# Decoupled Sequence and Structure Generation for Realistic Antibody Design

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

## Abstract

Recently, deep learning has made rapid progress in antibody design, which plays a key role in the advancement of therapeutics. A dominant paradigm is to train a model to jointly generate the antibody sequence and the structure as a candidate. However, the joint generation requires the model to generate both the discrete amino acid categories and the continuous 3D coordinates; this limits the space of possible architectures and may lead to suboptimal performance. In response, we propose an antibody sequence-structure decoupling (ASSD) framework, which separates sequence generation and structure prediction. Although our approach is simple, our idea allows the use of powerful neural architectures and demonstrates notable performance improvements. We also find that the widely used non-autoregressive generators promote sequences with overly repeating tokens. Such sequences are both out-of-distribution and prone to undesirable developability properties that can trigger harmful immune responses in patients. To resolve this, we introduce a composition-based objective that allows an efficient trade-off between high performance and low token repetition. Our results demonstrate that ASSD consistently outperforms existing antibody design models, while the composition-based objective successfully mitigates token repetition of non-autoregressive models. Our code is available at `https://anonymous.4open.science/r/ASSD_public-5E17`.

## 1 Introduction

Antibodies are Y-shaped proteins that detect and neutralize the disease-causing agents. Due to high specificity and binding affinity, they are recognized as one of the most promising drug modalities (Mullard, 2022). Historically, antibody development relied on experimental and physics-based computational approaches, which are laborious and time-consuming (Kim et al., 2023). In response, there is a growing trend to integrate deep learning-based approaches for antibody development.

Early deep generative models (Liu et al., 2020; Saka et al., 2021; Akbar et al., 2022) focused only on designing the antibody sequences, neglecting the importance of antibody structure in functionality (Martinkus et al., 2024). To address this, Jin et al. (2021) proposed the sequence-structure co-design to generate both antibody sequences and structures. Many works made notable progress, including Jin et al. (2021); Verma et al. (2023); Kong et al. (2022); Luo et al. (2022); Wu & Li (2024). Such models are based on the joint prediction of sequence and structure with a common architecture for both tasks. However, this restricts task-specific optimization of model architectures, potentially hindering higher sequence-structure modeling performance.

Additionally, recent works (Melnyk et al., 2023; Kong et al., 2022; Verma et al., 2023; Kong et al., 2023) show that sequence generation with non-autoregressive models achieves faster inference speed and better performance than the autoregressive counterparts, with its efficacy even extending to protein design (Gao et al., 2022; Zheng et al., 2023). However, we find that such models generate sequences that are disproportionately dominated by frequently occurring amino acid type (e.g., the natural sequence `ARMGSDYDVWFDY` versus non-autoregressive prediction `TRYYYYYYYYYDY`). Such sequences are out-of-distribution and prone to undesirable developability properties that can trigger harmful immune responses in patients (Jin et al., 2021; Wang et al., 2009).

**Contribution.** In this paper, we propose antibody sequence-structure decoupling (ASSD), a general framework for constructing a highly performant antibody sequence-structure co-design model. Our idea is to leverage Anfinsen's dogma, which states that the native conformation of a protein is determined by amino acid sequence (Anfinsen, 1973). To this end, instead of joint prediction, we decouple the sequence-structure design into two steps: a sequence design step followed by a structure prediction step. This decoupling approach is desirable since it allows the suitable choice of model architectures for each task and enables the use of large-scale sequence databases (without structure information) during the sequence design step. Despite the simplicity, such a decoupling of sequence-structure co-design has been overlooked in previous works.

We also propose a composition-based objective for antibody sequence generation that resolves the token repetition problem of non-autoregressive models. Specifically, our key idea is to augment the MLE objective with the maximization of similarity between amino acid compositions of generated and target sequences. To this end, we use REINFORCE trick (Williams, 1992; Fu et al., 2015) with respect to the composition vector which accounts for the rate of amino acids appearing in the sequences. Our idea is to let the model implicitly learn the dependency between the residues within a sequence, effectively preventing excessive repetition of any single amino acid.

Our experiments demonstrate that our sequence-structure decoupling approach improves performance in various antibody design experiments, while our training algorithm effectively prevents excessive token repetitions. Notably, our approach establishes a Pareto frontier over other non-autoregressive antibody design models, indicating optimal trade-offs between high sequence modeling capacity and low token repetition. Additionally, we demonstrate that our training algorithm can be generalized to protein design.

## 2 Related work

**Antibody design.** Many pioneering works on antibody design predicts only 1D sequences using CNN and LSTM (Liu et al., 2020; Saka et al., 2021; Akbar et al., 2022). Recently, Melnyk et al. (2023) improved 1D sequence modeling performance by repurposing a pre-trained Transformer-based English language model for sequence infilling. Frey et al. (2023) also demonstrated the potential of energy/score-based models for antibody sequence design with a walk-jump sampling scheme.

Instead of designing only 1D sequences, Jin et al. (2021) proposes sequence-structure co-design, which involves both 1D sequence and 3D structure infilling of CDRs. Specifically, Jin et al. (2021) represents the antibody-antigen complex with an E(3)-invariant graph and predicts the sequence in an autoregressive fashion. On the contrary, Kong et al. (2022) constructs an E(3)-equivariant graph to fully capture the 3D geometry and predicts sequence by a full-shot decoding scheme to speed up inference. Verma et al. (2023) formulates a coupled neural ODE system over the antibody nodes and further increases inference speed with a single round of full-shot decoding. Wu & Li (2024) introduces a four-level training strategy with various protein/antibody databases. Unlike deterministic GNN-based methods, Luo et al. (2022) and Martinkus et al. (2024) adopt diffusion probabilistic models to generate structures stochastically. Notably, Frey et al. (2023); Martinkus et al. (2024) supports their approach with laboratory experiments.

Previous approaches predict the sequence and structure jointly, except Martinkus et al. (2024) which introduces a structure-to-sequence decoupling strategy. However, this approach constrains itself to the Structural Antibody Database (SAbDab) (Dunbar et al., 2014), which is orders of magnitude smaller than sequence databases like the Observed Antibody Space (OAS) (Olsen et al., 2022). In this work, we demonstrate the superiority of sequence-to-structure decoupling over joint prediction strategy.

**Protein design.** Due to their relevance, methods in protein design can offer valuable insights into antibody design. Ingraham et al. (2019) represents the protein structure with a k-nearest neighbor graph and applies a Transformer-based encoder-decoder model to predict the sequences autoregressively. To fully capture the complex geometry within the graphs, Jing et al. (2020) introduces geometric vector perceptrons that can replace MLPs in GNNs. Tan et al. (2023) introduces an additional global module that captures the global context of the protein structure. To enable an application to a wide range of protein design problems, Dauparas et al. (2022) adopts an order-agnostic autoregressive model with random decoding orders. Instead, Gao et al. (2022) adopts a one-shot decoding scheme and drastically improves the inference speed. Recently,

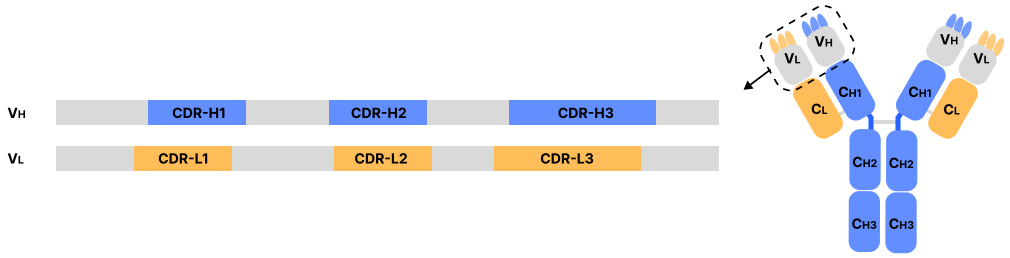



Figure 1: **Schematic structure of an antibody.** An antibody consists of a pair of heavy and light chains, each containing a variable region and constant regions. The variable region consists of three complementarity-determining regions (CDRs) and its complement called the framework regions. We aim to design heavy chain CDRs, which contribute the most to antibody-antigen interaction.

Zheng et al. (2023) showed state-of-the-art performance by incorporating a lightweight structural adapter into a protein language model.

Notably, Melnyk et al. (2023); Kong et al. (2022); Verma et al. (2023); Gao et al. (2022); Zheng et al. (2023) adopt non-autoregressive factorization trained with MLE objective or its slight variations. In this paper, we showcase that this approach promotes highly repetitive sequences and propose a novel training strategy to mitigate this problem while maintaining a comparable level of performance.

## 3 Preliminaries and notations

**Preliminaries.** As shown in Figure 1, an antibody consists of a pair of light chains and heavy chains. Both the light chain and heavy chain are composed of a variable region ($V_L$;$V_H$) and constant region(s) ($C_L; C_{H1}, C_{H2}, C_{H3}$). Each variable region could be further decomposed into three complementarity-determining regions (CDR-L1, CDR-L2, CDR-L3; CDR-H1, CDR-H2, CDR-H3) and a framework region, which is the complement of CDRs. Following previous works (Jin et al., 2021; Kong et al., 2022; Verma et al., 2023; Luo et al., 2022; Melnyk et al., 2023), we narrow the scope of our task to sequence-structure co-design of heavy chain CDRs as they contribute the most to antibody-antigen interaction.

