., 1998). AbDPO (Zhou et al., 2024) post-trains a base DiffAb model by repeatedly sampling new CDRs, ranking them based on binding affinity, determined with `Rossetta InterfaceAnalyzer` (Chaudhury et al., 2010) and using DPO (Rafailov et al., 2024). However, this RL training phase significantly lowers AAR and RMSD compared to the base method. AlignAb (Wen et al., 2024) points out that there are multiple valid energy-based rewards and finetune separate models for each reward using DPO. AbNovo (Ren et al., 2025) follows the approach of AbDPO with AbX instead of DiffAb as the base model and used Noise Contrastive Alignment (NCA) (Chen et al., 2024) as the RL objective instead of DPO.

One notable weakness of all online RL methods is the need to compute binding energy for newly designed CDRs. *In silico* methods such as `Rosetta` (Chaudhury et al., 2010; Adolf-Bryfogle et al., 2018a) or `OpenMM Yank` (Eastman et al., 2017; Rizzi et al., 2020) have only moderate corelation with the real binding energy(Vreven et al., 2012). Furthermore, the generated CDRs are not guaranteed to be geometrically plausible which might reduce the reliability of energy estimators further. In contrast, AbFlowNet does not require computing the energy of newly generated CDRs and can, in principle, be trained solely on *in vitro* affinity data of CDRs in the training set.

3. Background

3.1. Antibody-Antigen Complex

As shown in Figure 2, antibodies are composed of two heavy chains and two light chains. Each chain consists of a variable region and a constant region. The variable regions of both the heavy chain ($\mathrm{V_H}$) and the light chain ($\mathrm{V_L}$) contain three complementarity determining regions (CDRs): CDR1, CDR2, and CDR3, making a total of six CDRs per antibody. These regions are highly diverse due to genetic recombination and somatic hypermutation, allowing antibodies to recognize a vast array of antigens. The CDRs form a binding site that is complementary in shape and chemical properties to the antigen's binding site (epitope). In Figure 2, the third CDR in the heavy chain (CDR-H3) is highlighted due to its critical in determining the binding affinity to the antigen. The sequence and structure of CDRs vary widely among antibodies, enabling the immune system to recognize and respond to a wide range of antigens. CDRs interact with the antigen through non-covalent bonds (e.g., hydrogen bonds, electrostatic interactions, van der Waals forces) (Polonelli et al., 2008). In our work, we aim to design the sequences and structures of the CDR regions (often referred to as the framework regions), conditioned on the non-CDR regions of the antibody and on the target antigen.

3.2. GFlowNet

GFlowNets are generative models that learn to sample from a desired distribution by modeling flows on a directed acyclic graph (DAG) (Bengio et al., 2021). Given a DAG $G = (\mathcal{S}, \mathcal{A})$ with state space $\mathcal{S}$ and action space $\mathcal{A}$, and a positive reward function $R : \mathcal{X} \to \mathbb{R}_{\geq 0}$ defined on terminal states $\mathcal{X}$, a GFlowNet learns a policy that generates trajectories terminating at states with probability proportional to their rewards. Formally, a GFlowNet defines a flow F on trajectories $\tau = (s_0 \to s_1 \to \ldots \to s_n)$ from the initial state s_0 to terminal states. The ***state flow*** of state s is defined as $F(s) = \sum_{\tau=(\ldots\to s\ldots)} F(\tau)$ and the ***edge flow*** between states s and s' is defined as $F(s \to s') = \sum_{\tau=(\ldots\to s\to s'\ldots)} F(\tau)$. The following *flow matching* constraint (incoming flow = outgoing flow) is satisfied for all non-boundary states $F(s) = \sum_{(s''\to s)\in\mathcal{A}} F(s'' \to s) = \sum_{(s\to s')\in\mathcal{A}} F(s \to s')$. For terminal states s_n, the flow is the non-negative reward: $F(s_n) = R(s_n)$.

Flows also induce forward and backward transition policies: $P_F(s'|s) = \frac{F(s\to s')}{F(s)}$ and $P_B(s|s') = \frac{F(s\to s')}{F(s')}$. A GFlowNet aims to learn policies such that the terminal state flows match their rewards.

Trajectory Balance Trajectory Balance (TB) (Malkin et al., 2022) provides an elegant training objective for GFlowNets that enforces consistency between forward generation and backward reconstruction across entire trajectories. For any complete trajectory $\tau = (s_0 \to s_1 \to \ldots \to s_n)$ terminating at state x, TB enforces the constraint:

$$F(s_0)\prod_{t=1}^{n} P_F(s_t|s_{t-1}) = F(s_n)\prod_{t=1}^{n} P_B(s_{t-1}|s_t) \qquad (1)$$

Here, the flow of the terminal state s_n is $F(s_n) = R(s_n)$. The flow of the initial state $F(s_0)$ is the sum of the flow over all trajectories which is not tractable. Therefore, the authors of TB propose approximating $F(s_0)$ with Z_θ where θ is a neural network. This yields the final constraint:

$$Z_\theta \prod_{t=1}^{n} P_F(s_t|s_{t-1}) = R(s_n) \prod_{t=1}^{n} P_B(s_{t-1}|s_t) \tag{2}$$

4. Methodology

In Section 4.1, we discuss the training objectives of the base diffusion model, and in Section 4.2, we describe our reframing of the diffusion process as a GFlowNet. Equation 18 presents our final training objective, which jointly optimizes the diffusion losses and the binding energy.

4.1. Denoising Objective

We train a diffusion probabilistic model parameterized by a neural network for CDR design. Following Luo et al. (2022), we condition on the antigen structure and the antibody framework to generate CDR. The model is trained using the standard denoising objective across three protein properties: **amino acid type** $d_i \in \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, W, Y, V\}$, **3D coordinate** $x_i \in \mathbb{R}^3$, and **3D orientation** $O_i \in SO(3)$ where $SO(3)$ is the Lie group of 3D rotations.

Assume the CDR to be generated has m amino acids with index from $l+1$ to $l+m$. They are denoted as $S^t = \{s_j^t | j = l+1, \ldots, l+m\}$ where $s_j^t = (d_j^t, x_j^t, O_j^t)$. Our goal is to model the distribution of S^0 given the structure of the antibody-antigen complex $C = \{(d_i, x_i, O_i) \mid i \in \{1, \ldots, N\} \setminus \{l+1, \ldots, l+m\}\}$.

