# IgBlend: Unifying 3D Structures and Sequences in Antibody Language Models

**Cedric Malherbe** **Talip Uçar**

Centre for AI, DS&AI, BioPharmaceuticals R&D, AstraZeneca
{cedric.malherbe, talip.ucar}@astrazeneca.com

## Abstract

Large language models (LLMs) trained on antibody sequences have shown significant potential in the rapidly advancing field of machine learning-assisted antibody engineering and drug discovery. However, current state-of-the-art antibody LLMs often overlook structural information, which could enable the model to more effectively learn the functional properties of antibodies by providing richer, more informative data. In response to this limitation, we introduce IgBlend, which integrates both the 3D coordinates of backbone atoms (C-alpha, N, and C) and antibody sequences. Our model is trained on a diverse dataset containing over 4 million unique structures and more than 200 million unique sequences, including heavy and light chains as well as nanobodies. We rigorously evaluate IgBlend using established benchmarks such as sequence recovery, complementarity-determining region (CDR) editing and inverse folding and demonstrate that IgBlend consistently outperforms current state-of-the-art models across all benchmarks. Furthermore, experimental validation shows that the model's log probabilities correlate well with measured binding affinities.

## 1 Introduction

Antibodies are key components of the adaptive immune system, capable of recognizing and neutralizing a wide range of pathogens, including viruses, bacteria, and other foreign invaders. Their ability to bind specific targets with high affinity makes them essential tools in therapeutic development. Recent advancements in natural language processing (NLP) have led to the creation of foundational language models that can learn from and modify antibody sequences [Olsen et al., 2022b, 2024, Prihoda et al., 2022]. Moreover, the three-dimensional (3D) structure of an antibody is closely linked to its specificity, affinity, and interaction with antigens. Therefore, capturing the relationship between sequence and structure is crucial for tasks such as affinity maturation, de novo antibody design, and optimizing antibody-antigen interactions for therapeutic applications. While current language models excel at either sequence-to-sequence or structure-to-sequence (inverse folding) tasks, relying on only one of these modalities at the input limits their capability and flexibility in more complex antibody engineering tasks [Olsen et al., 2022b, 2024, Prihoda et al., 2022, Høie et al., 2023]. In this paper, we introduce IgBlend, a multi-modal model designed to incorporate both sequence and structural information for antibody engineering. Our approach can utilize either sequence, structure, or both, enabling the model to not only sample sequences that can fold to the same parental backbone but also generate more diverse sequences, providing greater flexibility in designing antibody sequences. Moreover, by utilizing both experimentally resolved structures [Dunbar et al., 2014] and synthetic data generated through structure prediction models [Abanades et al., 2023b, Ruffolo et al., 2023], we aim to improve model performance on key antibody engineering tasks. Our contributions can be summarized as follows:

AI for New Drug Modalities at NeurIPS 2024.

- We introduce IgBlend, a model that learns antibody representations from either sequence, structure or sequence-structure pairs when structural data is available.
- We present a pre-training strategy with multiple sub-objectives as well as a procedure for training and dataset processing, all of which can broadly be applied to other multi-modal training settings.
- We empirically demonstrate that integrating structural information, even when synthetically generated, significantly improves the performance of large models across a wide range of benchmarks.

The remainder of the paper is organized as follows. First, we discuss related works and introduce the notations. Section 2 details the architecture of IgBlend, datasets, and training procedures. Then, we compare the performance of IgBlend against existing models in Section 3. Lastly, we point out that a more detailed background on antibodies can be found in Appendix A.

**Related work.** In recent years, significant progress has been made in developing protein and antibody foundation models, drawing from advances in natural language processing (NLP). These models can be broadly categorized based on their focus—either on general protein design or antibody-specific tasks—and the way they approach the problem, such as sequence-to-structure prediction, structure-to-sequence generation, or sequence-structure co-design. For general protein design, sequence-to-structure models, including AlphaFold [Jumper et al., 2021] and RoseTTAFold [Baek et al., 2021], have significantly improved the prediction of protein structures from amino acid sequences. In the structure-to-sequence domain (inverse folding), ESM-IF [Hsu et al., 2022] predicts amino acid sequences that fold into a given structure. Meanwhile, sequence-to-sequence models, such as ESM [Rives et al., 2021] and its variants, excel at identifying patterns within sequences for tasks such as sequence recovery and mutation prediction. Moreover, some recent works have focused on sequence-structure co-design, with models such as ESM3 [Hayes et al., 2024] incorporating both sequence and structure as well as function to improve protein design. There is also hybrid approaches such as LM-Design [Zheng et al., 2023] that leverage both sequence and structural inputs to design new protein sequences, aligning with the approach used in this paper. In the context of antibodies, several models have emerged with a similar framework but tailored for immunoglobulins. For sequence-to-structure prediction, antibody-specific models such as ImmuneBuilder [Abanades et al., 2023b] and IgFold [Ruffolo et al., 2023] predict 3D antibody structures from sequence data. In the structure-to-sequence domain, AntiFold [Høie et al., 2023] addresses the inverse folding problem, predicting antibody sequences that correspond to a given backbone structure. For sequence-to-sequence tasks, models such as AbLang [Olsen et al., 2022b], AbLang-2 [Olsen et al., 2024], AntiBERTy [Ruffolo et al., 2021], and Sapiens [Prihoda et al., 2022], which are predominantly based on the BERT architecture [Devlin et al., 2018], are specifically designed for antibody sequences, helping to improve performance on tasks such as residue restoration and paratope identification. To the best of our knowledge, although these models focus on either sequence or structure, no existing antibody-specific LLM effectively integrates both modalities. In this paper, we address this gap by introducing a sequence-structure-to-sequence framework, similar to LM-Design [Zheng et al., 2023] for proteins, to learn a richer and more informative representation. IgBlend learns joint representations of structure and sequence during pre-training, improving upon models that rely on

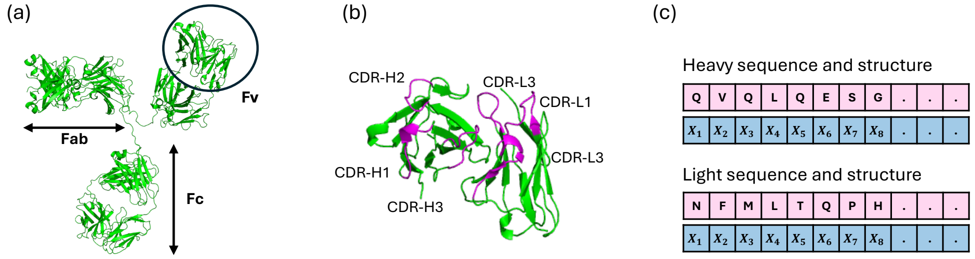

(a)
Fv
Fab
Fc
(b)
CDR-H2
CDR-L3
CDR-L1
CDR-L3
CDR-H1
CDR-H3
(c)
Heavy sequence and structure
Q V Q L Q E S G . . .
$x_1$ $x_2$ $x_3$ $x_4$ $x_5$ $x_6$ $x_7$ $x_8$ . . .
Light sequence and structure
N F M L T Q P H . . .
$x_1$ $x_2$ $x_3$ $x_4$ $x_5$ $x_6$ $x_7$ $x_8$ . . .