**Notations.** We denote the amino acid vocabulary set with $\mathcal{A}$. Let $L$ be the number of residues in CDR. We denote the *ground-truth* sequence and structure of a heavy chain CDR as $\mathbf{s} = (s_1, \ldots, s_L)$ and $\mathbf{x} = (x_1, \ldots, x_L)$ and the *predicted* sequence and structure as $\hat{\mathbf{s}}$ and $\hat{\mathbf{x}}$. We denote all conditional information (e.g., framework region, antigen, initializations) as $\mathbf{c}$.

## 4 Methods

Here, we provide an overview of this section. Our goal is to train a non-autoregressive generator that can generate the sequence of amino acids and the associated 3D information conditioned on some semantic information. Motivated by Anfinsen's dogma, which states that the protein structure is determined by its amino acid sequence (Anfinsen, 1973), we propose an antibody sequence-structure decoupling (ASSD) framework that decouples sequence-structure co-design into the sequence design step (Section 4.2), followed by the structure generation step (Section 4.3). Additionally, we resolve excessive token repetition of non-autoregressive models with a composition-based sequence-level objective (Section 4.2).

### 4.1 Problem formulation & algorithm overview

**Problem statement.** We first formulate our problem for designing an antibody. Following previous works (Jin et al., 2021; Kong et al., 2022; Verma et al., 2023; Luo et al., 2022; Melnyk et al., 2023), we narrow the scope of our task to sequence-structure co-design of heavy chain complementarity-determining regions (CDRs) in the antibody as they contribute the most to antibody-antigen interaction. The heavy

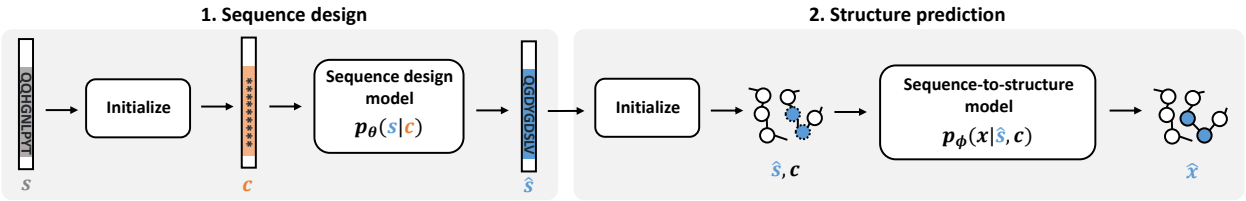



Figure 2: **Overview of antibody sequence-structure decoupling (ASSD) framework.** ASSD first designs CDR sequences with a sequence design model and then predicts the corresponding structure with a sequence-to-structure model. $\boldsymbol{s}, \boldsymbol{x}$ are the ground-truth, and $\hat{\boldsymbol{s}}, \hat{\boldsymbol{x}}$ are the predicted CDR sequence and structure. $\boldsymbol{c}$ denotes conditional information, e.g., framework region, antigen, and sequence/structure initialization.

chain CDRs are represented by an amino acid sequence associated with 3D positions. We provide more information on the antibody structure in Figure 1.

To be specific, we let $\mathcal{A}$ denote the amino acid vocabulary set and $L$ the number of residues in the CDR. The CDR is represented by the sequence $\boldsymbol{s} = (s_1, \ldots, s_L)$ and the positional information $\boldsymbol{x} = (x_1, \ldots, x_L)$ where $s_i \in \mathcal{A}$ corresponds to a particular type of amino acid and $x_i \in \mathbb{R}^{m \times 3}$ the associated 3D positional information with $m$ backbone atoms. Our goal is to train a generator $p_\theta(\boldsymbol{s}, \boldsymbol{x}|\boldsymbol{c})$ for the amino acid sequence $\boldsymbol{s}$ and the positional information $\boldsymbol{x}$, given the conditional information $\boldsymbol{c}$, e.g., the framework region, antigen, and initialization.

**Algorithm overview.** In ASSD, we parameterize the generator $p_{\theta,\phi}(\boldsymbol{s}, \boldsymbol{x}|\boldsymbol{c})$ as a sequential composition of sequence design module $p_\theta(\boldsymbol{s}|\boldsymbol{c})$ and the structure prediction module $p_\phi(\boldsymbol{x}|\boldsymbol{s}, \boldsymbol{c})$, i.e., we let $p_{\theta,\phi}(\boldsymbol{s}, \boldsymbol{x}|\boldsymbol{c}) = p_\theta(\boldsymbol{s}|\boldsymbol{c}) p_\phi(\boldsymbol{x}|\boldsymbol{s}, \boldsymbol{c})$. This decoupling is motivated by a well-known postulate in molecular biology, called Anfinsen's dogma, which states that the native conformation of a protein is determined solely by its amino acid sequence (Anfinsen, 1973). We provide an overview of our algorithm in Figure 2.

Additionally, we consider a non-autoregressive sequence generative model where the amino acids are predicted independently from each other, i.e., $p_\theta(\boldsymbol{s}|\boldsymbol{c}) = \prod_{i=1}^{L} p_\theta(s_i|\boldsymbol{c})$, as recent works (Kong et al., 2022; 2023; Verma et al., 2023; Zheng et al., 2023) demonstrate both faster speed and better performance compared to the autoregressive counterparts. However, since the non-autoregressive models generate sequences with excessively repetitive tokens, we propose a new loss function to prevent this.

### 4.2 Sequence design

In this section, we introduce the training algorithm and implementation details for a sequence-only antibody design model $p_\theta(\boldsymbol{s}|\boldsymbol{c})$. Our insight is that the repetitive tokens from the generator stem from the mode-covering behavior of MLEs, i.e., each amino acid prediction aims for an average of different modes, i.e., the most frequent element. To overcome this limitation, we augment MLE training with a new mode-seeking training objective based on the compositional similarity of protein sequences that mitigates the repetitive token problem.

**Intuitions for repetitive sequences.** We are motivated by the observation that non-autoregressive sequence generators exhibit the degenerate behavior of generating excessive token repetitions, e.g., $\boldsymbol{s} = \texttt{ARGYYYYYYY}$ where $\texttt{A}, \texttt{R}, \texttt{G}, \texttt{Y} \in \mathcal{A}$ are the possible types of amino acids. However, this is critical evidence that the generator fails to learn the underlying dataset – i.e., the generated sequences are out-of-distribution. Also, overly repetitive sequences should be avoided since they may cause developability issues such as aggregation that trigger harmful immune responses in patients (Jin et al., 2021; Wang et al., 2009).

Figure 3: Example to illustrate token repetition problem.

To better illustrate this problem, consider a hypothetical example with five possible amino acids $\{\texttt{A}, \texttt{R}, \texttt{G}, \texttt{Y}, \texttt{D}\}$. The training dataset consists of the same conditional in-

**Algorithm 1** Training sequence design model

**Require:** Antibody sequence dataset $\mathcal{D} = \{(\boldsymbol{s}, \boldsymbol{c})\}$, antibody sequence generator $p_\theta(\boldsymbol{s}|\boldsymbol{c})$.
**repeat**
  Sample $\mathcal{B} = \{(\boldsymbol{s}^{(i)}, \boldsymbol{c}^{(i)})\}_{i=1}^M \sim \mathcal{D}$.
  **for** $i = 1, \ldots, M$ **do**
    Sample $\hat{\boldsymbol{s}}^{(i)} \sim p_\theta(\boldsymbol{s}|\boldsymbol{c}^{(i)})$ and compute $d(\boldsymbol{s}^{(i)}, \hat{\boldsymbol{s}}^{(i)})$.
  **end for**
  Compute $b = \frac{1}{M}\sum_{i=1}^M d(\boldsymbol{s}^{(i)}, \hat{\boldsymbol{s}}^{(i)})$ and update $\theta \leftarrow \arg\min \mathcal{L}_{\text{seq}}(\theta, \mathcal{B})$ as in Equation (1).
**until** converged

formation $\boldsymbol{c}$ and sequences $\boldsymbol{s}$ in Figure 3. The MLE objective of a non-autoregressive model $p_\theta(\boldsymbol{s}|\boldsymbol{c}) = \prod_{i=1}^L p_\theta(s_i|\boldsymbol{c})$ is given by

$$\mathcal{L}_{\text{MLE}}(\theta) = \sum_{n=1}^N \sum_{i=1}^L \log p_\theta(s_i^{(n)}|\boldsymbol{c}) = \sum_{i=1}^L \sum_{n=1}^N \ell_{\text{ce}}(s_i^{(n)}, \boldsymbol{p}_i),$$

where $\boldsymbol{p} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_L) = f_\theta(\boldsymbol{c})$ is the head output distribution. If we train the model with this objective, it assigns the highest likelihood to the sequence `YYYYYY` which is overly repetitive (proof in Appendix A). Such a phenomenon indeed occurs for antibody or protein sequences, where in certain regions, some residues have a higher frequency than others (Nikula et al., 1995; Shi et al., 2014).