Multinomial Diffusion for Amino Acid Types The forward diffusion process for amino acid types is based on the multinomial diffusion process (Hoogeboom et al., 2021):

$$q(d_j^t|d_j^{t-1}) = \text{Mul}\left((1-\beta_{\text{type}}^t) \cdot \text{oh}(d_j^{t-1}) + \frac{\beta_{\text{type}}^t}{20} \cdot \mathbf{1}\right) \tag{3}$$

$$p(d_j^{t-1}|S^t, C) = \text{Mul}\left(F_\theta(S^t, C)[j]\right) \tag{4}$$

where p is the forward diffusion process, q is the backward denoising process, Mul() is the Multinomial function and oh() is the one-hot function. β_{type}^t is the probability of resampling another amino acid uniformly over the 20 types and $F_\theta(\cdot)[j]$ is a neural network model that predicts the probability of the amino acid type for the j-th amino acid on the CDR. The training objective is to minimize the expected KL divergence between the posterior distribution q and the predicted distribution p:

$$L_{\text{type}}^t = \mathbb{E}\left[\frac{1}{m}\sum_j D_{\text{KL}}\left(q(d_j^{t-1}|d_j^t, d_j^0) \| p(d_j^{t-1}|S^t, C)\right)\right] \tag{5}$$

Diffusion for 3D Coordinates The forward and backward diffusion processes for the coordinate x_j are defined as:

$$q(x_j^t|x_j^{t-1}) = \mathcal{N}\left(x_j^t \middle| \sqrt{1-\beta_{\text{pos}}^t} \cdot x_j^{t-1}, \beta_{\text{pos}}^t I\right) \tag{6}$$

$$p(x_j^{t-1}|S^t, C) = \mathcal{N}\left(x_j^{t-1} \middle| \mu_p(S^t, C), \beta_{\text{pos}}^t I\right) \tag{7}$$

$$\mu_p(S^t, C) = \frac{1}{\sqrt{1-\beta_{\text{pos}}^t}}\left(x_j^t - \frac{\beta_{\text{pos}}^t}{\sqrt{1-\tilde{\alpha}_{\text{pos}}^t}} G_\theta(S^t, C)[j]\right) \tag{8}$$

where β_{pos}^t controls the rate of diffusion in q. The denoising diffusion process p uses the reparameterization trick (Ho et al., 2020) and $G_\theta(\cdot)[j]$ is a neural network that predicts the standard Gaussian noise $\epsilon_j \sim \mathcal{N}(0, I)$ added instead of predicting x_j^{t-1} directly where $\tilde{\alpha}_{\text{pos}}^t = \prod_{t=1}^{T}(1-\beta_{\text{pos}}^t)$.

The training objective is to minimize the expected MSE between G and ϵ:

$$L_{\text{pos}}^t = \mathbb{E}\left[\frac{1}{m}\sum_j \|\epsilon_j - G(S^t, C)[j]\|^2\right] \tag{9}$$

SO(3) Denoising for Amino Acid Orientations Following Leach et al. (2022); Luo et al. (2022), the denoising process for orientation directly attempts to predict the final orientation O_j^0 from O_j^t. The transitions are defined as:

$$q(O_j^t|O_j^0) = \mathcal{IG}_{SO(3)}\Big(O_j^t|\lambda\big(\sqrt{\bar{\alpha}_{ori}^t}, O_j^0\big),\ 1-\bar{\alpha}_{ori}^t\Big) \tag{10}$$

$$p(O_j^{t-1}|S^t, C) = \mathcal{IG}_{\text{SO(3)}}\left(O_j^{t-1} \middle| H_\theta(S^t, C)[j], \beta_{\text{ori}}^t\right) \tag{11}$$

where $\bar{\alpha}_{ori}^t = \prod_{\tau=1}^{t}(1-\beta_{ori}^\tau)$ with β_{ori}^t being the variance increased with step t, $\mathcal{IG}_{\text{SO(3)}}$ denotes the isotropic Gaussian distribution on $SO(3)$ (Leach et al., 2022) and $\lambda(\gamma, x) = \exp\big(\gamma \log(x)\big)$ is the geodesic interpolation (or "scaling") of the rotation $x \in SO(3)$ from the identity. $H_\theta(\cdot)[j]$ is a neural network that denoises the orientation and outputs the denoised orientation matrix. The training objective simply minimizes the difference between the real and predicted orientation matrices:

$$L_{\text{ori}}^t = \mathbb{E}\left[\frac{1}{m}\sum_j \|(O_j^0)^\top \tilde{O}_j^{t-1} - I\|^2\right] \tag{12}$$

4.2. Trajectory Balance Objective

In addition to the denoising objectives, we aim to optimize the binding energy of the generated CDR with respect to the antigen and antibody. However, binding energy can only be computed for the final CDR after a complete denoising process, making it a sparse reward. To address this, we use the **Trajectory Balance (TB)** objective (Malkin et al., 2022) which propagates rewards back through the diffusion trajectory by enforcing global flow-matching constraints.

We begin by defining a GFlowNet state as the partially denoised CDR at timestep t. A CDR is composed of a sequence of amino acids and the transition probability for each amino acid location j is the product of the three independent denoising processes (Equations 13 and 14). We define the GFlowNet edge flow as the transition probability of the entire CDR, which is simply the product of the probability of each location (Equations 15 and 16):

$$q(s_j^t|s_j^{t-1}) = q(d_j^t|d_j^{t-1}) \cdot q(x_j^t|x_j^{t-1}) \cdot q(O_j^t|O_j^{t-1}) \quad (13)$$

$$\begin{aligned} &p(s_j^{t-1} \mid s_j^t) = p(s_j^{t-1} \mid S^t, C) \\ &= p(d_j^{t-1} \mid S^t, C) \cdot p(x_j^{t-1} \mid S^t, C) \cdot p(O_j^{t-1} \mid S^t, C) \end{aligned} \quad (14)$$

$$q(S^t|S^{t-1}) = \prod_{j=l}^{l+m} q(s_j^t|s_j^{t-1}) \quad (15)$$

$$p(S^{t-1}|S^t) = \prod_{j=l}^{l+m} p(s_j^t|s_j^{t-1}) \quad (16)$$

For each data point during a mini-batch update, we uniformly sample a timestep t to compute L_{type}^t, L_{pos}^t and L_{ori}^t. However, we require a complete trajectory to enforce TB. Therefore, we compute all forward $q(S_{t-1}|S_t)$ and backward probabilities $p(S_t|S_{t-1};\theta)$ for t in $(0,T)$. Following Kim et al. (2024), we precompute the reward $\mathcal{R}(S^0) = \exp(-\alpha \cdot \text{BindingEnergy}(S^0))$ for each CDR S^0 in the training dataset and enforce the TB objective:

$$L_{\text{TB}} = \left(\log \frac{Z_\theta \prod_{t=0}^{T} p(S^t|S^{t-1};\theta)}{\mathcal{R}(S^0) \prod_{t=0}^{T} q(S^{t-1}|S^t)} \right)^2 \quad (17)$$

where Z_θ is the estimated initial state flow and $R(x)$ is the binding energy reward. Therefore, the overall training objective combines the denoising losses and the TB objective:

$$L = \mathbb{E}_{t\sim\text{Uniform}(1,\ldots,T)} \left[L_{\text{type}}^t + L_{\text{pos}}^t + L_{\text{ori}}^t \right] + w \cdot L_{\text{TB}} \quad (18)$$

where w is a scaling factor for balancing the denoising objectives computed per diffusion step and the TB objective calculated over the entire trajectory.

4.3. Sampling Algorithm

For sequence-structure co-design, we construct S^T by sampling amino acid types for each position from the uniform distribution $d_j^T \sim \text{Uniform}(20)$, CDR positions from the standard normal distribution: $x_j^T \sim \mathcal{N}(0, I_3)$, and orientations from the uniform distribution over SO(3): $O_j^T \sim \text{Uniform(SO(3))}$. AbFlowNet iteratively denoises the sequence and structures following the standard diffusion process until $t = 0$. Upon generating the amino acid sequence and the structure of the backbone, we optimize the side-chain angles using `PackRotamersMover` in `PyRosetta` (Chaudhury et al., 2010).