Figure 1: (a) Antibody structure with antigen binding (Fab), crystallizable (Fc), and variable (Fv) regions, (b) Zoom over the variable region which contains an heavy and a light chain, CDRs regions are displayed in magenta, (C) Modalities that we exploit in this paper for antibody modeling.

a single modality. Finally, we leave out diffusion, flow matching, and graph-based approaches to antibody design to maintain our focus on language models.

**Notations.** For any single unpaired chain (heavy, light or nanobody), we denote the backbone structure and sequence of the chain with $n$ residues as follows:

$$\text{structure: } \mathbf{x} := (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{3\times3\times n} \text{ and sequence: } \mathbf{s} := (\mathbf{s}_1, \dots, \mathbf{s}_n) \in \mathbb{A}^n$$

where $\mathbf{x}_i \in \mathbb{R}^{3\times3}$ represents the 3D coordinates of the C-alpha, N, and C atoms of the $i^{th}$ residue, while $\mathbf{s}_i \in \mathbb{A} :=$ [A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V, *] specifies the amino acid type corresponding to the $i^{th}$ residue, where $i \in \{1, \dots, n\}$. For consistency in notation, we will also use $*$ to denote the unknown token for both structure and sequence tokens, acknowledging a slight abuse of notation. Moreover, we stress that this work solely focuses on unpaired sequences and leaves the fine-tuning on purely paired sequences for future work. In the remainder of this paper, we will use $\mathbb{P}$, $\mathbb{E}$, and $\mathbb{I}$ to denote the standard probability, expectation, and indicator functions, with $\mathbb{I}$ specifically taking values in $\{0, 1\}$. To compute the differences between two sequences $(\mathbf{s}, \widehat{\mathbf{s}}) \in \mathbb{A}^{|\mathbf{s}|}$ of the same length, we will use the normalized Levenshtein distance: $\text{Levenshtein}(\mathbf{s}, \widehat{\mathbf{s}}) = (1/|\mathbf{s}|) \cdot \sum_{i=1}^{|\mathbf{s}|} \mathbb{I}\{\mathbf{s}_i \neq \widehat{\mathbf{s}}_i\}$. To compute differences between two backbone structures $(\mathbf{x}, \widehat{\mathbf{x}}) \in \mathbb{R}^{3\times3\times|\mathbf{x}|}$, we will use the Root Mean Square Deviation (RMSD) defined as $\text{RMSD}(\mathbf{x}, \widehat{\mathbf{x}}) = \arg\min_{R\in\Omega_3, t\in\mathbb{R}^3} (1/3|\mathbf{x}|) \cdot \sum_{i\leq|\mathbf{x}|, j\leq3} \|\mathbf{x}_i^j - R^*\widehat{\mathbf{x}}_i^j - t^*\|_2^2)^{1/2}$ where $R^* \in \mathbb{R}^{3\times3}$ and $t^* \in \mathbb{R}^3$ respectively denote the optimal rotation matrix and translation after finding the optimal rigid alignment with the Kabsch algorithm [Kabsch, 1976] between the backbone structures where $\Omega_3 \subset \mathbb{R}^{3\times3}$ denotes the set of 3D rotations and $\|\cdot\|_2$ denotes the standard Euclidean distance.

## 2 Methods

In this section, we present the IgBlend architecture, the dataset, and the pre-training objectives.

### 2.1 Model architecture

The proposed architecture, IgBlend, is illustrated in Fig 2 and consists of three primary components: a structure encoder that handles the backbone coordinates of the antibody, a sequence encoder that processes the amino acid sequence, and a multi-modal trunk that processes both structural and sequential representations.

**Structure encoder.** It generates an abstract representation vector for each set of coordinates $\mathbf{x}_i \in \mathbb{R}^{3\times3}$ from the full sequence of coordinates $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{3\times3\times n}$. This representation (a 512-dimensional embedding) encapsulates the geometry of the global backbone structure. The architecture comprises four GVP-GNN (Graph Neural Network Geometric Vector Perceptron) layers [Jing et al., 2020], followed by two generic Transformer encoder layers [Vaswani et al., 2017]. This

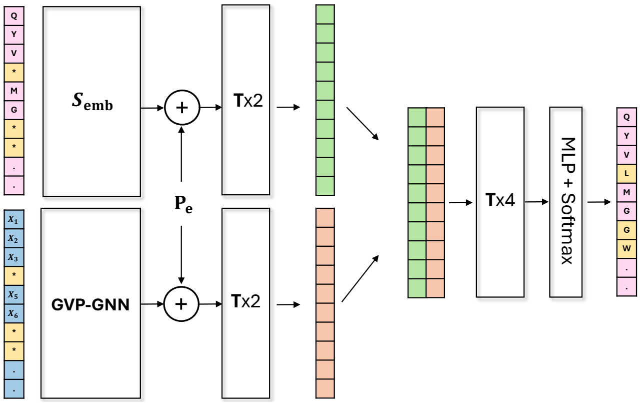


Figure 2: Architecture of the IgBlendmodel. It takes as input both: a series of amino acids (top) and a series of 3D coordinates (bottom). The symbol * denotes either a masked amino acid or a masked set of coordinates. Note that the model can process each modality independently by setting all the tokens of one modality to mask. $S_e$ denotes the sequence embedding (i.e. look-up table), $T$ denotes a transformer block, $P_e$ denotes the sinusoidal position embedding. The sequence encoder is displayed of the bottom left, the structure encoder on the bottom left and on the right the multi-modality processor.

design is invariant to rotation and translation of the input coordinates and has been demonstrated to effectively capture protein geometries in various learning tasks [Jing et al., 2020], including structure-to-sequence models such as ESM-inverse folding [Hsu et al., 2022] and AntiFold [Høie et al., 2023]. The input to the encoder consists of a series of residue coordinates $\mathbf{x}$, with a local reference frame established for each amino acid, as per the approach used in AlphaFold2 [Jumper et al., 2021]. A change of basis is then performed according to this local reference frame, rotating the vector features from the GVP-GNN outputs into the local reference frame of each amino acid. Finally, the output of the GVP is processed through two Transformer blocks, producing a 512-dimensional embedding for each residue. Notably, each or all sets of coordinates can be masked using the $*$ token.