**Loss function.** We train our sequence-only antibody design model $p_\theta(\boldsymbol{s}|\boldsymbol{c})$ on the loss function $\mathcal{L}_{\text{seq}}(\theta) = \mathcal{L}_{\text{NLL}}(\theta) + \alpha\mathcal{L}_{\text{comp}}(\theta)$, where $\mathcal{L}_{\text{NLL}}(\theta)$ is the negative log-likelihood (NLL) objective, $\mathcal{L}_{\text{comp}}$ is the proposed composition-based loss function that regularize the repetitive tokens, and $\alpha > 0$ is some hyperparameter. In particular, given a sample $(\boldsymbol{s}, \boldsymbol{c})$ from the dataset, the loss function is defined as follows:

$$\mathcal{L}_{\text{NLL}}(\theta) = -\log p_\theta(\boldsymbol{s}|\boldsymbol{c}), \quad \mathcal{L}_{\text{comp}}(\theta) = \mathbb{E}_{\hat{\boldsymbol{s}} \sim p_\theta(\cdot|\boldsymbol{c})}[d(\boldsymbol{s}, \hat{\boldsymbol{s}})],$$

where $d(\boldsymbol{s}, \hat{\boldsymbol{s}})$ is the compositional distance between the input amino acid sequence $\boldsymbol{s}$ and the reconstructed amino acid sequence $\hat{\boldsymbol{s}}$. Following the existing NLP literature (Mathur et al., 2019; Kim et al., 2021; Zhang et al., 2019), we propose to use the cosine similarity defined as follows:

$$d(\boldsymbol{s}, \hat{\boldsymbol{s}}) = -\operatorname{CosineSimilarity}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\frac{\boldsymbol{y}^\top \hat{\boldsymbol{y}}}{\|\boldsymbol{y}\|\|\hat{\boldsymbol{y}}\|},$$

where $\boldsymbol{y}, \hat{\boldsymbol{y}}$ is the composition vector of the sequence $\boldsymbol{s}, \hat{\boldsymbol{s}}$ that counts the occurrence of each amino acid type in the sequence. For example, $y_j = \sum_{i=1}^L \mathbf{1}_j(s_i)$ where $\mathbf{1}_j$ is a binary indicator that has the value of one if the amino acid $s_i$ is the $j$-th type in the vocabulary $\mathcal{A}$ and zero otherwise.

**Training with REINFORCE.** To train our generator, we use the REINFORCE trick (Williams, 1992; Fu et al., 2015) with a baseline $b = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{c})\sim\mathcal{B}}[d(\boldsymbol{s}, \hat{\boldsymbol{s}})]$ to reduce the variance of the gradients (Appendix B). This results in the following training objective:

$$\mathcal{L}_{\text{seq}}(\theta, \mathcal{B}) = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{c})\sim\mathcal{B}}\left[-\sum_{i=1}^L \log p_\theta(s_i|\boldsymbol{c}) + \alpha \cdot [d(\boldsymbol{s}, \hat{\boldsymbol{s}}) - b] \log p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})\right]. \tag{1}$$

We describe the full algorithm in Algorithm 1.

**Implementation.** Our sequence-structure decoupling approach enables the use of a large-scale sequence database during the sequence design step. We implicitly exploit this advantage by adopting a protein language model (pLM) as the sequence generator. In particular, we choose ESM2-650M (Lin et al., 2023) as our starting point for training and use the LoRA fine-tuning strategy (Hu et al., 2021) (details in Appendix C).

**Algorithm 2** Training structure prediction model

**Require:** A trained antibody sequence generator $p_\theta(\boldsymbol{s}|\boldsymbol{c})$, a sequence-to-structure model $p_\phi(\boldsymbol{x}|\boldsymbol{s},\boldsymbol{c})$, and antibody sequence-structure dataset $\mathcal{D} = \{(\boldsymbol{s},\boldsymbol{x},\boldsymbol{c})\}$.
**repeat**
  Sample $\mathcal{B} = \{(\boldsymbol{s}^{(i)},\boldsymbol{x}^{(i)},\boldsymbol{c}^{(i)})\}_{i=1}^{M} \sim \mathcal{D}$.
  **for** $i = 1,\ldots,M$ **do**
    Sample $\hat{\boldsymbol{s}}^{(i)} \sim p_\theta(\boldsymbol{s}|\boldsymbol{c}^{(i)})$ and predict $\hat{\boldsymbol{x}}^{(i)} \sim p_\phi(\boldsymbol{x}|\hat{\boldsymbol{s}}^{(i)},\boldsymbol{c}^{(i)})$.
  **end for**
  Update $\phi \leftarrow \arg\min \mathcal{L}_{\text{struct}}(\phi,\mathcal{B})$ as in Equation (2).
**until** converged

### 4.3 Structure prediction

Here, we describe our sequence-to-structure design model $p_\phi(\boldsymbol{x}|\boldsymbol{s},\boldsymbol{c})$. We train the sequence-to-structure model based on the following loss function:

$$\mathcal{L}_{\text{struct}}(\phi,\mathcal{B}) = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{x},\boldsymbol{c})\sim\mathcal{B}}\mathbb{E}_{\hat{\boldsymbol{s}}\sim p_\theta(\cdot|\boldsymbol{c})}\left[\frac{1}{L}\sum_{i=1}^{L}\ell_{\text{huber}}(x_i,\hat{x}_i)\right], \tag{2}$$

where $\ell_{\text{huber}}$ is the Huber loss to avoid numerical stability (Jin et al., 2021; Kong et al., 2022), $\hat{\boldsymbol{x}} = (\hat{x}_1,\ldots,\hat{x}_L) \sim p_\phi(\cdot|\hat{\boldsymbol{s}},\boldsymbol{c})$ is the predicted sequence, and $p_\theta(\cdot|\boldsymbol{c})$ is the fixed sequence generative model trained using the loss function proposed in Section 4.2. Note that we train the model to predict the structure from generated antibody sequence $\hat{\boldsymbol{s}}$ instead of the ground-truth sequence from the dataset. This is to prevent poor generalization performance during inference, caused by the exposure bias (Arora et al., 2022; Bengio et al., 2015; Ranzato et al., 2016).

Exposure bias may occur since during training, the model conditions on ground-truth sequence, while during inference, it conditions on its sequence predictions. This discrepancy leads to error accumulation, where initial mistakes propagate and compound, causing the generated sequence to deviate significantly from realistic outputs. Consequently, using ground-truth sequence for training may severely undermine the model robustness (Arora et al., 2022; Bengio et al., 2015; Ranzato et al., 2016). We empirically demonstrate the benefit of using the generated sequence as input to the structure prediction model in Appendix D.

While one could consider any graph neural network (GNN) to generate the structural information, we adopt the architecture of MEAN (Kong et al., 2022) as our structure prediction model. In ablations, we demonstrate that our sequence-structure decoupling approach can improve the performance of any GNN-based joint sequence-structure co-design model.[1]

## 5 Antibody design experiments

**Overview.** In this section, we evaluate our model on four antibody benchmark experiments, including the SAbDAb benchmark (Section 5.1), RAbD benchmark (Section 5.2), antibody affinity optimization (Section 5.3), and CDR-H3 design with docked templates (Section 5.4).

**Baselines.** We select LSTM, AR-GNN, and RefineGNN as the autoregressive baselines with implementations provided in Jin et al. (2021). For non-autoregressive baselines, we adopt MEAN and LM-Design adapted for antibody design. We have not included Verma et al. (2023); Wu & Li (2024); Martinkus et al. (2024); Frey et al. (2023) as baselines as their training codes are not publicly available. The training details of ASSD and baselines are provided in Appendix C.

**Metrics.** Our main sequence and structure modeling evaluation metrics are amino acid recovery (AAR) and root mean squared deviation (RMSD), respectively. AAR is defined as $\text{AAR}(\mathbf{s},\hat{\mathbf{s}}) = \sum_{i=1}^{L}\mathbb{I}(s_i = \hat{s}_i)/L$,

[1]We note that antibody structure prediction models such as IgFold (Ruffolo et al., 2023) cannot predict the structure of incomplete sequences like CDR and thus perform poorly under this setting.