Crucially, AbFlowNet applies the GFlowNet balance objective solely during training, not during inference. This approach enables AbFlowNet to operate as a standard diffusion model at sampling time, without requiring energy calculations via the `Rosetta InterfaceAnalyzer`.

5. Experiments

Dataset Curation We use the Structural Antibody Database (SAbDab) (Dunbar et al., 2014) as the training dataset. We first remove structures whose resolution is less than 4Å and discard antibodies targeting non-protein antigens (Luo et al., 2022). We cluster antibodies in the database according to CDR-H3 sequences at 50% sequence identity using MMSeq2 (Steinegger & Söding, 2017). Our final training dataset contains 9410 antigen-antibody complexes. We evaluate sequence-structure codesign on RAbD test dataset, consisting of 60 diverse antibody-antigen complexes (Adolf-Bryfogle et al., 2018b). We also evaluate on the test set proposed by DiffAb (Luo et al., 2022) which contains 19 complexes with antigens from several well-known pathogens including SARS-CoV-2, MERS, influenza, and so on. For both test sets, we strictly remove the overlap between the training set and the testing sets using a CDR-H3 sequence identity threshold of 50%.

Metrics We use standard metrics to evaluate designed antibodies (Adolf-Bryfogle et al., 2018b; Luo et al., 2022; Zhu et al., 2024), namely, (1) **AAR**: the amino acid recovery rate measured by the sequence identity between the reference CDR sequences and the generated sequences, (2) **RMSD**: the C_α root-mean-square deviation (RMSD) between the generated structure and the original structure, and (3) **IMP**: the percentage of designed CDRs with lower (better) binding energy (ΔG) than the original CDR. The binding energy is calculated by `InterfaceAnalyzer` in the Rosetta software package (Adolf-Bryfogle et al., 2018a; Chaudhury

Table 1. **Top-1** CDR E_{total} and CDR-Ag ΔG (kcal/mol) for *de novo* CDR-H3 design. (↓) indicates lower is better. We also show the percentage reduction over DiffAb with the same sampling budget.

Methods	**Sampling**	**CDR E_{total} (↓)**	**CDR-Ag ΔG (↓)**	**Test Set Used**	**Code**
	Budget	(with Δ)	(with Δ)	**in Sampling**	**Availability**
Reference		4.52	-13.72	—	—
GNN Baselines					
HERN (Jin et al., 2022)	2,528	7594.94	1159.34	—	Yes
MEAN (Kong et al., 2022)	2,528	3113.70	114.98	—	Yes
dyMEAN^1 (Kong et al., 2023)	2,528	15025.67	2391.00	—	Yes
dyMEAN^2	2,528	3234.30	1619.24	—	Yes
Diffusion-Based					
DiffAb (Luo et al., 2022)	2,528	211.00	9.54	—	Yes
AbDPO (Zhou et al., 2024)†	2,528	162.75 (↓23.4%)	**-4.85 (↓61.9%)**	Yes	No
DiffAb	100	480.25	11.20	—	Yes
AbFlowNet (Ours)	100	**362.03 (↓24.8%)**	1.71 (↓38.1%)	**No**	**Yes**

†AbDPO is not open-sourced and cannot be independently reproduced. Results are shown for reference only. Improvements for AbDPO and AbFlowNet are relative to DiffAb under the same sampling budget.

et al., 2010). Diffusion models are capable of generating diverse data points from the target distribution by randomly sampling from the initial distribution (Ho et al., 2020; Leach et al., 2022). This is especially advantageous in CDR design where we can generate multiple candidates CDRs *in silico* and select only the most promising CDR for *in vitro* validation according to some desirable property. To this end, the authors of AbDPO (Zhou et al., 2024) proposed: (1) **Top-1 CDR $\mathbf{E_{total}}$**: total energy of the whole designed CDR (kcal/mol) of the *best CDR* out of N; (2) **Top-1 CDR-Ag $\mathbf{\Delta G}$** : the difference in total energy between the bound state and the unbound state of that CDR and antigen. Following AbDPO, we generate N CDRs for each antigen-antibody complex in the RAbD test dataset and choose the best CDR ranked by $\mathbf{E_{total}} + \mathbf{\Delta G}$.

Model Architecture and Hyperparameters We use the transformer-based parametrization defined in Luo et al. (2022) to encode antigen-antibody complex C and conditionally generate d_j^t, x_j^t and O_j^t. We add a learnable parameter to predict a Z_θ which is learned solely through backpropagation since Z_θ global estimation of the initial state's flow independent of individual training samples. Following DiffAb (Luo et al., 2022), we train both DiffAb and AbFlowNet for $200,000$ steps using Adam optimizer (Kingma & Ba, 2014) with learning rate $1e-6$. For AbFlowNet, computing TB loss requires sampling full trajectories which is computationally expensive ($\sim$ 20 seconds per step). Therefore, we train first $195,000$ steps without TB loss and set TB loss weight $w = 5e-6$ for the final 5000 steps. We present details about parameter sweep over w in Appendix A.2 and discuss the effect of training longer in Appendix A.3.

Baselines Our primary point of comparison is AbDPO (Zhou et al., 2024), an online RL method that post-trains a DiffAb model to optimize CDR-H3 binding energy. AbDPO samples 10,122 CDRs per test complex to construct a preference dataset and updates the model via Direct Preference Optimization (DPO) (Rafailov et al., 2024). We also report results from several graph neural network (GNN) baselines — HERN (Jin et al., 2022), MEAN (Kong et al., 2022), and DyMEAN (Kong et al., 2023). Unlike the other baselines, which rely on side-chain packing algorithms to determine optimal side-chain orientations, DyMEAN jointly generates both the CDR backbone and side-chain orientations. We report two versions of DyMEAN: DyMEAN^1, which jointly generates side-chain orientations, and DyMEAN^2, which uses the generated backbone but packs side-chains with `PyRosetta PackRotamersMover`. Results for methods using sampling budget of $N = 2,528$ were taken from Zhou et al. (2024). We use $N = 100$ when evaluating AbFlowNet for computational efficiency.

6. Results

In Section 6.1, we show that AbFlowNet significantly improves Top-1 energy-based metrics and is comparable to AbDPO (Zhou et al., 2024), a far more computationally expensive method. In Section 6.2 we demonstrate that AbFlowNet's joint optimization improves upon the base diffusion model across all metrics for the same number of training steps. Finally, in Section 6.3, we highlight a qualitative example of the CDR-H3 designed for the `PDB 5MES` complex by DiffAb and AbFlowNet.

6.1. *De novo* CDR-H3 design using Top-1 Energies

The H3 region is especially difficult for all models to generate because the H3 loop undergoes independent mutation before joining the rest of the antibody sequence (Graves et al., 2020), introducing variability and significantly affecting the structure and function of the antibody.

Table 1 shows that **AbFlowNet is competitive with AbDPO without relying on test-set structures**. Specifically, AbFlowNet significantly outperforms DiffAb at $N = 100$, achieving performance gains comparable to those reported by AbDPO (Zhou et al., 2024), despite not using test-set complexes to generate preference datasets. To evaluate relative improvement, we apply the formula (METHOD - BASELINE) / (REFERENCE - BASELINE), which normalizes performance gains with respect to the baseline and reference. We further validate our findings on the DiffAb test benchmark (Luo et al., 2022), which includes 19 complexes with antigens from SARS-CoV-2, MERS, and influenza; full results are provided in Appendix Table 4.