**Sequence encoder.** In parallel to the structure encoder, the sequence encoder generates a vector representation (i.e., embedding of size 512) for each amino acid $\mathbf{s}_i \in \mathbb{A}$ in the full sequence $\mathbf{s} = (\mathbf{s_1}, \ldots, \mathbf{s_n})$. The architecture includes a one-hot encoded input followed by two blocks of a standard Transformer model [Vaswani et al., 2017]. This architecture has already been shown to learn relevant information from antibody sequences [Olsen et al., 2022b, 2024]. Specifically, the module utilizes sinusoidal positional embeddings, a SwiGLU activation function [Shazeer, 2020], and has an embedding dimension of 512. Additionally, any amino acid within the sequences can be masked using the masked token $*$.

**Multi-modality encoder.** The fusion layer processes both modalities in two steps. First, it combines the abstract representations from the sequence and structure encoders by concatenating them along the embedding dimension, forming a single vector of size 1024 for each residue. It then processes the concatenated modalities through a series of four Transformer blocks with SwiGLU activations.

**Classification head.** Finally, the classification head consists of a multi-layer perceptron (MLP) followed by a softmax function and processes the multi-modal representation to generate a probability distribution over amino acid types at each position. Further details can be found in Appendix B.

### 2.2 Data preparation

To create a model capable of processing both sequential and structural information, we compiled (1) a structural dataset $\mathcal{D}_{\text{struct}} = \{(\mathbf{s}, \mathbf{x})_1, \ldots, (\mathbf{s}, \mathbf{x})_{|\mathcal{D}_{\text{struct}}|}\}$, which includes structures paired with their corresponding sequences, and (2) a sequential dataset $\mathcal{D}_{\text{seq}} = \{(s, *)_1, \ldots, (s, *)_{|\mathcal{D}_{\text{seq}}|}\}$, which consists solely of sequence data. These datasets were derived from four sources: SAbDab [Dunbar et al., 2014]; PLAbDab [Abanades et al., 2023a]; OAS datasets [Olsen et al., 2022a]; and INDI [Deszyński et al., 2022] fully described in the Appendix C and summarized in Table 1. The datasets are further divided into samples for heavy chains, light chains, and nanobodies, resulting in $\mathcal{D}_{\text{struct}} = \mathcal{D}_{\text{struct,H}} \cup \mathcal{D}_{\text{struct,L}} \cup \mathcal{D}_{\text{struct,N}}$ and $\mathcal{D}_{\text{seq}} = \mathcal{D}_{\text{seq,H}} \cup \mathcal{D}_{\text{seq,L}} \cup \mathcal{D}_{\text{seq,N}}$. Due to the significant imbalance in the number of samples across modalities, as noted in Table 1, we implemented a new sampling scheme to rebalance the data. For each modality $\text{M} \in \{\text{seq}, \text{struct}\}$ and each chain type $\text{C} \in \{\text{L, H, N}\}$, we clustered the datasets $\mathcal{D}_{\text{M,C}}$ using MMseqs2 [Steinegger and Söding, 2017], clustering over the full sequences with the parameters "−cov-mode 1", "−c 0.8", and "−min_seq_id 0.8" for the sequential datasets and over the concatenated CDR regions with the parameter "−min_seq_id 0.9" for the structure datasets. This process resulted in a union of $n_{\text{cluster}}$ clustered samples $\mathcal{D}_{\text{M,C}} = \bigcup_{i=1}^{n_{\text{cluster}}} \mathcal{C}_{\text{M,C}}(i)$ for each modality and chain type. Based on these clusters, we defined the distributions $\mathcal{P}(\mathcal{D}_{\text{struct}})$ and $\mathcal{P}(\mathcal{D}_{\text{seq}})$ over each dataset modality as follows: first, we sample a chain type C with equal probability: $\mathbb{P}(\text{C = H}) = \mathbb{P}(\text{C = L}) = \mathbb{P}(\text{C = N}) = 1/3$, then we select a sample within the corresponding dataset $\mathcal{D}_{\text{C,M}}$ according to the size of its corresponding cluster:

| Modality | Heavy sequences | Light sequences | Heavy structures | Light structures |
|---|---|---|---|---|
| OAS paired | 1 804 122 | 443 129 | 1 418 312 | 535 130 |
| OAS unpaired | 156 314 998 | 34 464 420 | 1 057 850 | 643 647 |
| PLAbDab paired | 51 740 | 45 620 | 47 554 | 42 021 |
| PLAbDab unpaired | 139 706 | 89 743 | - | - |
| INDI (nanobodies) | 11 231 660 | - | 895 008 | - |
| SAbDab | - | - | 2 056 | 2 024 |
| Total | 169 542 226 | 35 042 912 | 3 420 780 | 1 222 822 |

Table 1: Number of unique samples per modalities and chain types after the first pre-processing step.

$$\mathbb{P}(\mathbf{s},\mathbf{x})_{|\text{M,C}} = \begin{cases} 1/|\mathcal{C}_{\text{M,C}}(i_s)| & \text{if } (\mathbf{s},\mathbf{x}) \in \mathcal{D}_{\text{M,C}} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $i_s$ denotes the index of the cluster containing $\mathbf{s}$, and $|\mathcal{C}_{\text{M,C}}(i_s)|$ indicates the size of its corresponding cluster. This clustering-based distribution approach enables us to preserve the entire dataset while re-weighting each cluster to improve diversity in the training set. After clustering, 10% of the clusters are set aside for validation, and another 10% are reserved for testing. Both sets are completely excluded from the training set and have less than 0.8 sequence identity with the training data, ensuring that the validation and test sets are sufficiently distinct from the training set.

### 2.3 Pre-training objectives

To train IgBlend, the data distribution defined by Equation (1) was used, ensuring a balanced representation of heavy, light and nanobodies chains across the two datasets, $\mathcal{D}_{\text{seq}}$ and $\mathcal{D}_{\text{struct}}$. We employ a specialized masked language modeling objective capable of handling both sequential and structural data. The model parameters, $\theta$, are optimized by minimizing the sum of three losses based on cross-entropy:

$$\mathcal{L}_{\text{multi-modal}} := \mathcal{L}_{\text{seq2seq}} + \mathcal{L}_{\text{seq+struct2seq}} + \mathcal{L}_{\text{struct2seq}} \tag{2}$$

where:

$$\begin{cases} \mathcal{L}_{\text{seq2seq}} & = & \mathbb{E}_{(\mathbf{s},*)\sim\mathcal{P}(\mathcal{D}_{\text{seq}})}\left[\sum_{i\in\mathcal{T}_s} -\log(p_\theta(s_i|\mathbf{s}_{/\mathcal{M}_s}, *)\right] \\ \mathcal{L}_{\text{seq+struct2seq}} & = & \mathbb{E}_{(\mathbf{s},\mathbf{x})\sim\mathcal{P}(\mathcal{D}_{\text{struct}})}\left[\sum_{i\in\mathcal{T}_s} -\log(p_\theta(s_i|\mathbf{s}_{/\mathcal{M}_s}, \mathbf{x}_{/\mathcal{M}_x})\right] \\ \mathcal{L}_{\text{struct2seq}} & = & \mathbb{E}_{(\mathbf{s},\mathbf{x})\sim\mathcal{P}(\mathcal{D}_{\text{struct}})}\left[\sum_{i\in\mathcal{T}_s} -\log(p_\theta(s_i|*, \mathbf{x}_{/\mathcal{M}_x})\right] \end{cases} \tag{3}$$