Table 1: **AAR, RMSD, and $p_{\text{rep}}$ on CDR-H1 of SAbDab dataset.** AR denotes autoregressive models and NAR denotes non-autoregressive models. The best and suboptimal are bolded and underlined, respectively, with $p_{\text{rep}}$ restricted to non-autoregressive models. ASSD (MLE) achieves the best performance, while ASSD (MLE + RL) reduces $p_{\text{rep}}$ with comparable performance.

| | **Models** | CDR-H1 | | |
|---|---|---|---|---|
| | | AAR(↑) (%) | RMSD(↓) | $p_{\text{rep}}$(↓) (%) |
| AR | LSTM | 39.47±2.11 | - | 0.00±0.00 |
| | AR-GNN | 48.39±4.72 | 2.90±0.17 | 0.00±0.00 |
| | RefineGNN | 42.03±2.88 | 0.87±0.10 | 0.00±0.00 |
| NAR | MEAN | 59.32±4.71 | 0.91±0.10 | 0.29±0.49 |
| | LM-Design | 64.88±1.96 | - | 0.02±0.06 |
| | ASSD (MLE) | **69.41±2.72** | **0.80±0.09** | 0.02±0.06 |
| | ASSD (MLE + RL) | 68.71±2.25 | 0.86±0.09 | **0.00±0.00** |

Table 2: **AAR, RMSD, and $p_{\text{rep}}$ on CDR-H2 of SAbDab dataset.** ASSD (MLE) achieves the best performance, while ASSD (MLE + RL) reduces $p_{\text{rep}}$ with comparable performance.

| | **Models** | CDR-H2 | | |
|---|---|---|---|---|
| | | AAR(↑) (%) | RMSD(↓) | $p_{\text{rep}}$(↓) (%) |
| AR | LSTM | 30.41±2.47 | - | 0.00±0.00 |
| | AR-GNN | 38.03±2.81 | 2.30±0.16 | 0.00±0.00 |
| | RefineGNN | 32.05±2.23 | 0.79±0.06 | 0.00±0.00 |
| NAR | MEAN | 48.85±2.32 | 0.88±0.07 | 0.03±0.08 |
| | LM-Design | 55.31±3.92 | - | 0.02±0.07 |
| | ASSD (MLE) | **62.10±3.87** | **0.73±0.06** | 0.02±0.07 |
| | ASSD (MLE + RL) | 61.07±0.08 | 0.78±0.06 | **0.00±0.00** |

where $\hat{\mathbf{s}}$ is the generated CDR sequence and $\mathbf{s}$ is the ground-truth CDR sequence. RMSD is defined by comparing $C_\alpha$ coordinates of generated and ground-truth CDR structures, following Jin et al. (2021); Kong et al. (2022). Although there is a controversy about whether AAR and RMSD are suitable metrics, they are still the most widely used metrics across the antibody/protein literature to assess sequence/structure modeling abilities. Further justifications and discussions on evaluation metrics are in **??**.

As none of the antibody sequences in the SAbDab dataset contain more than six token repetitions, we define a sequence with more than six repetitions as repetitive. Based on this definition, we develop a new metric called the percentage of repetitive tokens, $p_{\text{rep}}$. Given a model's sequence predictions $\hat{\mathcal{D}} = \{\hat{\mathbf{s}}^{(n)}\}_{n=1}^{N}$, $p_{\text{rep}} = \sum_{n=1}^{N} \mathbb{I}(\hat{\mathbf{s}}^{(n)} \text{ is repetitive}) \times 100\%/N$.

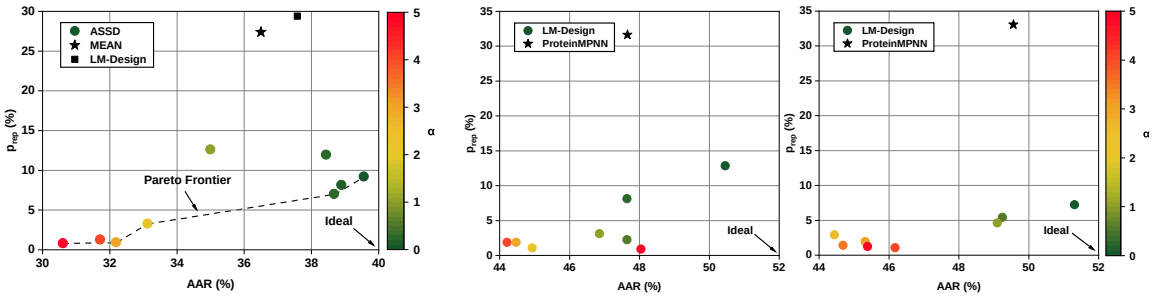
### 5.1 SAbDab benchmark

In this section, we evaluate the sequence and structure modeling performance for CDR-H1, CDR-H2, and CDR-H3 using the Structural Antibody Database (SAbDab) (Dunbar et al., 2014).

**Data.** Following Kong et al. (2022), we select 3977 IMGT-numbered (Lefranc et al., 2003) complexes from SAbDab that contain the full heavy chain, light chain, and antigen sequence and structure. We then split the complexes into train, validation, and test sets according to the CDR clusterings. Specifically, we use MMseqs2 (Steinegger & Söding, 2017) to assign antibodies with CDR sequence identity above 40% to the same cluster, where the sequence identity is computed with the BLOSUM62 substitution matrix (Henikoff & Henikoff, 1992). Then we conduct a 10-fold cross-validation by splitting the clusters into a ratio of 8:1:1 for train/valid/test sets, respectively. Detailed statistics of the 10-fold dataset splits are provided in Appendix E.

Table 3: **AAR, RMSD, and $p_{\text{rep}}$ on CDR-H3 of SAbDab dataset.** ASSD (MLE) achieves the best performance, while ASSD (MLE + RL) reduces $p_{\text{rep}}$ with comparable performance.

| | **Models** | CDR-H3 | | |
|---|---|---|---|---|
| | | AAR(↑) (%) | RMSD(↓) | $p_{\text{rep}}$(↓) (%) |
| AR | LSTM | 15.82±1.63 | - | 0.00±0.00 |
| | AR-GNN | 18.72±0.82 | 3.60±0.58 | 0.03±0.07 |
| | RefineGNN | 24.44±1.94 | 2.24±0.14 | 0.11±0.13 |
| NAR | MEAN | 36.50±1.60 | 2.23±0.07 | 27.40±9.23 |
| | LM-Design | 37.58±1.16 | - | 29.40±10.01 |
| | ASSD (MLE) | **39.56±1.39** | 2.21±0.08 | 9.22±3.76 |
| | ASSD (MLE + RL) | 38.67±1.64 | **2.19±0.09** | **7.02±2.91** |




(a) (b)

Figure 4: **Effect of $\alpha$ on AAR and $p_{\text{rep}}$.** (a) Result for CDR-H3 of SAbDab benchmark. ASSD approach achieves the Pareto frontier over the non-autoregressive baselines. (b) Result for CATH 4.2 (*left*) and CATH 4.3 (*right*) benchmarks. LM-Design trained with our MLE + RL objective maintains performance comparable to ProteinMPNN, while reducing $p_{\text{rep}}$ significantly.

**Results.** Table 1, Table 2, and Table 3 presents our results on the SAbDab benchmark. First, we highlight that ASSD trained with the MLE objective outperforms all baselines in AAR and RMSD. However, its $p_{\text{rep}}$ is much higher than the autoregressive baselines, implying that some of its sequence designs are invalid. Training with our MLE-RL objective in Equation (1) successfully reduces token repetitions, while maintaining a similar performance. In particular, for CDR-H1 and CDR-H2, we reduced token repetition to 0; for CDR-H3, we reduced $p_{\text{rep}}$ to 7.02%, which corresponds to a 23.86% reduction in token repetition compared to the MLE-only objective.

We focus on CDR-H3 to showcase that $p_{\text{rep}}$ can be further decreased by increasing $\alpha$ in the objective function. Figure 4(a) show shows that our approach establishes a Pareto frontier over other non-autoregressive baselines in terms of $p_{\text{rep}}$ and AAR. Note that $p_{\text{rep}}$ can be reduced to a level comparable to autoregressive models while maintaining AAR above 30%, a level of performance required for CDR design models (Melnyk et al., 2023). This contrasts MEAN and LM-Design, whose $p_{\text{rep}}$ is above 25% despite the high AAR.

### 5.2 RAbD benchmark

In this section, we assess the ability to model CDR-H3 on the RosettaAntibodyDesign (RAbD) dataset, a benchmark containing 60 diverse antibody-antigen complexes curated by Adolf-Bryfogle et al. (2018).

**Data.** We use the SAbDab dataset as the train/validation set and the RAbD dataset as the test set. We eliminate antibodies in SAbDab whose CDR-H3 sequences are in the same cluster as the RAbD dataset, then split the remaining clusters into train and validation sets with a ratio of 9:1. This gives 3418 and 382 antibodies for the training and validation sets, respectively.

Table 4: **AAR, Cosim, TM-score, RMSD, and $p_{\text{rep}}$ on RAbD dataset.** AR denotes autoregressive models and NAR denotes non-autoregressive models. The best and suboptimal are bolded and underlined, respectively, with $p_{\text{rep}}$ restricted to non-autoregressive models. ASSD (MLE) and ASSD (MLE + RL) achieve the best performance, with ASSD (MLE + RL) achieving the lowest $p_{\text{rep}}$ among NAR models.

| | **Models** | AAR (↑) (%) | Cosim (↑) | TM-score (↑) | RMSD (↓) | $p_{\text{rep}}$ (↓) (%) |
|---|---|---|---|---|---|---|
| AR | LSTM | 16.15 | 0.5462 | - | - | 0.00 |
| | AR-GNN | 18.85 | 0.5963 | 0.9630 | 3.56 | 0.00 |
| | RefineGNN | 26.33 | 0.5797 | 0.9648 | 1.80 | 0.00 |
| NAR | MEAN | 37.15 | 0.5832 | 0.9806 | 1.82 | 8.33 |
| | LM-Design | 38.77 | 0.5869 | - | - | 15.00 |
| | ASSD (MLE) | **40.81** | 0.6015 | 0.9825 | 1.78 | 5.00 |
| | ASSD (MLE + RL) | 40.53 | **0.6157** | **0.9830** | **1.76** | **3.33** |

Table 5: **Change in binding affinity $\Delta\Delta G$ after affinity optimization on SKEMPI V2.0 dataset.** The best and suboptimal are bolded and underlined, respectively, with $p_{\text{rep}}$ restricted to non-autoregressive models. ASSD (MLE) generates re-designs with the highest binding affinities on average. ASSD (MLE + RL) achieves similar performance, while reducing $p_{\text{rep}}$ by 16.67%.

| | Random* | AR-GNN | RefineGNN | MEAN | ASSD (MLE) | ASSD (MLE + RL) |
|---|---|---|---|---|---|---|
| $\Delta\Delta G$(↓) | +1.520 | -2.279 | -5.437 | -7.199 | **-8.454** | -8.162 |
| $p_{\text{rep}}$(↓) (%) | - | 0.000 | 3.774 | 50.943 | 11.321 | **9.434** |

**Metrics.** In addition to AAR and RMSD, we include cosine similarity (abbreviated CoSim) and TM-score (Zhang & Skolnick, 2004; Xu & Zhang, 2010) to evaluate the sequence composition similarity and global structural similarity.