Compared to AbDPO (Zhou et al., 2024), which is an online RL method that post-trains a DiffAb model to optimize CDR binding energy, AbFlowNet offers several advantages. AbDPO samples 10,122 CDRs per test complex to construct a preference dataset, and updates the model via Direct Preference Optimization (DPO) (Rafailov et al., 2024). This approach has two main limitations: (1) sampling at this scale and computing binding energies is computationally expensive, and (2) using test-set antibody–antigen complexes during RL introduces potential bias. In contrast, AbFlowNet relies solely on precomputed binding energies from the 9,410 training examples, eliminating the need for expensive sampling and reward computation during optimization.

Additionally, AbFlowNet maintains strong reconstruction performance, whereas DPO-based RL reduces it. AbDPO, which post-trains only on binding energy, does not preserve the training distribution as well and leads to a 9.96% reduction in average AAR and an increase in RMSD by 0.14 Å. By jointly optimizing for both reconstruction and binding, AbFlowNet achieves better AAR and RMSD scores (see Section 6.2, Table 2). Thus, AbFlowNet improves binding energy without sacrificing structural accuracy.

Finally, we note that both CDR E_{total} and CDR-Ag ΔG are *Top-1* metrics that select the best-scoring sample among N generated CDR-H3s. These metrics are inherently sensitive to the sampling budget. For example, DiffAb's E_{total} improves significantly from 480.25 kcal/mol at $N = 100$ to 211.00 kcal/mol at $N = 2,528$. While our evaluation of AbFlowNet uses a modest budget of 100 samples for efficiency, increasing N would likely yield even better results. This suggests that AbFlowNet's performance could scale further with additional samples—without relying on test-set structures or incurring the computational cost of online reward evaluation.

6.2. Joint Optimization Outperforms Diffusion Baseline

Table 2. Evaluation of the generated antibody CDRs (sequence-structure co-design) on the RAbD test dataset (60 sequences) using AAR (%), RMSD (Å) and IMP (%) metrics.

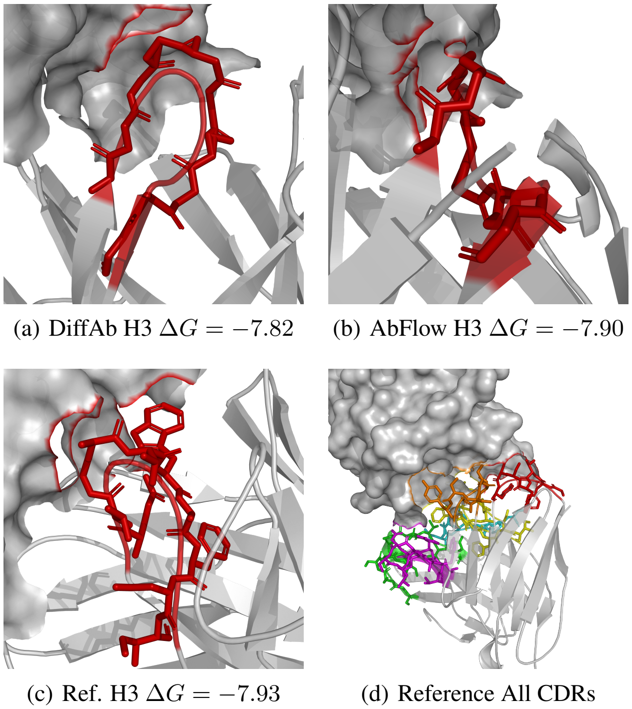
CDR	Method	AAR↑	RMSD↓	IMP↑
H1	Diffab	**64.23%**	1.153Å	69.27%
	AbFlowNet	63.49%	**0.974Å**	**73.66%**
H2	Diffab	35.87%	1.095Å	46.79%
	AbFlowNet	**38.06%**	**0.848Å**	**60.07%**
H3	Diffab	24.34%	3.236Å	**14.38%**
	AbFlowNet	**25.08%**	**3.194Å**	12.65%
L1	DiffAb	53.69%	1.153Å	55.51%
	AbFlowNet	**55.62%**	**0.974Å**	**56.83%**
L2	DiffAb	50.46%	0.795Å	68.78%
	AbFlowNet	**54.09%**	**0.782Å**	**70.64%**
L3	DiffAb	**44.87%**	3.840Å	**36.98%**
	AbFlowNet	44.68%	**1.310Å**	34.70%

Table 2 shows the performance of AbFlowNet and the baseline diffusion model DiffAb on the RaBD dataset. Both models were trained with identical hyperparameters and the same number of gradient updates; the only difference is that AbFlowNet incorporates the TB objective from Eq. 18.

AbFlowNet outperforms DiffAb in all three metrics: +3.06% in AAR, +20.40% in RMSD and +3.60% in IMP. For the CDR-L3 chain, in particular, the RMSD achieved by AbFlowNet is considerably lower than those of other methods. We find consistent improvements in most CDR regions when using the test set proposed by DiffAb (Luo et al., 2022), shown in Appendix 3.

6.3. Qualitative Example

Figure 3 (d) shows Protein Data Bank entry 5MES is a complex where a chimeric mouse-human Fab antibody fragment chaperon is bound to the Mcl-1 antigen. The antigen, a chimeric human/mouse Mcl-1 homolog is over-expressed in various tumors and prevents tumor cells from undergoing apoptosis. The Fab antibody serves to stabilize the complex, allowing researchers to resolve at at 2.24 Å resolution by X-ray diffraction (Johannes et al., 2017). Figure 3 (a) and (b) show that the CDR-H3 region designed by AbFlowNet establishes tighter adhesion between the antibody and antigen. However, *de novo* generation methods still underperform the reference H3, in which side chains contribute a significant



(a) DiffAb H3 $\Delta G = -7.82$

(b) AbFlow H3 $\Delta G = -7.90$

(c) Ref. H3 $\Delta G = -7.93$

(d) Reference All CDRs

Figure 3. De novo Generated and Reference CDR-H3s for 5MES complex. For DiffAb and AbFlowNet, we generated 100 CDRs and selected the one with the highest ΔG.

fraction of binding affinity. Indeed, CDR side chains may account for the majority of an antibody's binding affinity and specificity (Peng et al., 2014; Robin et al., 2014). Because diffusion-based generators produce only the backbone, we rely on a side-chain packing algorithm (e.g., PyRosetta PackRotamerMover) to place geometrically plausible side-chain orientations. This key limitation of diffusion models is discussed further in Appendix C.

7. Discussion

On the Use of Rosetta InterfaceAnalyzer AbFlowNet does not strictly require the use of the energy estimators such PyRosetta InterfaceAnalyzer (Chaudhury et al., 2010) and could in principle work with *only* the reliable binding-energy measurements obtained from *in vitro* experiments. This is particularly relevant because prior works have raised concerns about the accuracy of energy estimators (Vreven et al., 2012; Chaves et al., 2023; Conti et al., 2022).

In contrast to RL-based approaches such as AbDPO (Zhou et al., 2024), AlignAb (Wen et al., 2024) and AbNovo (Ren et al., 2025), which require energy estimates for hundreds of thousands of potentially implausible de novo CDRs, AbFlowNet needs energies only for existing reference CDRs. However, of the 9410 antigen–antibody complexes in the SAbDab database, only 736 have experimental affinity data available. As such, we used InterfaceAnalyzer to estimate the energies for our training set. We opted for InterfaceAnalyzer to maintain parity with existing baselines (Luo et al., 2022; Zhou et al., 2024; Wen et al., 2024).