with $p_\theta(\mathbf{s}_i|\mathbf{s},\mathbf{x})$ denoting the output of the softmax layer shown in Figure 2 at position $i \in \{1,\ldots,n\}$, given $(\mathbf{s},\mathbf{x})$ as input. By using this combination, the model dedicates equal time on each task. The masking strategy for each pre-training objective is outlined below, defining the positions of the amino acids to predict $\mathcal{T}_s$, the masked residues in the sequence $\mathcal{M}_s$, and the masked structures $\mathcal{M}_x$:

- **seq2seq.** This task, used in training sequence-only antibody models [Devlin et al., 2018, Olsen et al., 2024], is applied to the sequential dataset $\mathcal{D}_{\text{seq}}$, which lacks structural information (i.e., $\mathbf{x} = *$). For each sequence, between 10 and 40 of the amino acids are selected for masking using one of two methods: (i) randomly sampling individual residues throughout the sequence or (ii) masking continuous spans of residues, with the starting position chosen at random. The positions of the residues to be predicted are the same as those masked, $\mathcal{M}_s = \mathcal{T}_s$. The masked residues in $\mathcal{M}_s$ are then processed using one of three strategies: (a) replaced by the unknown token $*$ with 80% probability, (b) substituted with a different amino acid with 10% probability, or (c) left unchanged with 10% probability. The masking distribution is also slightly adjusted to ensure balanced coverage of both CDR and framework regions.
- **seq+struct2seq.** Both sequential and structural information are used to predict masked amino acids, with masking applied to both the structure and sequence simultaneously. The same residues are used for both prediction and masking, with $\mathcal{T}_s = \mathcal{M}_x$. Following the seq2seq approach, 10 to 40 of the amino acids are masked, using a mix of continuous spans and random positions. With equal probability, we either (i) mask the corresponding coordinates $\mathcal{M}_x = \mathcal{M}_s$ or (ii) retain the full structural information $\mathcal{M}_x = \emptyset$ to use it as guidance.
- **struct2seq.** Only the structural information from the structural dataset $\mathcal{D}_{\text{struct}}$ is used to predict amino acids $\mathbf{s}_i$ at specific target positions $\mathcal{T}_s$. The input sequence data is completely disregarded, replaced by a series of unknown tokens $*$, leaving only the structural information $\mathbf{x}$. The target positions for amino acid prediction and masked structures, $\mathcal{T}_s = \mathcal{M}_x$, are

chosen using the same distribution as in the seq2seq task, alternating between continuous spans and random positions.

### 2.4 Training details

The model was trained on 8 A10G GPUs using a distributed DDP strategy and the PyTorch Zero Redundancy Optimizer [Rajbhandari et al., 2020]. The total number of training steps was predetermined at 125,000. The learning rate was warmed up over the first 200 steps to a peak of 0.001, after which it was gradually reduced to zero using a cosine scheduler. Training was conducted in 16-bit precision. To conserve memory and enable a larger batch size, gradient activation checkpointing was implemented immediately after the structural module. The effective batch size was set to 90 per GPU, resulting in a total batch size of 720 samples per step. The AdamW optimizer was used with a weight decay parameter of 0.1, epsilon of 0.00001, and betas of [0.9, 0.95] for regularization. More details about the hyperparameters can be found in the Appendix B.

## 3 Empirical results

In this section, we evaluate the impact of incorporating structural information into the pre-training of antibody LLMs. Our evaluation focuses on three tasks: (i) sequence recovery of the variable region, (ii) editing of the CDR regions, and (iii) inverse folding. We compare the performance of IgBlend with five existing open-source antibody and nanobody language models, including AbLang [Olsen et al., 2022b], AbLang2 [Olsen et al., 2024], AntiBERTy [Ruffolo et al., 2021], Sapiens [Prihoda et al., 2022] and Nanobert [Hadsund et al., 2024] as well as two inverse folding models, including AntiFold [Høie et al., 2023] and ESM-IF [Hsu et al., 2022].

### 3.1 Sequence recovery

First, we evaluated the task of recovering missing residues in the variable region of an antibody. This task is particularly relevant for various applications where the goal is either to recover, edit, or mutate specific amino acids within a sequence. Following the benchmark established in [Olsen et al., 2022b, 2024], we proceeded as follows. First, we sampled 1,000 sequences/structures pairs, $(\mathbf{s}, \mathbf{x})$, per chain type from the test distribution, using Equation (1). Then, for each pair $(\mathbf{s}, \mathbf{x})$, we randomly sample a sequential mask $\mathcal{M}_s$ that contains between 10% and 40% of the residue indices from the full sequence $[1, \ldots, |\mathbf{s}|]$. To evaluate the benefits of incorporating structural information, we compared the performance of IgBlend using different input types. Specifically, we assessed each model's ability to recover the full sequence under various conditions: for sequence-only models, we used $\widehat{s} = \text{Model}(\mathbf{s}_{/\mathcal{M}_s})$; for structure-guided sequential models, we used $\widehat{s} = \text{Model}(\mathbf{s}_{/\mathcal{M}_s}, \mathbf{x})$; for sequential models that use masked structural information, we used $\widehat{s} = \text{Model}(\mathbf{s}_{/\mathcal{M}_s}, \mathbf{x}_{/\mathcal{M}_s})$; and for inverse folding models, we used $\widehat{s} = \text{Model}(\mathbf{x})$. For each chain type and each region reg∈[FW1, CDR1, FW2, CDR2, FW3, CDR3, FW4], where FW and CDR refer to framework and CDR regions of the chain respectively, we recorded the empirical accuracy of the models by computing the Levenshtein$(\{\mathbf{s}_i, i \in \mathcal{M}_s \cap \text{reg}\}, \{\widehat{\mathbf{s}}_i, i \in \mathcal{M}_s \cap \text{reg}\})$ distance over each region and averaged the results over 1,000 sequences. Results are reported in Table 2 for the CDR3 regions and the remaining regions can be found in Table 4 of the Appendix. A few remarks are of order:

- First, sequence-only models (AbLang, AbLang2, AntiBERTy, Sapiens, IgBlend) performed similarly across all regions, with at most a 3% difference in accuracy between the models in most regions. In this sense, it has to be noted that IgBlend, trained using a combination of three different objectives shown in Equation (2), performed on par with models trained solely on the seq2seq task, indicating that multi-modal training does not compromise performance on individual tasks.
- Secondly, it is important to note that the performance of IgBlend consistently improves with the addition of more input modalities across all chain types. Specifically, for each chain type, IgBlend shows the same ranking in recovery rate: IgBlend(seq+struct guided) > IgBlend(seq+masked struct) > IgBlend(seq-only) and IgBlend(seq+struct guided) > IgBlend(inverse folding). Hence, adding information helps the model to be more precise and we deduce that proposed training procedure allows us to merge both modalities successfully.

| Mode | Model | Heavy | | | Light | | | Nanobody | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | CDR1 | CDR2 | CDR3 | CDR1 | CDR2 | CDR3 | CDR1 | CDR2 | CDR3 |
| **Sequence Only** | **AbLang** | **84.12** | 80.44 | 53.13 | 74.60 | 72.68 | 66.62 | 44.83 | 44.84 | 21.69 |
| | **AbLang2** | 83.79 | **80.50** | **53.82** | **75.40** | 72.01 | 68.06 | 44.52 | 43.83 | 20.71 |
| | **Antiberty** | 83.72 | 80.30 | 48.37 | 75.12 | 72.75 | **68.21** | 46.16 | 47.29 | 25.63 |
| | **Sapiens** | 81.65 | 76.90 | 48.76 | 72.41 | 69.45 | 63.29 | 45.87 | 42.66 | 19.41 |
| | **Nanobert** | 56.22 | 42.58 | 25.31 | 7.76 | 05.64 | 06.98 | **64.20** | 61.43 | 33.09 |
| | **IgBlend** | 83.80 | 80.07 | 51.91 | 74.63 | **73.79** | 67.37 | 63.83 | **62.68** | **37.37** |
| **Inverse Folding** | **Antifold** | 76.73 | 71.53 | 36.27 | 59.04 | 59.85 | 46.79 | 45.48 | 44.40 | 23.50 |
| | **ESM-IF** | 50.08 | 46.74 | 20.27 | 34.59 | 45.00 | 31.59 | 31.01 | 41.56 | 16.10 |
| | **IgBlend** | **88.15** | **84.88** | **53.35** | **78.26** | **82.42** | **73.01** | **71.49** | **71.33** | **44.42** |
| **Seq + Masked Struct** | **IgBlend** | 85.00 | 80.76 | 54.07 | 75.46 | 75.61 | 69.11 | 67.77 | 64.23 | 40.05 |
| **Seq + Struct Guided** | **IgBlend** | **88.98** | **85.50** | **61.50** | **79.16** | **83.70** | **74.66** | **72.90** | **73.43** | **49.50** |

Table 2: **Sequence recovery results.** The task involves masking a proportion of residues in a sequence and having the model predict them. The table shows the average percentage of successfully recovered masked residues by region and chain type, with bold font highlighting the best results in each category. IgBlend in "Seq + Struct Guided" mode demonstrates the best overall performance.

- Most notably, by incorporating structural information alongside the masked sequence (IgBlend(seq+struct guidance)), we achieved consistently better results than sequence-only models across all regions and for all modalities. This improvement was particularly notable in the CDR3 region of the nanobody (N-CDR3), where IgBlend(seq+struct) outperformed the second-best model by 11.8% in accuracy. Similar improvements were observed in the CDR3 regions of the light chain (L-CDR3) with a 6.6% increase, and the heavy chain (H-CDR3) with a 7.7% increase.

### 3.2 Complementarity-determining region (CDR) editing

Second, we focused on the task of editing/recovering the CDR regions of a single chain, which is of particular importance in the process of optimizing antibodies for affinity. In this task, one of the CDR regions is randomly selected and fully masked, i.e., $\mathcal{M}_s \in \{\text{CDR1}, \text{CDR2}, \text{CDR3}\}$, and the models are asked to predict the residues within the selected fully masked CDR. Using the same experimental setup with the masked CDR $\mathcal{M}_s$, we evaluated the models—seq-only, structure-guided, and inverse folding—on 1,000 sequences sampled from the test set as described in Equation (1). These sequences were not seen during the training of IgBlend, and we recorded the percentage of successfully recovered residues for each chain type. The results can be found in the table shown in Figure 3 with all models being evaluated on the same masked sequences. To further evaluate how well models using structural information adhere to structural instructions, we assessed the structural similarities between the generated sequences and the input structure $\mathbf{x}$. We compared the top models in each category: AbLang2 for heavy/light chains, NanoBert for nanobodies, and AntiFold for inverse folding. We sampled 500 sequences per chain type from the test distribution and tasked each model with recovering a missing CDR. For each recovered sequence $\widehat{s}$, we computed its structural approximation $\widehat{\mathbf{x}} = \text{IgFold}(\widehat{s})$ using IgFold with PyRosetta refinement. We then measured the Levenshtein$(\{s_i, i \in \mathcal{M}_s\}, \{\widehat{s}_i, i \in \mathcal{M}_s\})$ distance in the masked region and the RMSD$(\{x_i, i \in \mathcal{M}_s\}, \{\widehat{x}_i, i \in \mathcal{M}_s\})$ between the original and recovered structures to assess structural similarity. Results are shown in Figure 3, with extended findings available in Appendix D.2. From both evaluations, key observations include:

- First, as in previous experiments, the top-performing sequence-only models (AbLang, AbLang2, AntiBERTy, IgBlend) showed similar performance across different CDR regions. However, IgBlend outperformed the best sequence-only model by over 9% in accuracy for nanobodies. Notably, incorporating additional information consistently improved IgBlend's performance across all chain types (i.e., IgBlend (Seq+Struct Guided) > IgBlend (Seq+Masked Struct) > IgBlend (Seq-only)). Specifically, adding structural information to the masked sequence (IgBlend (seq+struct guidance)) significantly enhanced performance compared to the best sequence-only models, with improvements of 11.8% in H-CDR3, 6.74% in L-CDR3, and 15.43% in N-CDR3.
- Second, unlike the previous experiments, IgBlend (seq+struct guidance), which uses both $(\mathbf{s}_{\mathcal{M}_s}, \mathbf{x}_{\mathcal{M}_x})$, shows performance closer to IgBlend (inverse folding), which relies solely on the structure $\mathbf{x}$, than to IgBlend (seq-only), which depends only on the sequential information $\mathbf{s}_{\mathcal{M}_s}$. This suggests that, in the task of re-editing complete CDR regions, IgBlend relies more on structural information than on sequential data.