**Results.** We report our results in Table 4. ASSD with MLE-only training achieves the highest AAR. While maintaining a similar level of AAR, we reduce token repetition by 33.4% with our MLE-RL objective function. Additionally, ASSD shows the highest structure modeling capacity, reflected by a high TM-score and low RMSD. This indicates the superiority of our sequence-structure decoupling approach over the joint sequence-structure co-design models like MEAN, RefineGNN, and AR-GNN.

### 5.3 Antibody affinity optimization

The goal of this task is to optimize the binding affinity of a given antibody-antigen complex by re-designing the CDR-H3 sequence and structure. Following Kong et al. (2022), we pre-train the models on the SAbDab dataset and fine-tune them with the ITA algorithm (Yang et al., 2020) on the SKEMPI V2.0 dataset (Jankauskaitė et al., 2019). To evaluate the change in binding affinity $\Delta\Delta G$, we use the official checkpoint of the oracle $f$ from Shan et al. (2022). Since $\Delta\Delta G$ evaluation requires structure, only sequence-structure co-design models are used as the baselines.

**ITA algorithm.** Following Kong et al. (2022), we adopt the ITA algorithm adjusted for continuous properties. The algorithm consists of two parts – (1) augmenting the dataset and (2) training the model on the augmented dataset. To augment the dataset $\mathcal{D}$, we select an antibody-antigen complex from $\mathcal{D}$ and generate $\mathcal{M} = 20$ candidate structures with the model. If a candidate is valid and exhibits $f < 0$, we add it to $\mathcal{D}$ and form the augmented dataset $\mathcal{Q}$. Then, we fine-tune the model on $\mathcal{Q}$. We repeat this two-step process for $T = 20$ iterations. The details of this algorithm are in Appendix F.

**Results.** As shown in Table 5, ASSD generates re-designs with the lowest binding affinities on average. By training ASSD with the MLE-RL objective (Equation (1)), we can reduce $p_{\text{rep}}$ by 16.67% while maintaining a similar level of performance. Note that this corresponds to 81.48% reduction in token repetition with better performance compared to MEAN, whose approximately half of its re-designed sequences are repetitive and

Table 6: **Average binding affinity of re-designed complexes from docked templates, measured in Rosetta Energy Units.** ASSD (MLE) designs antibodies with the highest binding affinities, while ASSD (MLE + RL) achieves similar performance with $p_{\text{rep}}$ reduced by 66.7%.

| | RefineGNN | MEAN | ASSD (MLE) | ASSD (MLE + RL) |
|---|---|---|---|---|
| Affinity (↓) | 3036.17 | 815.32 | **624.65** | 666.90 |
| $p_{\text{rep}}$(↓) (%) | 0.00 | 11.67 | 10.00 | **3.33** |

thus out-of-distribution. For better comparison, we also provide the effect of random mutation from Kong et al. (2022), denoted Random*.

### 5.4 CDR-H3 design with docked templates

Here, we explore a more challenging setting where both CDR-H3 and the binding complex are unknown. The goal of this task is to generate CDR-H3 designs with higher binding affinities in this challenging setting.

**Data.** We modify the RAbD test set from Section 5.2 to create docked templates. More concretely, we first segregate each antibody-antigen complex into a CDR-H3-removed antibody and antigen, which is then used to generate 10,000 docked templates with MEGADOCK (Ohue, 2023). Among them, we include the top-10 scoring docked templates in the test set.

**Evaluation.** We prepare the best validation checkpoints from Section 5.2 and generate CDR-H3 re-design for each docked template in the test set. We then evaluate the binding affinities of the re-designed complexes with pyRosetta (Chaudhury et al., 2010) including the side-chain packing. Considering the risk of docking inaccuracy, we report the binding affinities of the top-scoring re-design for each antibody-antigen complex following Kong et al. (2022).

**Results.** Table 6 shows our results. Although ASSD trained with regular MLE objectives performs the best, about 10% of its generated sequences are overly repetitive. By training the model with the MLE-RL objective, we reduce $p_{\text{rep}}$ to 3.33%, a level comparable to the autoregressive model RefineGNN.

## 6 Protein design experiments

In this section, we demonstrate that our training algorithm is generalizable to single-chain protein design with CATH 4.2 and CATH 4.3 benchmarks. Specifically, we train LM-Design (Zheng et al., 2023) with our objective at varying levels of $\alpha$ and report the sequence modeling performance with AAR and sequence validity with $p_{\text{rep}}$. As none of the sequences in CATH 4.2 and CATH 4.3 test sets contain over ten repeated tokens, we define a sequence with more than ten token repetitions as repetitive. We also report the result of ProteinMPNN (Dauparas et al., 2022) for comparison.

**Results.** Figure 4(b) shows our results on CATH 4.2 (*left*) and CATH 4.3 (*right*) benchmarks. Both LM-Design and ProteinMPNN, trained with their original objective, achieve high $p_{\text{rep}}$, indicating that a considerable portion of its sequence predictions are out-of-distribution. By increasing $\alpha$ in the MLE-RL objective (Equation (1)), we reduce the $p_{\text{rep}}$ while maintaining a decent level of sequence recovery. In particular, our objective allows achievement of $p_{\text{rep}}$ ∼0% for CATH 4.2 and ∼5% for CATH 4.3, while maintaining AAR similar to ProteinMPNN.

## 7 Ablations

### 7.1 Effect of sequence structure decoupling

Thus far, we have validated that the sequence-structure decoupling approach with ESM2-650M and MEAN outperforms other models in antibody benchmark experiments. Here, we demonstrate that our decoupling approach can improve the performance of other joint sequence-structure co-design models (i.e., RefineGNN,

Table 7: **Antibody benchmark results of the joint sequence-structure co-design models and their decoupled variants.** The decoupled variants improve both AAR and RMSD across all CDRs.

| | SAbDab | | RAbD | |
|---|---|---|---|---|
| | AAR (↑) (%) | RMSD (↓) | AAR (↑) (%) | RMSD (↓) |
| AR-GNN | 17.47 | 3.638 | 18.81 | 3.237 |
| AR-GNN* | **41.32** | **2.807** | **40.45** | **3.199** |
| RefineGNN | 23.49 | 2.173 | 22.43 | 1.716 |
| RefineGNN* | **41.32** | **2.000** | **40.45** | **1.581** |
| MEAN | 35.39 | 2.188 | 37.15 | 1.686 |
| MEAN* | **41.32** | **2.180** | **40.45** | **1.672** |

Table 8: **AAR, RMSD, and $p_{\mathbf{rep}}$ on CDR-H1 of SAbDab dataset.** ASSD still achieves the best performance without using ESM. This suggests that its performance is mainly driven by sequence-structure decoupling rather than data leakage of ESMs.

| | **Models** | CDR-H1 | | |
|---|---|---|---|---|
| | | AAR(↑) (%) | RMSD(↓) | $p_{\text{rep}}$(↓) (%) |
| AR | LSTM | 39.47±2.11 | - | 0.00±0.00 |
| | AR-GNN | 48.39±4.72 | 2.90±0.17 | 0.00±0.00 |
| | RefineGNN | 42.03±2.88 | 0.87±0.10 | 0.00±0.00 |
| NAR | MEAN | 59.32±4.71 | 0.91±0.10 | 0.29±0.49 |
| | ASSD (MLE) | **68.94±2.90** | **0.86±0.19** | **0.02±0.06** |
| | ASSD (MLE + RL) | 65.23±4.59 | 0.87±0.12 | **0.02±0.05** |

AR-GNN) by revisiting the benchmarks experiments in Section 5.1 (SAbDab) and Section 5.2 (RAbD) with a focus on CDR-H3 region.

**Baselines.** We select AR-GNN, RefineGNN, and MEAN as our baselines. We also create the decoupled variants of each model by using ESM2-650M for the sequence design model and each baseline as a structure prediction model. We denote each variant with an asterisk *.

**Training.** We train the baselines with the objectives proposed in their original papers (Jin et al., 2021; Kong et al., 2022). For the decoupled variants, we train ESM2-650M with the cross-entropy objective and each structure prediction model with the same objective as the baseline. To fully reveal the benefit of our approach, we train all models to their full capacity by setting the training epochs to 9999 and patience to 10.

**Results.** Table 7 shows that the decoupled variants outperform the original joint prediction models. This demonstrates the advantage of task-specific optimization with the ASSD framework.