Experimental affinities are typically measured using label-free biophysical techniques such as isothermal titration calorimetry (Boudker & Oh, 2015), surface plasmon resonance (Hearty et al., 2012; Murali et al., 2022) or bio-layer interferometry (Abdiche et al., 2008), each with its own advantages and trade-offs. GPU-accelerated programs such as OpenMM Yank (Rizzi et al., 2020) are more accurate but require hours to process a single complex. AbFlowNet makes it feasible to augment experimental data with GPU-accelerated simulations because these simulations need to be computed only once for authentic CDRs, rather than iteratively for synthetic CDRs.

Efficiency Analysis Computing the Trajectory Balance (TB) objective requires complete trajectories, i.e., sampling from the initial Gaussian state to the final denoised CDR conditioned on the antigen-antibody complex. Following (Luo et al., 2022; Zhou et al., 2024; Zhu et al., 2024), we use 100 denoising steps. The wall-clock time for 100 denoising steps with a batch size of 16 is $\sim$20 seconds which dominates the run-time of training AbFlowNet. Furthermore, we do backward propagation for only one random time step since storing the activations for all 100 steps is computationally infeasible. This deviates from the original TB (Bengio et al., 2021) formulation and hence is only as approximation. Following (Zhang et al., 2024b), we discuss our attempt at using an alternative GFlowNet optimization objective, Detailed Balance, in Appendix D.

8. Conclusion

We presented AbFlowNet, a novel framework integrating Diffusion Models and GFlowNets for antibody CDR design. AbFlowNet directly incorporates binding energy signals throughout training, jointly optimizing sequence/structure generation and binding affinity. This approach avoids the trade-offs seen in RL-based methods such as strong reliance on *in silico* binding energy estimation and usage of test set data. Experimentally, AbFlowNet outperforms its base diffusion model (DiffAb) in all metrics and is competitive with expensive RL approaches while only using precomputed rewards.

References

Abdiche, Y., Malashock, D., Pinkerton, A., and Pons, J. Determining kinetics and affinities of protein interactions using a parallel real-time label-free biosensor, the octet.

Analytical biochemistry, 377(2):209–217, 2008.

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

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

Bengio, E., Jain, M., Korablyov, M., Precup, D., and Bengio, Y. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

Bennett, N. R., Watson, J. L., Ragotte, R. J., Borst, A. J., See, D. L., Weidle, C., Biswas, R., Yu, Y., Shrock, E. L., Ault, R., et al. Atomically accurate de novo design of antibodies with rfdiffusion. *bioRxiv*, pp. 2024–03, 2025.

Boudker, O. and Oh, S. Isothermal titration calorimetry of ion-coupled membrane transporters. *Methods*, 76:171–182, 2015.

Chaudhury, S., Lyskov, S., and Gray, J. J. Pyrosetta: a script-based interface for implementing molecular modeling algorithms using rosetta. *Bioinformatics*, 26(5):689–691, 2010.

Chaves, E., Mhrous, E., Nascimento-Filho, M., Cruz, C., Ferraz, M., and Lins, R. Prediction of absolute protein-protein binding free energy by a super learner model. 2023.

Chen, H., He, G., Yuan, L., Cui, G., Su, H., and Zhu, J. Noise contrastive alignment of language models with explicit rewards. *arXiv preprint arXiv:2402.05369*, 2024.

Conti, S., Lau, E. Y., and Ovchinnikov, V. On the rapid calculation of binding affinities for antigen and antibody design and affinity maturation simulations. *Antibodies*, 11 (3), 2022. ISSN 2073-4468. doi: 10.3390/antib11030051. URL https://www.mdpi.com/2073-4468/11/3/51.

Deng, C., Zhao, Y., Tang, X., Gerstein, M., and Cohan, A. Investigating data contamination in modern benchmarks for large language models. *arXiv preprint arXiv:2311.09783*, 2023.

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

Eastman, P., Swails, J., Chodera, J. D., McGibbon, R. T., Zhao, Y., Beauchamp, K. A., Wang, L.-P., Simmonett, A. C., Harrigan, M. P., Stern, C. D., et al. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology*, 13(7): e1005659, 2017.

Gao, 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*, pp. 506–517, 2023.

Graves, J., Byerly, J., Priego, E., Makkapati, N., Parish, S. V., Medellin, B., and Berrondo, M. A review of deep learning methods for antibodies. *Antibodies*, 9(2):12, 2020.

Hearty, S., Leonard, P., and O'Kennedy, R. Measuring antibody–antigen binding kinetics using surface plasmon resonance. *Antibody Engineering: Methods and Protocols, Second Edition*, pp. 411–442, 2012.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

Jin, W., Barzilay, R., and Jaakkola, T. Antibody-antigen docking and design via hierarchical structure refinement. In *International Conference on Machine Learning*, pp. 10217–10227. PMLR, 2022.

Jirholt, P., Ohlin, M., Borrebaeck, C., and Söderlind, E. Exploiting sequence space: shuffling in vivo formed complementarity determining regions into a master framework. *Gene*, 215(2):471–476, 1998.

Johannes, J. W., Bates, S., Beigie, C., Belmonte, M. A., Breen, J., Cao, S., Centrella, P. A., Clark, M. A., Cuozzo, J. W., Dumelin, C. E., et al. Structure based design of non-natural peptidic macrocyclic mcl-1 inhibitors. *ACS medicinal chemistry letters*, 8(2):239–244, 2017.

Jones, P. T., Dear, P. H., Foote, J., Neuberger, M. S., and Winter, G. Replacing the complementarity-determining regions in a human antibody with those from a mouse. *Nature*, 321(6069):522–525, 1986.

Kim, M., Choi, S., Kim, H., Son, J., Park, J., and Bengio, Y. Ant colony sampling with gflownets for combinatorial optimization. *arXiv preprint arXiv:2403.07041*, 2024.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kong, X., Huang, W., and Liu, Y. Conditional antibody design as 3d equivariant graph translation. *arXiv preprint arXiv:2208.06073*, 2022.

Kong, X., Huang, W., and Liu, Y. End-to-end full-atom antibody design. *arXiv preprint arXiv:2302.00203*, 2023.

Lapidoth, G. D., Baran, D., Pszolla, G. M., Norn, C., Alon, A., Tyka, M. D., and Fleishman, S. J. Abdesign: An algorithm for combinatorial backbone design guided by natural conformations and sequences. *Proteins: Structure, Function, and Bioinformatics*, 83(8):1385–1406, 2015.

Leach, A., Schmon, S. M., Degiacomi, M. T., and Willcocks, C. G. Denoising diffusion probabilistic models on SO(3) for rotational alignment. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022. URL https://openreview.net/forum?id=BY88eBbkpe5.

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. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

Lu, R.-M., Hwang, Y.-C., Liu, I.-J., Lee, C.-C., Tsai, H.-Z., Li, H.-J., and Wu, H.-C. Development of therapeutic antibodies for the treatment of diseases. *Journal of biomedical science*, 27:1–30, 2020.

Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., and Ma, J. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *bioRxiv*, 2022. doi: 10.1101/2022.07.10.499510. URL https://www.biorxiv.org/content/early/2022/10/16/2022.07.10.499510.