| Mode | Model | Heavy | | | Light | | | Nanobody | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | CDR1 | CDR2 | CDR3 | CDR1 | CDR2 | CDR3 | CDR1 | CDR2 | CDR3 |
| **Sequence Only** | **AbLang** | 82.97 | **80.53** | 41.68 | 72.21 | 69.27 | 67.47 | 43.73 | 45.09 | 20.90 |
| | **AbLang2** | 82.85 | 80.31 | 41.62 | 72.94 | 69.66 | 68.03 | 43.05 | 41.43 | 20.16 |
| | **Antiberty** | 82.90 | 80.37 | 41.23 | 72.64 | 69.20 | 68.61 | 40.48 | 47.76 | 23.12 |
| | **Sapiens** | 81.44 | 77.13 | 38.45 | 71.18 | 67.22 | 63.03 | 44.25 | 39.99 | 19.79 |
| | **Nanobert** | 57.33 | 40.00 | 24.02 | 10.16 | 08.53 | 07.22 | 60.49 | 61.09 | 29.08 |
| | **IgBlend** | **83.15** | 80.33 | **41.84** | **73.14** | **69.79** | **68.70** | **62.58** | **63.81** | **29.53** |
| **Inverse Folding** | **AntiFold** | 75.41 | 70.99 | 36.97 | 57.05 | 58.98 | 49.12 | 44.70 | 44.92 | 22.02 |
| | **ESM-IF** | 49.90 | 44.19 | 19.65 | 33.68 | 43.70 | 31.46 | 30.74 | 39.98 | 15.34 |
| | **IgBlend** | **86.18** | **84.44** | **52.69** | **76.69** | **82.03** | **73.9** | **69.72** | **72.58** | **43.77** |
| **Seq + Masked Struct** | **IgBlend** | 84.00 | 80.61 | 43.37 | 74.00 | 73.10 | 70.61 | 65.93 | 64.75 | 32.28 |
| **Seq + Struct Guided** | **IgBlend** | **87.27** | **85.04** | **53.65** | **77.08** | **83.59** | **75.44** | **71.40** | **73.52** | **44.96** |

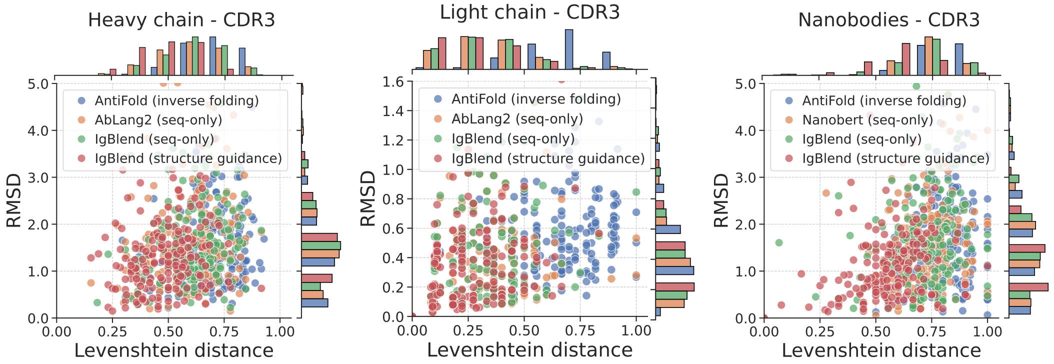


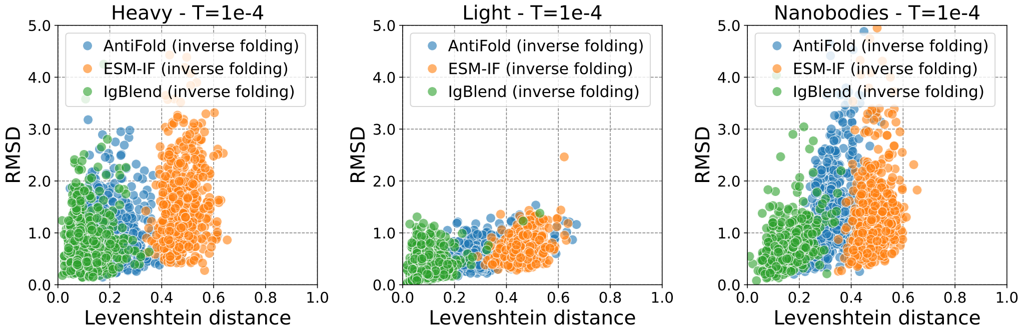
Figure 3: **CDR in-filling results:** One CDR region (CDR1, CDR2, or CDR3) is fully masked, and the model attempts to recover it. **Top:** The table shows the average percentage of correctly recovered residues for heavy chain (H), light chain (L) and nanobodies (N). **Bottom:** The graphs show Levenshtein distances of generated CDR3 regions from the original sequence on the x-axis, and RMSD of the predicted structures from the original backbone on the y-axis for each chain type.

- In terms of structural similarity, it is noteworthy that IgBlend (structure guided) consistently achieves the highest percentage of structures within the bins corresponding to the smallest RMSD values for each chain type, surpassing even the best inverse folding model. Specifically, IgBlend shows 37%, 46%, and 51% of structures in the smallest RMSD bins (1 Å for heavy chains, 0.2 Å for light chains, and 1 Å for nanobodies), compared to 32%, 5%, and 26% for AntiFold in the same bins. See Figure 3 for a detailed comparison. Thus, in addition to outperforming sequential models, IgBlend(seq+struct guided) excels at generating sequences that can more accurately fold to the original backbone structure compared to those produced by AntiFold.

### 3.3 Inverse folding

Finally, we assessed IgBlend's ability to perform the inverse folding task [Hsu et al., 2022, Høie et al., 2023], which involves recovering a sequence $\mathbf{s}$ from its structure $\mathbf{x}$ alone. As with previous experiments, we sampled 500 structure-sequence pairs $(\mathbf{s}, \mathbf{x})$ for each chain type from the test set, which was not seen during IgBlend's training. Each inverse folding model was then asked to predict the sequence $\widehat{\mathbf{s}} = \text{Model}(\mathbf{x})$ based solely on the structure $\mathbf{x}$. To evaluate model performance, we tested different temperatures: $T = 1e - 4$ for the highest probability sequence, $T = 1$ for unbiased results, and $T = 2$ and $T = 3$ for more diverse sequences. We recorded the normalized Levenshtein$(\mathbf{s}, \widehat{\mathbf{s}})$ distance between the predicted and original sequences, and the RMSD$(\mathbf{x}, \widehat{\mathbf{x}})$ of the approximated structure $\widehat{\mathbf{x}} = \text{IgFold}(\widehat{\mathbf{s}})$ as a measure of structural similarity. Results for the lowest temperature are shown in Figure 4, with additional details in Table 5 and Figure 6 in the Appendix. Key observations include:

- First, we observe a positive correlation between Levenshtein distance and RMSD for every model: as the sequence diverges more from the original (larger Levenshtein distance), the RMSD tends to increase, indicating a trade-off between sequence diversity and structural precision. Consequently, as temperature is increased to generate more diverse sequences,



| | Heavy | | | Light | | | Nanobodies | | |
|---|---|---|---|---|---|---|---|---|---|
| **RMSD<1Å** | **AntiFold** | **ESM-IF** | **IgBlend** | **AntiFold** | **ESM-IF** | **IgBlend** | **AntiFold** | **ESM-IF** | **IgBlend** |
| **T=1e-4** | 63.60 | 28.20 | **73.60** | 94.40 | 89.20 | **97.80** | 43.00 | 38.00 | **81.00** |
| **T=1** | 60.80 | 14.40 | **69.40** | 85.20 | 71.00 | **97.00** | 45.40 | 24.80 | **79.00** |
| **T=2** | 35.11 | 02.01 | **52.80** | 61.25 | 18.84 | **94.60** | 37.15 | 03.01 | **57.80** |
| **T=3** | 06.11 | 00.00 | **17.60** | 10.03 | 00.43 | **54.60** | 05.26 | 00.00 | **23.40** |

Figure 4: **Inverse Folding Results:** In this task, the sequence is fully masked, and the model attempts to recover it from the structure. The top graph shows the normalized Levenshtein distance between generated and original sequences, with the y-axis displaying the RMSD of the generated structures relative to the original; greater spread in Levenshtein distance and lower RMSD indicate better performance. The bottom table lists the percentage of times each method produced a sequence with RMSD < 1 Å across 500 samples per modality. More details are available in Table 5 in the Appendix.

the RMSD increases. However, IgBlend demonstrates greater robustness to temperature changes. Second, all models perform better on light chains compared to heavy chains and nanobodies, suggesting that the inverse folding task is more challenging for heavy chains and nanobodies.

- Second, across all temperatures, chain types, and RMSD thresholds (see Figure 4 and Table 5 in the Appendix), the models consistently rank as follows based on the number of samples with RMSD below the threshold: IgBlend> AntiFold > ESM-IF. This shows that IgBlend outperforms current state-of-the-art methods in generating sequences that accurately fold back to the original structure. Notably, IgBlend is the first inverse folding model to achieve results on nanobodies comparable to heavy chains. However, this high accuracy comes with lower Levenshtein distance, a limitation seen in all tested settings.

## 4 Conclusion and future work

In this study, we investigated how incorporating structural information into antibody LLMs enhances performance. We introduced a model that integrates both structural and sequential data, showing that this combination consistently improves performance across all benchmarks compared to sequence-based and inverse-folding models. However, while effective, our approach sometimes sacrifices sequence diversity for accuracy. Future work will focus on including side-chain information and expanding structural datasets.

## Acknowledgement

We extend our gratitude to everyone in the MLAB program at AstraZeneca, with special thanks to Massimo Sammito and Lorena Roldan Martin for their invaluable support. Additionally, we are grateful to Tom Diethe and Rebecca Croasdale-Wood for their unwavering support.

## References

Brennan Abanades, Tobias H Olsen, Matthew I J Raybould, Broncio Aguilar-Sanjuan, Wing Ki Wong, Guy Georges, Alexander Bujotzek, and Charlotte M Deane. The Patent and Literature Antibody Database (PLAbDab): an evolving reference set of functionally diverse, literature-annotated antibody sequences and structures. *Nucleic Acids Research*, 52(D1):D545–D551, 11 2023a. ISSN 0305-1048. doi: 10.1093/nar/gkad1056. URL `https://doi.org/10.1093/nar/gkad1056`.

Brennan Abanades, Wing Ki Wong, Fergus Boyles, Guy Georges, Alexander Bujotzek, and Charlotte M Deane. ImmuneBuilder: Deep-learning models for predicting the structures of immune proteins. *Communications Biology*, 6(1):575, 2023b.

Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

Piotr Deszyński, Jakub Młokosiewicz, Adam Volanakis, Igor Jaszczyszyn, Natalie Castellana, Stefano Bonissone, Rajkumar Ganesan, and Konrad Krawczyk. Indi—integrated nanobody database for immunoinformatics. *Nucleic Acids Research*, 50(D1):D1273–D1281, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 2018.

James Dunbar and Charlotte M Deane. ANARCI: antigen receptor numbering and receptor classification. *Bioinformatics*, 32(2):298–300, 2016.

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.

François Ehrenmann, Patrice Duroux, Véronique Giudicelli, and Marie-Paule Lefranc. Standardized sequence and structure analysis of antibody using IMGT®. *Antibody engineering*, pages 11–31, 2010.

Johannes Thorling Hadsund, Tadeusz Satława, Bartosz Janusz, Lu Shan, Li Zhou, Richard Röttger, and Konrad Krawczyk. nanobert: a deep learning model for gene agnostic navigation of the nanobody mutational space. *Bioinformatics Advances*, 4(1):vbae033, 2024.

Tomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *bioRxiv*, pages 2024–07, 2024.

Magnus Høie, Alissa Hummer, Tobias Olsen, Morten Nielsen, and Charlotte Deane. Antifold: Improved antibody structure design using inverse folding. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023.

Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pages 8946–8970. PMLR, 2022.

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.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *nature*, 596(7873):583–589, 2021.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976.

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, 2022a.

Tobias H Olsen, Iain H Moal, and Charlotte M Deane. AbLang: an antibody language model for completing antibody sequences. *Bioinformatics Advances*, 2(1):vbac046, 2022b.

Tobias Hegelund Olsen, Iain H Moal, and Charlotte Deane. Addressing the antibody germline bias and its effect on language models for improved antibody design. *bioRxiv*, pages 2024–02, 2024.

David Prihoda, Jad Maamary, Andrew Waight, Veronica Juan, Laurence Fayadat-Dilman, Daniel Svozil, and Danny A Bitton. BioPhi: A platform for antibody design, humanization, and humanness evaluation based on natural antibody repertoires and deep learning. In *MAbs*, volume 14, page 2020203. Taylor & Francis, 2022.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118. URL `https://www.pnas.org/doi/full/10.1073/pnas.2016239118`. bioRxiv 10.1101/622803.

Jeffrey A Ruffolo, Jeffrey J Gray, and Jeremias Sulam. Deciphering antibody affinity maturation with language models and weakly supervised learning. *arXiv preprint arXiv:2112.07782*, 2021.

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.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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*, pages 42317–42338. PMLR, 2023.