### 7.2 Data leakage of ESMs

In this section, we address the potential concern regarding the influence of data leakage from protein language models on our performance. To address this concern, we repeat the experiment on the SAbDab benchmark (Section 5.1) by training the corresponding Transformer architecture from scratch.

**Results.** As shown in Table 8, Table 9, and Table 10, our approach still achieves the best results, excluding the baseline that uses ESM. This indicates that the high performance is mainly due to sequence-structure decoupling rather than data leakage. Our findings are consistent with Wu & Li (2024) and Wang et al. (2022), which show that protein language models alone are insufficient for antibody-related tasks.

Table 9: **AAR, RMSD, and $p_{\mathbf{rep}}$ on CDR-H2 of SAbDab dataset.** ASSD still achieves the best performance without using ESM. This suggests that its performance is mainly driven by sequence-structure decoupling rather than data leakage of ESMs.

| | **Models** | CDR-H2 | | |
|---|---|---|---|---|
| | | AAR(↑) (%) | RMSD(↓) | $p_{\text{rep}}$(↓) (%) |
| AR | LSTM | 30.41±2.47 | - | 0.00±0.00 |
| | AR-GNN | 38.03±2.81 | 2.30±0.16 | 0.00±0.00 |
| | RefineGNN | 32.05±2.23 | **0.79±0.06** | 0.00±0.00 |
| NAR | MEAN | 48.85±2.32 | 0.88±0.07 | 0.03±0.08 |
| | ASSD (MLE) | **60.39±3.53** | **0.79±0.08** | **0.02±0.07** |
| | ASSD (MLE + RL) | 60.23±4.76 | 0.83±0.07 | **0.02±0.07** |

Table 10: **AAR, RMSD, and $p_{\mathbf{rep}}$ on CDR-H3 of SAbDab dataset.** ASSD still achieves the best performance without using ESM. This suggests that its performance is mainly driven by sequence-structure decoupling rather than data leakage of ESMs.

| | **Models** | CDR-H3 | | |
|---|---|---|---|---|
| | | AAR(↑) (%) | RMSD(↓) | $p_{\text{rep}}$(↓) (%) |
| AR | LSTM | 15.82±1.63 | - | 0.00±0.00 |
| | AR-GNN | 18.72±0.82 | 3.60±0.58 | 0.03±0.07 |
| | RefineGNN | 24.44±1.94 | 2.24±0.14 | 0.11±0.13 |
| NAR | MEAN | 36.50±1.60 | **2.23±0.07** | 27.40±9.23 |
| | ASSD (MLE) | 37.44±1.48 | **2.23±0.07** | 25.32±12.2 |
| | ASSD (MLE + RL) | **37.84±1.21** | 2.24±0.10 | **21.57±14.78** |

## 8 Discussions

**Conclusion.** In this paper, we proposed antibody sequence-structure decoupling (ASSD), a general framework for highly performant sequence-structure co-design models, and a composition-based objective for reducing token repetitions for non-autoregressive models. Experiments on various benchmarks demonstrate that our approach achieves high sequence-structure modeling capacity with limited token repetitions compared to previous works.

**Structure-based drug design.** A key consideration is whether our approach overlooks the principles of structure-based drug design, given that our sequence design step does not explicitly condition on the structural information. However, recent wet lab experiments (Hie et al., 2024) have shown that pre-trained protein language models can design functional antibody sequences with high binding affinity without explicit structure modeling. Our work builds on this empirical finding, assuming that ESM has implicitly learned structural information, as evidenced in Hie et al. (2024) and Lin et al. (2023).

**Limitations and future work.** A limitation of our work is using AAR and RMSD to assess the antibody design models. Both metrics cannot measure the functional properties of the designed antibodies, which are important in real-life drug discovery pipelines. However, currently, there is no standardized metric for assessing the functional properties; thus, AAR and RMSD are the most widely used metrics for antibody/protein design tasks (Jin et al., 2021; Kong et al., 2022; 2023; Verma et al., 2023; Melnyk et al., 2023; Luo et al., 2022; Wu & Li, 2024). A related limitation is that the amino acid composition used to construct our composition-based objective is not a functional property. Therefore, promising future research would be developing and training antibody design models based on functional property evaluators.

## References

Jared Adolf-Bryfogle, Oleks Kalyuzhniy, Michael Kubitz, Brian D Weitzner, Xiaozhen Hu, Yumiko Adachi, William R Schief, and Roland L Dunbrack Jr. Rosettaantibodydesign (rabd): A general framework for computational antibody design. *PLoS computational biology*, 14(4):e1006112, 2018.

Rahmad Akbar, Philippe A Robert, Cédric R Weber, Michael Widrich, Robert Frank, Milena Pavlović, Lonneke Scheffer, Maria Chernigovskaya, Igor Snapkov, Andrei Slabodkin, et al. In silico proof of principle of machine learning-based antibody design at unconstrained scale. In *MAbs*, volume 14, pp. 2031482. Taylor & Francis, 2022.

Christian B Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 700–710, 2022.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

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

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. Sabdab: the structural antibody database. *Nucleic acids research*, 42(D1):D1140–D1146, 2014.

Nathan C Frey, Dan Berenberg, Karina Zadorozhny, Joseph Kleinhenz, Julien Lafrance-Vanasse, Isidro Hotzel, Yan Wu, Stephen Ra, Richard Bonneau, Kyunghyun Cho, et al. Protein discovery with discrete walk-jump sampling. In *The Twelfth International Conference on Learning Representations*, 2023.

C Fu, C Fu, and Michael Michael. *Handbook of simulation optimization*. Springer, 2015.

Zhangyang Gao, Cheng Tan, and Stan Z Li. Pifold: Toward effective and efficient protein inverse folding. In *The Eleventh International Conference on Learning Representations*, 2022.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6112–6121, 2019.

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

Brian L Hie, Varun R Shanker, Duo Xu, Theodora UJ Bruun, Payton A Weidenbacher, Shaogeng Tang, Wesley Wu, John E Pak, and Peter S Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2):275–283, 2024.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in neural information processing systems*, 32, 2019.

Justina Jankauskaitė, Brian Jiménez-García, Justas Dapkūnas, Juan Fernández-Recio, and Iain H Moal. Skempi 2.0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. *Bioinformatics*, 35(3):462–469, 2019.

Wengong Jin, Jeremy Wohlwend, Regina Barzilay, and Tommi S Jaakkola. Iterative refinement graph neural network for antibody sequence-structure co-design. In *International Conference on Learning Representations*, 2021.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020.

Jisun Kim, Matthew McFee, Qiao Fang, Osama Abdin, and Philip M Kim. Computational and artificial intelligence-based methods for antibody development. *Trends in Pharmacological Sciences*, 44(3):175–189, 2023.

Suyoun Kim, Duc Le, Weiyi Zheng, Tarun Singh, Abhinav Arora, Xiaoyu Zhai, Christian Fuegen, Ozlem Kalinli, and Michael L. Seltzer. Evaluating user perception of speech recognition system quality with semantic distance metric. In *Interspeech*, 2021.

Xiangzhe Kong, Wenbing Huang, and Yang Liu. Conditional antibody design as 3d equivariant graph translation. In *The Eleventh International Conference on Learning Representations*, 2022.

Xiangzhe Kong, Wenbing Huang, and Yang Liu. End-to-end full-atom antibody design. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 17409–17429, 2023.

Marie-Paule Lefranc, Christelle Pommié, Manuel Ruiz, Véronique Giudicelli, Elodie Foulquier, Lisa Truong, Valérie Thouvenin-Contet, and Gérard Lefranc. Imgt unique numbering for immunoglobulin and t cell receptor variable domains and ig superfamily v-like domains. *Developmental & Comparative Immunology*, 27(1):55–77, 2003.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Ge Liu, Haoyang Zeng, Jonas Mueller, Brandon Carter, Ziheng Wang, Jonas Schilz, Geraldine Horny, Michael E Birnbaum, Stefan Ewert, and David K Gifford. Antibody complementarity determining region design using high-capacity machine learning. *Bioinformatics*, 36(7):2126–2133, 2020.

Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems*, 35:9754–9767, 2022.

Karolis Martinkus, Jan Ludwiczak, WEI-CHING LIANG, Julien Lafrance-Vanasse, Isidro Hotzel, Arvind Rajpal, Yan Wu, Kyunghyun Cho, Richard Bonneau, Vladimir Gligorijevic, et al. Abdiffuser: full-atom generation of in-vitro functioning antibodies. *Advances in Neural Information Processing Systems*, 36, 2024.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2799–2808, 2019.

Igor Melnyk, Vijil Chenthamarakshan, Pin-Yu Chen, Payel Das, Amit Dhurandhar, Inkit Padhi, and Devleena Das. Reprogramming pretrained language models for antibody sequence infilling. In *International Conference on Machine Learning*, pp. 24398–24419. PMLR, 2023.

Asher Mullard. 2021 fda approvals. *Nat. Rev. Drug Discov.*, 21(2):83–88, 2022.

Tuomo K Nikula, Monica Bocchia, Michael J Curcio, George Sgouros, Yan Ma, Ronald D Finn, and David A Scheinberg. Impact of the high tyrosine fraction in complementarity determining regions: measured and predicted effects of radioiodination on igg immunoreactivity. *Molecular immunology*, 32(12):865–872, 1995.