Malkin, N., Jain, M., Bengio, E., Sun, C., and Bengio, Y. Trajectory balance: Improved credit assignment in gflownets. *Advances in Neural Information Processing Systems*, 35:5955–5967, 2022.

Martinkus, K., Ludwiczak, J., Liang, W.-C., Lafrance-Vanasse, J., Hotzel, I., Rajpal, A., Wu, Y., Cho, K., Bonneau, R., Gligorijevic, V., et al. Abdiffuser: full-atom generation of in-vitro functioning antibodies. *Advances in Neural Information Processing Systems*, 36, 2024.

Murali, S., Rustandi, R. R., Zheng, X., Payne, A., and Shang, L. Applications of surface plasmon resonance and biolayer interferometry for virus–ligand binding. *Viruses*, 14(4):717, 2022.

Peng, H.-P., Lee, K. H., Jian, J.-W., and Yang, A.-S. Origins of specificity and affinity in antibody–protein interactions. *Proceedings of the National Academy of Sciences*, 111 (26):E2656–E2665, 2014.

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

Polonelli, L., Pontón, J., Elguezabal, N., Moragues, M. D., Casoli, C., Pilotti, E., Ronzi, P., Dobroff, A. S., Rodrigues, E. G., Juliano, M. A., et al. Antibody complementarity-determining regions (cdrs) can display differential antimicrobial, antiviral and antitumor activities. *PLoS One*, 3 (6):e2371, 2008.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Ren, M., He, Z., and Zhang, H. Multi-objective antibody design with constrained preference optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Rizzi, A., Grinaway, P. B., Parton, D. L., Shirts, M. R., Wang, K., Eastman, P., Friedrichs, M., Pande, V. S., Branson, K., Mobley, D. L., and Chodera, J. D. Yank: A gpu-accelerated platform for alchemical free energy calculations. https://getyank.org, 2020.

Robin, G., Sato, Y., Desplancq, D., Rochel, N., Weiss, E., and Martineau, P. Restricted diversity of antigen binding residues of antibodies revealed by computational alanine scanning of 227 antibody–antigen complexes. *Journal of Molecular Biology*, 426(22):3729–3743, 2014.

Steinegger, M. and Söding, J. Mmseqs2: sensitive protein sequence searching for the analysis of massive data sets. *bioRxiv*, 2017. doi: 10.1101/079681. URL https://www.biorxiv.org/content/early/2017/06/07/079681.

Sutton, R. S., Barto, A. G., et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Tang, X., Dai, H., Knight, E., Wu, F., Li, Y., Li, T., and Gerstein, M. A survey of generative ai for de novo drug design: New frontiers in molecule and protein generation, 2024. URL https://arxiv.org/abs/2402.08703.

Tsurushita, N., Hinton, P. R., and Kumar, S. Design of humanized antibodies: from anti-tac to zenapax. *Methods*, 36(1):69–83, 2005.

Venkatraman, S., Jain, M., Scimeca, L., Kim, M., Sendera, M., Hasan, M., Rowe, L., Mittal, S., Lemos, P., Bengio, E., et al. Amortizing intractable inference in diffusion models for vision, language, and control. *arXiv preprint arXiv:2405.20971*, 2024.

Vreven, T., Hwang, H., Pierce, B. G., and Weng, Z. Prediction of protein–protein binding free energies. *Protein Science*, 21(3):396–404, 2012.

Wen, Y., Xu, C., Hu, J. Y.-C., and Liu, H. Alignab: Pareto-optimal energy alignment for designing nature-like antibodies. *arXiv preprint arXiv:2412.20984*, 2024.

Zhang, D., Chen, R. T. Q., Liu, C.-H., Courville, A., and Bengio, Y. Diffusion generative flow samplers: Improving learning signals through partial trajectory optimization, 2024a. URL https://arxiv.org/abs/2310.02679.

Zhang, D., Zhang, Y., Gu, J., Zhang, R., Susskind, J., Jaitly, N., and Zhai, S. Improving gflownets for text-to-image diffusion alignment. *arXiv preprint arXiv:2406.00633*, 2024b.

Zhang, R., Jiang, X., Cao, D., Yu, J., Chen, M., Fan, Z., Kong, X., Xiong, J., Zhang, Z., Zhang, W., et al. Packdock: a diffusion based side chain packing model for flexible protein-ligand docking. *bioRxiv*, pp. 2024–01, 2024c.

Zhang, Y., Zhang, Z., Zhong, B., Misra, S., and Tang, J. Diffpack: A torsional diffusion model for autoregressive protein side-chain packing. *Advances in Neural Information Processing Systems*, 36:48150–48172, 2023.

Zhang, Z., Lu, M., Qin, Y., Gao, W., Tao, L., Su, W., and Zhong, J. Neoantigen: a new breakthrough in tumor immunotherapy. *Frontiers in immunology*, 12:672356, 2021.

Zhou, X., Xue, D., Chen, R., Zheng, Z., Wang, L., and Gu, Q. Antigen-specific antibody design via direct energy-based preference optimization. *arXiv preprint arXiv:2403.16576*, 2024.

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

Technical Appendices and Supplementary Material

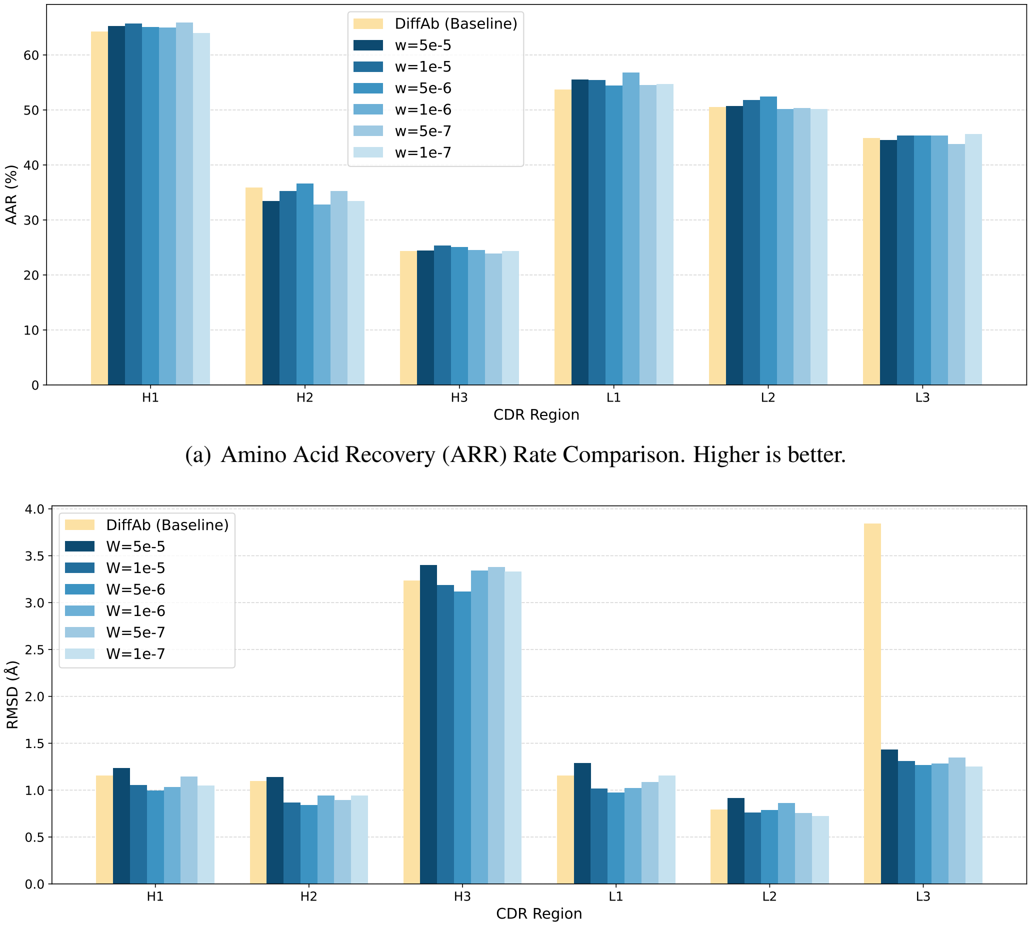
A. Experimental Setup Details

A.1. Hardware Specifications and Runtime

We conduct all experiments using a Linux machine with Intel(R) Xeon(R) Silver 4314 CPU with 512GB memory and one NVIDIA RTX A6000 48GB GPU. Training the first $195,000$ steps without the GFlowNet TB objective took ~ 27 hours and training the last $5,000$ steps took ~ 18 hours. Sampling 100 times for each CDR regions for every complex in the RAbD test dataset took ~ 12 hours.

A.2. Balancing Between Diffusion and Trajectory Balance Objectives

Although the method for computing the forward and backward flow of rewards is computationally expensive, the final trajectory balance loss is simply added to the diffusion reconstruction losses, as shown in Eqn. 18. The Trajectory Balance (TB) loss is typically ranges from 10^4 to 10^6, while the three diffusion losses have magnitudes between 0 and 1 after 195000 training steps. This necessitates a TB loss weight w to balance between the flow matching and reconstruction objectives. We train and test AbFlowNet with w ranging from $5e-5$ to $1e-7$ and find that learning rates between $1e-5$ and $1e-6$ are consistently better than the baseline set by DiffAb. Detailed results are shown in Figure 4.



(a) Amino Acid Recovery (ARR) Rate Comparison. Higher is better.

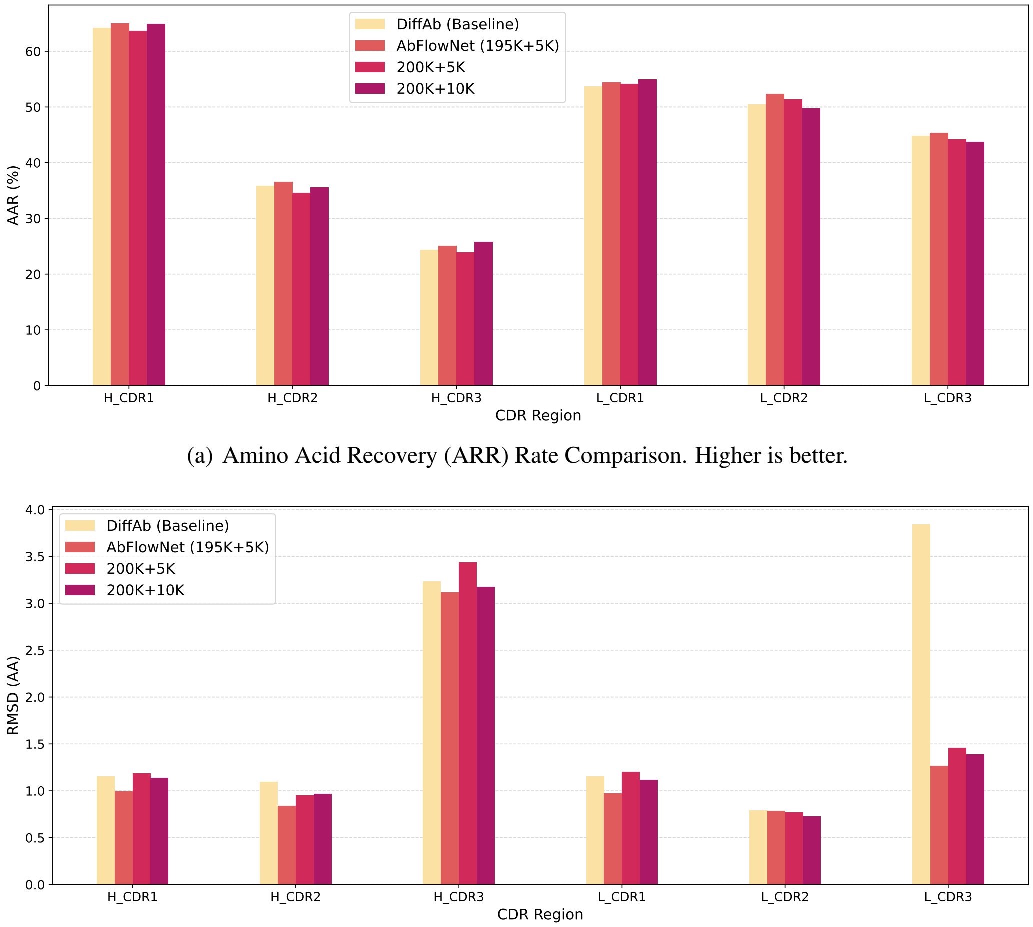
(b) Root Mean Square Deviation (RMSD) Comparison. Lower is better.

Figure 4. Hyperparameter search for TB loss weight w in Eqn. 18 on the RAbD (Adolf-Bryfogle et al., 2018b) dataset. The RMSD of DiffAb on L3 CDR region is significantly worse than AbFlowNet. We repeated the retrained DiffAb using a different seed to confirm this discrepancy (RMSD 4.06 Å).

A.3. Post-Training with Trajectory Balance

So far, we have focused on matching the number of gradient updates between the DiffAb baseline and AbFlowNet to isolate the effect of optimizing binding energy via trajectory balance.

However, training with sparse feedback is often framed as a separate stage after unsupervised learning (Zhou et al., 2024; Zhang et al., 2024b). We test this setup by first training the diffusion model on only the reconstruction objectives for 200K steps and a further 10K steps with the weighted TB loss enabled ($w = 5e-6$). We find that training beyond 200K steps with reconstruction objectives enables generally tends to overfit the dataset while training with only the TB loss objective harms metrics such as AAR and RMSD, similar to the findings of Zhou et al. (2024). Detailed results are shown in Figure 5.



(a) Amino Acid Recovery (ARR) Rate Comparison. Higher is better.

(b) Root Mean Square Deviation (RMSD) Comparison. Lower is better.

Figure 5. Training Diffusion+GFlowNet models with different training steps on the RAbD dataset (Adolf-Bryfogle et al., 2018b). Separating the reconstruction and flow matching steps do not meaningfully improve performance over AbFlowNet.

B. DiffAb Test Set Performance

The authors of DiffAb (Luo et al., 2022) proposed a test set consisting of 19 complexes with antigens from several well-known pathogens including SARS-CoV-2, MERS, influenza, and so on. Since these complexes are part of the SAbDab dataset (Dunbar et al., 2014) used for training, we filter our training complexes against the test set using a CDR-H3 sequence identity threshold of 50%. We retrain both DiffAb and AbFlowNet with this new filtered training set.

Table 3. Evaluation of the generated antibody CDRs (sequence-structure co-design) on the DiffAb test dataset (19 sequences).