Masahito Ohue. Megadock-on-colab: an easy-to-use protein–protein docking tool on google colaboratory. *BMC Research Notes*, 16(1):229, 2023.

Tobias H Olsen, Fergus Boyles, and Charlotte M Deane. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science*, 31(1): 141–146, 2022.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

Jeffrey A Ruffolo, Lee-Shin Chu, Sai Pooja Mahajan, and Jeffrey J Gray. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Nature communications*, 14(1):2389, 2023.

Koichiro Saka, Taro Kakuzaki, Shoichi Metsugi, Daiki Kashiwagi, Kenji Yoshida, Manabu Wada, Hiroyuki Tsunoda, and Reiji Teramoto. Antibody design using lstm based deep generative model from phage display library for affinity maturation. *Scientific reports*, 11(1):5852, 2021.

Sisi Shan, Shitong Luo, Ziqing Yang, Junxian Hong, Yufeng Su, Fan Ding, Lili Fu, Chenyu Li, Peng Chen, Jianzhu Ma, et al. Deep learning guided optimization of human antibody against sars-cov-2 variants with broad neutralization. *Proceedings of the National Academy of Sciences*, 119(11):e2122954119, 2022.

Bin Shi, Long Ma, Xiaoyan He, Xiaomei Wang, Peng Wang, Li Zhou, and Xinsheng Yao. Comparative analysis of human and mouse immunoglobulin variable heavy regions from imgt/ligm-db with imgt/highv-quest. *Theoretical Biology and Medical Modelling*, 11:1–11, 2014.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

Cheng Tan, Zhangyang Gao, Jun Xia, Bozhen Hu, and Stan Z Li. Global-context aware generative protein design. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

Yogesh Verma, Markus Heinonen, and Vikas Garg. Abode: Ab initio antibody design using conjoined odes. In *International Conference on Machine Learning*, pp. 35037–35050. PMLR, 2023.

Danqing Wang, YE Fei, and Hao Zhou. On pre-training language model for antibody. In *The Eleventh International Conference on Learning Representations*, 2022.

Xiaoling Wang, Tapan K Das, Satish K Singh, and Sandeep Kumar. Potential aggregation prone regions in biotherapeutics: a survey of commercial monoclonal antibodies. In *MAbs*, volume 1, pp. 254–267. Taylor & Francis, 2009.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

Fang Wu and Stan Z Li. A hierarchical training paradigm for antibody structure-sequence co-design. *Advances in Neural Information Processing Systems*, 36, 2024.

Jinrui Xu and Yang Zhang. How significant is a protein structure similarity with tm-score= 0.5? *Bioinformatics*, 26(7):889–895, 2010.

Kevin Yang, Wengong Jin, Kyle Swanson, Regina Barzilay, and Tommi Jaakkola. Improving molecular design by stochastic iterative target augmentation. In *International Conference on Machine Learning*, pp. 10716–10726. PMLR, 2020.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2019.

Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

Zaixiang Zheng, Yifan Deng, Dongyu Xue, Yi Zhou, Fei Ye, and Quanquan Gu. Structure-informed language models are protein designers. In *International Conference on Machine Learning*, pp. 42317–42338. PMLR, 2023.

Chunting Zhou, Jiatao Gu, and Graham Neubig. Understanding knowledge distillation in non-autoregressive machine translation. In *International Conference on Learning Representations*, 2019.

## A Proof for hypothetical example

| | | | | | |
|---|---|---|---|---|---|
| Y | Y | Y | G | D | R |
| Y | G | D | Y | Y | Y |
| D | Y | Y | R | A | G |
| G | Y | A | Y | Y | Y |

Figure 5: Sequence dataset for hypothetical example in Section 4.1.

We here provide proof for the hypothetical example in Section 4.1. Following from Section 4.1, the MLE objective is given by:

$$\mathcal{L}_{\text{MLE}}(\theta) = \sum_{i=1}^{L} \sum_{n=1}^{N} \ell_{\text{ce}}(s_i^{(n)}, \boldsymbol{p}_i)$$

where $\boldsymbol{p} = (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_L) = f_\theta(\boldsymbol{c})$ is the head output distributions. We can re-parameterize the objective for each $\boldsymbol{p}_i$ as:

$$\mathcal{L}_{\text{MLE}}(\boldsymbol{p}_i) = \sum_{n=1}^{N} \ell_{\text{ce}}(s_i^{(n)}, \boldsymbol{p}_i)$$

where $\sum \boldsymbol{p}_i = 1$ is the constraint. We then compute the optimal $\boldsymbol{p}_i^*$ subject to this constraint with Sequential Least Squares Programming (SLSQP):

$$\begin{aligned}
\boldsymbol{p}_1^* &= \left(0, \frac{\mathbf{2}}{\mathbf{4}}, \frac{1}{4}, \frac{1}{4}, 0\right) \\
\boldsymbol{p}_2^* &= \left(0, \frac{\mathbf{3}}{\mathbf{4}}, \frac{1}{4}, 0, 0\right) \\
\boldsymbol{p}_3^* &= \left(\frac{1}{4}, \frac{\mathbf{2}}{\mathbf{4}}, 0, \frac{1}{4}, 0\right) \\
\boldsymbol{p}_4^* &= \left(0, \frac{\mathbf{2}}{\mathbf{4}}, \frac{1}{4}, 0, \frac{1}{4}\right) \\
\boldsymbol{p}_5^* &= \left(\frac{1}{4}, \frac{\mathbf{2}}{\mathbf{4}}, 0, \frac{1}{4}, 0\right) \\
\boldsymbol{p}_6^* &= \left(0, \frac{\mathbf{2}}{\mathbf{4}}, \frac{1}{4}, 0, \frac{1}{4}\right)
\end{aligned}$$

where $\boldsymbol{p}_i = (p_A, p_Y, p_G, p_D, p_R)$ respectively. Since $p_{\theta^*}(\boldsymbol{s}|\boldsymbol{c}) = \prod_{i=1}^{6} \text{Cat}(s_i|\boldsymbol{p}_i^*)$, the sequence with highest likelihood is `YYYYYY`.

## B Derivation of objective $\mathcal{L}_{\text{comp}}(\theta)$

We here derive a differentiable objective function of $\mathcal{L}_{\text{comp}}(\theta) = \mathbb{E}_{\hat{\boldsymbol{s}} \sim p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})}\left[d(\boldsymbol{s}, \hat{\boldsymbol{s}})\right]$ using the REINFORCE estimator (Fu et al., 2015). Consider a sample $(\boldsymbol{s}, \boldsymbol{c})$ from the dataset.

$$
\begin{aligned}
\nabla_\theta \mathcal{L}_{\text{comp}}(\theta) &= \nabla_\theta \mathbb{E}_{\hat{\boldsymbol{s}} \sim p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})}\left[d(\boldsymbol{s}, \hat{\boldsymbol{s}})\right] \\
&= \nabla_\theta \sum_{\hat{\boldsymbol{s}}} p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c}) d(\boldsymbol{s}, \hat{\boldsymbol{s}}) \\
&= \sum_{\hat{\boldsymbol{s}}} \left[\nabla_\theta p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})\right] d(\boldsymbol{s}, \hat{\boldsymbol{s}}) \\
&= \sum_{\hat{\boldsymbol{s}}} p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c}) \nabla_\theta \log p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c}) d(\boldsymbol{s}, \hat{\boldsymbol{s}}) \\
&= \mathbb{E}_{p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})}\left[d(\boldsymbol{s}, \hat{\boldsymbol{s}}) \nabla_\theta \log p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})\right] \\
&= \mathbb{E}_{p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})}\left[(d(\boldsymbol{s}, \hat{\boldsymbol{s}}) - b) \nabla_\theta \log p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c}) + b \nabla_\theta \log p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})\right] \\
&= \mathbb{E}_{p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})}\left[(d(\boldsymbol{s}, \hat{\boldsymbol{s}}) - b) \nabla_\theta \log p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})\right]
\end{aligned}
$$

where the last line follows from $\mathbb{E}_{p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})}\left[\nabla_\theta \log p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})\right] = 0$. With a single-sample Monte Carlo approximation, we have

$$
\nabla_\theta \mathcal{L}_{\text{comp}}(\theta) \approx \left[d(\boldsymbol{s}, \hat{\boldsymbol{s}}) - b\right] \nabla_\theta \log p_\theta(\hat{\boldsymbol{s}}|\boldsymbol{c})
$$

## C Training details

**Hyperparameters.** Table 11 and Table 12 shows the hyperparameters of baselines and ESM2-650M, respectively. For baselines, we follow the training settings in their original papers (Jin et al., 2021; Kong et al., 2022), and for ESM2-650M, we limit the batch size with a maximum token length of 6000 and train for 30 epochs. We set hyperparameter $\alpha = 0.2$ in our sequence objective and use rank 2 for weights $W_q, W_k, W_v$, and $W_o$ in the multi-head attention module for LoRA fine-tuning.

**Implementation for LM-Design.** In our paper, we adopt LM-Design (Zheng et al., 2023) for antibody design. LM-Design is originally a pLM-based protein design model $p_\theta(\boldsymbol{s}|\boldsymbol{x})$ where $\boldsymbol{x}$ is the protein structure and $\boldsymbol{s}$ is the corresponding sequence that folds into this structure. For antibody design tasks, we maintain the original architecture, yet use antigen structure as $\boldsymbol{x}$ and initialize $\boldsymbol{s}$ with framework/antigen sequence information.

**Code.** Our implementation is built upon `https://github.com/facebookresearch/esm`, `https://github.com/BytedProtein/ByProt/tree/main`, `https://github.com/wengong-jin/RefineGNN`, and `https://github.com/THUNLP-MT/MEAN/tree/main`. We deeply appreciate the authors (Lin et al., 2023; Zheng et al., 2023; Jin et al., 2021; Kong et al., 2022) for their contributions to our project.

**Machine specs.** All models were trained on a machine with 48 CPU cores and 8 NVIDIA Geforce RTX 3090. We have used 1-3 GPUs for all experiments.

Table 11: Hyperparameters for baselines.

| | LSTM | AR-GNN | RefineGNN | MEAN |
|---|---|---|---|---|
| Vocab size | | 25 | | |
| Dropout | | 0.1 | | |
| Hidden dim | | 256 | | 128 |
| Number of layers | 4 | 4 | 4 | 3 |
| K neighbors | - | 9 | 9 | - |
| Block size | - | - | 4 | - |
| Number of RBF kernels | - | 16 | 16 | - |
| Embed dim | - | - | - | 64 |
| Alpha | - | - | - | 0.8 |
| Number of iterations | - | - | - | 3 |
| Optimizer | | Adam | | |
| Learning rate | | 0.001 | | |

Table 12: Hyperparameters for ESM2-650M used as the sequence design model.

| | esm2_t33_650M_UR50D |
|---|---|
| Vocab size | 33 |
| Embed dim | 1280 |
| Number of layers | 33 |
| Optimizer | AdamW |
| Learning rate | 0.001 |

## D Mitigating exposure bias

As stated in Section 4.3, we address exposure bias by using the generated sequence as the input to the structure prediction model during training. We here demonstrate the benefit of this approach by comparing the performance of the structure prediction model trained with generated sequence (denoted $\hat{\boldsymbol{s}}$) and ground-truth sequence (denoted $\boldsymbol{s}$). Specifically, we train the structure prediction model on SAbDab (fold 0) and RAbD benchmark, then compare their RMSD and TM-score.

Table 13: RMSD and TM-score of structure prediction model trained with generated sequence as input ($\hat{\boldsymbol{s}}$) versus ground-truth sequence as input ($\boldsymbol{s}$). Training the structure prediction model with the generated sequence outperforms its counterpart.

| | CDR-H1 (SAbDab) | | CDR-H2 (SAbDab) | | CDR-H3 (SAbDab) | | CDR-H3 (RAbD) | |
|---|---|---|---|---|---|---|---|---|
| | RMSD(↓) | TM-score(↑) | RMSD(↓) | TM-score(↑) | RMSD(↓) | TM-score(↑) | RMSD(↓) | TM-score(↑) |
| $\hat{\boldsymbol{s}}$ | **0.7511** | **0.9901** | **0.7993** | **0.9910** | **2.2094** | **0.9748** | **1.78** | **0.9825** |
| $\boldsymbol{s}$ | 0.7077 | 0.9892 | 0.8825 | 0.9895 | 2.4743 | 0.9695 | 1.86 | 0.9789 |

## E Data statistics for SAbDab

In Table 14, we detail the number of antibodies in each fold for 10-fold cross-validation in Section 5.1. Following Kong et al. (2022), $\forall i = 0, \ldots, 9$, we let fold $i$ be the test set, fold $i-1$ be the validation set, and the remaining folds to be the train set.

Table 14: Number of antibodies in each fold for Sequence and Structure Modeling (Section 5.1)

| | CDR-H1 | CDR-H2 | CDR-H3 |
|---|---|---|---|
| fold 0 | 539 | 425 | 390 |
| fold 1 | 529 | 366 | 444 |
| fold 2 | 269 | 363 | 369 |
| fold 3 | 391 | 380 | 389 |
| fold 4 | 305 | 392 | 420 |
| fold 5 | 523 | 381 | 413 |
| fold 6 | 294 | 409 | 403 |
| fold 7 | 310 | 369 | 327 |
| fold 8 | 472 | 489 | 355 |
| fold 9 | 345 | 403 | 467 |
| **Total** | 3977 | 3977 | 3977 |

## F ITA algorithm for affinity optimization

**Criterion for validity.** Following (Jin et al., 2021), we force all generated sequences to satisfy the following constraints: (1) net charge must be in the range $[-2.0, 2.0]$, (2) sequence must not contain N-X-S/T motif, (3) a token should not repeat more than five times, (4) perplexity of the sequence must be below 10.

Algorithm 3 details the ITA algorithm in Section 5.3, where we have followed Kong et al. (2022) for the implementation.

**Algorithm 3** ITA algorithm for antibody affinity optimization

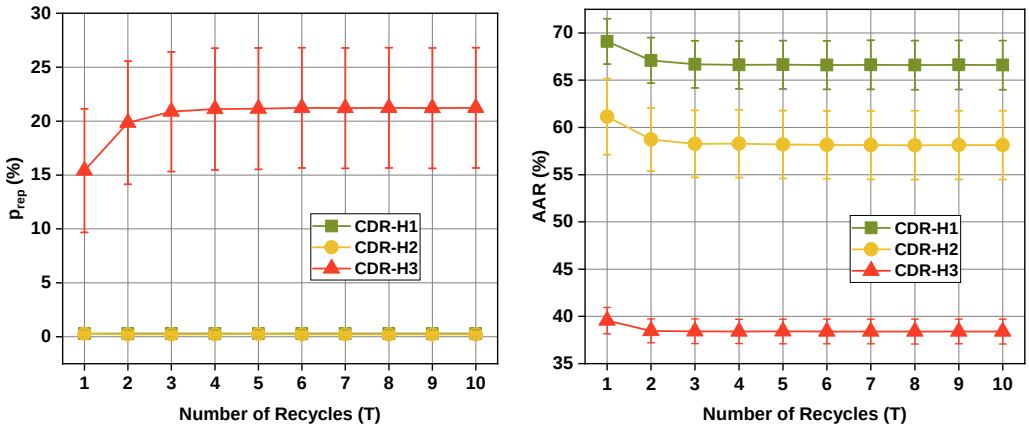
**Input:** SKEMPI V2.0 antibody-antigen complex dataset $\mathcal{D}$, pre-trained $p_\theta(\boldsymbol{s}, \boldsymbol{x}|\boldsymbol{c})$, top-$k$ candidates to maintain

Initialize $\mathcal{Q} \leftarrow \mathcal{D}$
**for** $t = 1, \ldots, T$ **do**
  **for** $(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{c}) \in \mathcal{D}$ **do**
    Initialize $\mathcal{C} \leftarrow \emptyset$
    **for** $i = 1, \ldots, M$ **do**
      $(\hat{\boldsymbol{s}}_i, \hat{\boldsymbol{x}}_i) \sim p_\theta(\boldsymbol{s}, \boldsymbol{x}|\boldsymbol{c})$
      **if** valid$(\hat{\boldsymbol{s}}_i, \hat{\boldsymbol{x}}_i)$ and $f((\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{c}), (\hat{\boldsymbol{s}}_i, \hat{\boldsymbol{x}}_i, \boldsymbol{c})) < 0$ **then**
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{((\hat{\boldsymbol{s}}_i, \hat{\boldsymbol{x}}_i, \boldsymbol{c}), f((\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{c}), (\hat{\boldsymbol{s}}_i, \hat{\boldsymbol{x}}_i, \boldsymbol{c})))\}$
      **end if**
    **end for**
    $\mathcal{Q}[(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{c})] \leftarrow \mathcal{Q}[(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{c})] \cup \mathcal{C}$
    Select top-$k$ elements in $\mathcal{Q}[(\boldsymbol{s}, \boldsymbol{x}, \boldsymbol{c})]$ by $f(\cdot, \cdot)$
  **end for**
  **for** $n = 1, \ldots, N$ **do**
    Select batch $\mathcal{B}$ from $\mathcal{Q}$
    Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\mathcal{B}, \theta)$
  **end for**
**end for**

## G Effect of sample recycling on sequence quality

A common NLP approach for reducing token repetition of non-autoregressive models is recycling tokens for $T$ iterations (Ghazvininejad et al., 2019; Zhou et al., 2019). We here demonstrate that this approach is not effective for antibody design.

**Results.** As shown in Figure 6(a), sample recycling does not reduce token repetition of non-autoregressive models on antibody modeling tasks. It even increases $p_{\text{rep}}$ for CDR-H3 from about 15% to 20%. Also, Figure 6(b) shows that sample recycling decreases AARs for all CDR regions. Contrary to this approach, our composition-based objective effectively reduces $p_{\text{rep}}$ while maintaining high AAR, as demonstrated in Section 5.




(a) (b)

Figure 6: **Effect of sample recycling on $p_{\text{rep}}$ and AAR.** (a) Sample recycling does not reduce token repetition for antibody design tasks. (b) Sample recycling even reduces AAR across all CDRs.