CDR	Method	AAR↑	RMSD↓	IMP↑
H1	DiffAb	68.29%	1.090Å	53.96%
	AbFlowNet	**69.41%**	**0.944Å**	**63.50%**
H2	DiffAb	35.94%	**0.804Å**	**55.60%**
	AbFlowNet	**36.39%**	0.862Å	54.38%
H3	DiffAb	26.53%	**3.183Å**	**12.75%**
	AbFlowNet	**26.66%**	3.321Å	8.75%
L1	DiffAb	54.40%	**0.960Å**	62.48%
	AbFlowNet	**55.97%**	1.019Å	**64.84%**
L2	DiffAb	42.55%	**0.735**Å	80.61%
	AbFlowNet	**45.52%**	0.757Å	**84.47%**
L3	DiffAb	46.15%	1.127Å	37.07%
	AbFlowNet	**46.19%**	**1.180Å**	**37.26%**

Table 4. Summary of Top-1 CDR E_{total} and CDR-Ag ΔG (kcal/mol) of CDR-H3's designed by DiffAb and AbFlowNet on the DiffAb Test Dataset. (↓) denotes a smaller number is better.

Methods	**# Samples**	**CDR $\mathbf{E_{total}}$ (↓)**	**$+\mathbf{E_{total}}$ (%)**	**CDR-Ag $\mathbf{\Delta G}$ (↓)**	**$+\mathbf{\Delta G}$ (%)**	**Test Set Used**
Reference		1.63		-4.80		
DiffAb	100	26.33		11.50		
AbFlowNet (Ours)		4.23	89.5	1.47	149.7	No

C. Approaches to Determining Side-Chain Orientation with Neural Networks

Most generative methods—including DiffAb, AbDPO, AbFlowNet, AlignAb (Wen et al., 2024), AbX (Zhu et al., 2024), AbNovo (Ren et al., 2025), etc.—generate only the backbone structure. The orientation of the amino-acid chains isn't generated through a diffusion process, since it must follow structural constraints such as avoiding overlaps. Instead, we rely on a side-chain packing algorithm such as `PyRosetta PackRotamerMover` to find the ideal orientation of side-chains. The GNN based dyMEAN (Kong et al., 2023) jointly generates the backbone and side-chain orientations jointly but Table 1 shows that this approach under-performs using `PyRosetta`. There has been notable research into generating only side-chains conditioned on the backbone with diffusion neural networks (Zhang et al., 2023; 2024c) and given the limitations, it is a critical direction of future research.

D. Detailed Balance Objective

Detailed Balance (DB) (Bengio et al., 2021) is an alternative training objective to the standard flow matching constraint and Trajectory Balance which doesn't require enumerating states or sampling complete trajectories. Rather, DB requires the forward flow from state s to s', $F(s)P_F(s'|s)$ to match the backward flow $F(s')P_B(s|s')$. Concretely the DB objective is

$$F(s)P_F(s'|s) = F(s')P_B(s_{t-1}|s_t) \tag{19}$$

However, the flow of a nonterminal state s is generally not tractable, and hence it is parameterized with a neural network $F_\phi(\cdot)$. The forward and backward transition probabilities of the entire CDR S^t are $p(S^{t-1}|S^t) = \prod_{j=l}^{l+m} p(s_j^t|s_j^{t-1})$ and $q(S^t|S^{t-1}) = \prod_{j=l}^{l+m} q(s_j^t|s_j^{t-1})$ respectively. Therefore, the final DB objective is:

$$L_{\text{DB}}^t = \left(\log \frac{F_\phi(S^t)p(S^t|S^{t-1};\theta)}{F_\phi(S^{t-1})q(S^{t-1}|S^t)}\right)^2 \tag{20}$$

Pilot Attempt Using DB Objective In the GFlowNet framework, there are three equivalent optimization objectives: Flow Matching (FM), Detailed Balance (DB) and Trajectory Balance (TB) - each with their own tradeoffs. Flow Matching requires enumerating states and enforcing parity between incoming and outgoing flow. FM is not applicable since the number of states in diffusion models is infinite. We attempt using DB which only requires computing the forward and backward flow between two states and enforcing parity.

Similar to optimizing the TB objective outlined in Section 4.2, we uniformly sample a timestep t to compute L_{type}^t, L_{pos}^t and L_{ori}^t. Since we require adjacent state pairs to compute DB, we do a single step of denoising to obtain S^{t-1} from S^t. To enforce the DB objective 19, we must compute $F(S^t)$ and $F(S^{t-1})$. However, intermediate states S^{t-1} and S^t are noisy and therefore are not appropriate to be evaluated by a reward function, which would give noisy results. Following Zhang et al. (2024b), we define the linearization.

$$F_\phi(S^t) = \tilde{F}_\phi(S^t)R(\hat{S^0}) = \tilde{F}_\phi(S^t)R(FullDenoise_\theta(S^t)) \tag{21}$$

where $\tilde{F}_\phi()$ is a scalar function that scales the reward of the estimated fully denoised state. Being a diffusion model, we can fully denoise any noisy state albeit with a sacrifice in quality.

Key Bottlenecks At this stage, we run into the key issue that precludes the use of DB in CDR design. We need to compute the energy of the designed CDR using a tool such as `InterfaceAnalyzer` in the `Rosetta` (Chaudhury et al., 2010) software package. `InterfaceAnalyzer` requires the designed antigen-antibody structure to be complete with side-chains. However, diffusion models generally generate only the backbone and rely on a search-based side-chain packing algorithms such as `PackRotamerMover`. Both `PackRotamerMover` and `InterfaceAnalyzer` are CPU-based utilities and it takes 10.81 seconds to process a single CDR .pdb file. Determining the energy for the two states for each item in the mini-batch (16 in our experiments) requires $\sim$ 93 seconds even when parallelized over a 32+ core machine, including multiple data migration costs between the GPU, CPU and disk. This is in contrast to the millisecond-scale time required for the forward and backward passes.

Therefore, the training runtime is dominated by the time it takes to compute the binding energy reward and training becomes infeasible.

Neural Surrogate for Rosetta's InterfaceAnalyzer To the best of our knowledge, a neural network alternative to `PackRotamerMover` and `InterfaceAnalyzer` does not exist. We tried to train a transformer-based neural network to simulate the function of `InterfaceAnalyzer` directly from the output of the diffusion model. However, this neural network had very low agreement with the `InterfaceAnalyzer` tool (Pearson's coefficient 0.21), which itself is unreliable (Vreven et al., 2012). This is expected since side-chains play a central role in determining binding affinity (Polonelli et al., 2008). Another drawback of the DB objective is the need to compute binding energy of generated CDRs which are not guaranteed to be geometrically plausible.

In light of our findings, we finally committed to Trajectory Balance as the only feasible objective for training AbFlowNet despite the need to sample full trajectories.

E. Limitation

1) The Trajectory Balance objective requires fully generating a CDR, which in our setup requires 100 forward passes with the neural network for each gradient update. 2) As the *in vitro* affinity data for all training complexes is not available and for fair comparison with existing methods, we used `Pyrosetta InterfaceAnalyzer` which is an unreliable estimator of binding energy. 3) Due to compute constraints, we sampled 100 CDRs per complex in Table 6.1. Sampling at higher rates would potentially increase Top-1 metrics.