# A Background on antibodies

**Background.** In humans, antibodies are classified into five isotypes: IgA, IgD, IgE, IgG, and IgM. This work primarily focuses on IgG antibodies, which are Y-shaped glycoproteins produced by B-cells (see Figure 1), as well as nanobodies, which are antibody fragments consisting of a single monomeric variable domain. Henceforth, "antibody" will specifically refer to IgG antibodies. Antibodies consist of distinct regions that play specific roles in the immune response. The Fab (fragment antigen-binding) region, composed of both variable (V) and constant (C) domains from the heavy and light chains, is primarily responsible for antigen binding. Within this region, the antigen-binding site is formed by the variable domains — VH for the heavy chain and VL for the light chain — which determine the specificity of the antibody and enable it to recognize and bind to specific antigens. The Fv (fragment variable) region is the smallest functional unit of an antibody that can still bind to an antigen. It consists solely of the variable domains (VH and VL) of the heavy and light chains, without the constant domains. Within the variable domains, there are two key distinct regions: the framework regions and the complementarity-determining regions (CDRs). The framework regions provide structural support, maintaining the overall shape of the variable domains, while the CDRs, comprising three loops on both the VH and VL chains, are directly involved in binding to the antigen. These CDRs are crucial for the precise recognition and interaction with specific antigens. While the Fv region is essential for the initial recognition and binding of antigens, it lacks the effector functions present in the full antibody. The Fab region, being larger and more complex due to the inclusion of both variable and constant domains, is generally more stable and has a higher affinity for antigens. The Fv region, on the other hand, is simpler and more easily engineered for various applications, such as in the development of single-chain variable fragment (scFv) antibodies. The base of the Y-shaped antibody, known as the Fc (fragment crystallizable) region, is involved in regulating immune responses. It interacts with proteins and cell receptors, ensuring that the antibody generates an appropriate immune response. Moreover, nanobodies, which are small, single-domain antibodies derived from heavy-chain-only antibodies found in certain animals such as camels and llamas, are even more compact than traditional Fv regions. They retain full antigen-binding capacity while offering advantages such as increased stability and easier production, making them valuable tools in both therapeutic and diagnostic applications.

# B Architectural details

In Table 3, we collect the full architectural details of the IgBlendarchitecture used in the paper.

# C Data set

**Data source.** To create a model capable of processing both sequential and structural information, we needed to address the significant asymmetry in the availability of data across these modalities (204M sequences and 3M structures as shown in Table 1). Therefore, we compiled two datasets: (1) a structural dataset $\mathcal{D}_{\text{struct}}$, which includes structures paired with their corresponding sequences, and (2) a sequential dataset $\mathcal{D}_{\text{seq}}$, which consists solely of sequence data. These datasets were derived from four primary sources: SAbDab [Dunbar et al., 2014], which contains experimentally determined structures using techniques such as electron crystallography and X-ray diffraction; PLAbDab [Abanades et al., 2023a], which provides sequences derived from patents; OAS datasets [Olsen et al., 2022a], which compile and annotate immune repertoires; and INDI [Deszyński et al., 2022], which contains sequences of nanobodies. Given the relatively small number of experimentally determined structures (e.g., approximately 2,000 samples from SAbDab, as shown in Table 1 after applying our selection criteria), we expanded our structural dataset by incorporating inferred structures. In addition to the inferred structures already present in the PLAbDab dataset (folded with ImmuneBuilder), we generated additional structures from the OAS paired, unpaired and INDI. The paired sequences from OAS were folded with ImmuneBuilder [Abanades et al., 2023b] and a clustered version of the unpaired OAS and INDI dataset were folded using IgFold [Ruffolo et al., 2023]. This process resulted in approximately 4 million unique structures. For the sequential dataset, we extracted data from four repertoires: OAS paired, OAS unpaired, PLAbDab paired, PLAbDab unpaired and INDI.

For each of the datasets $\mathcal{D}_{\text{struct}}$ and $\mathcal{D}_{\text{seq}}$, we begin by removing all duplicates, defined as pairs of data with identical sequences. Next, only the data that meet the following criteria are retained:

| Structure module | value |
|---|---|
| gvp_eps | 0.0001 |
| gvp_node_hidden_dim_scalar | 512 |
| gvp_node_hidden_dim_vector | 256 |
| gvp_num_encoder_layers | 4 |
| gvp_dropout | 0.1 |
| gvp_encoder_embed_dim | 512 |
| transformer_encoder_layers | 2 |
| encoder_embed_dim | 512 |
| transformer_dropout | 0.1 |
| encoder_attention_heads | 8 |
| encoder_ffn_embed_dim | 1024 |
| Sequence Module | |
| d_model | 512 |
| dropout | 0.1 |
| layer_norm_eps | 0.0001 |
| nhead | 8 |
| activation | SwiGLU |
| dim_feedforward | 512 |
| layer_norm_eps | 0.0001 |
| Multi-modal encoder | |
| d_model | 1024 |
| num_layers | 4 |
| n_head | 16 |
| dim_feedforward | 1024 |
| activation | SwiGLU |
| prediction_head | |
| d_model | 1024 |
| activation | GELU |

Table 3: Hyper-parameters of the IgBlendmodel.

(1) no unknown residues, (2) no missing residues, and (3) no shorter than expected IMGT regions [Ehrenmann et al., 2010], as determined by running ANARCI [Dunbar and Deane, 2016]. After these cleaning steps, we are left with two datasets: $\mathcal{D}_{\text{struct}} = \{(\mathbf{s}, \mathbf{x})_1, \ldots, (\mathbf{s}, \mathbf{x})_{|\mathcal{D}_{\text{struct}}|}\}$, which contains pairs of sequences and structures, and $\mathcal{D}_{\text{seq}} = \{(s, *)_1, \ldots, (s, *)_{|\mathcal{D}_{\text{seq}}|}\}$, which contains only sequential information.

# D Experimental results

## D.1 Sequence recovery

Table 4 records the sequence recovery rate on all regions and for each modality.

## D.2 CDR editing

Figure 5 collects the result of the CDR recovery experiment in all CDR regions.

## D.3 Inverse folding

Figure 6 displays the inverse folding results for the different temperatures. Table 5 reports the results for different RMSD thresholds.

| Heavy chains | FW-1 | CDR-1 | FW-2 | CDR-2 | FW-3 | CDR-3 | FW-4 |
|---|---|---|---|---|---|---|---|
| AbLang (seq-only) | 95.65 | 84.12 | 93.49 | 80.44 | 92.22 | 53.13 | 96.32 |
| AbLang2 (seq-only) | 95.54 | 83.79 | 93.67 | 80.50 | 92.21 | 53.82 | 96.16 |
| Antiberty (seq-only) | 95.71 | 83.72 | 93.24 | 80.30 | 92.15 | 48.37 | 96.27 |
| Sapiens (seq-only) | 94.23 | 81.65 | 91.13 | 76.90 | 89.21 | 48.76 | 95.31 |
| Nanobert (seq-only) | 74.48 | 56.22 | 72.97 | 42.58 | 65.39 | 25.31 | 85.17 |
| IgBlend(seq-only) | 95.66 | 83.80 | 93.25 | 80.07 | 91.91 | 51.91 | 96.23 |
| IgBlend(seq+masked struct) | 95.86 | 85.00 | 93.32 | 80.76 | 91.96 | 54.07 | 96.10 |
| IgBlend(seq+struct guided) | **96.52** | **88.98** | **95.38** | **85.50** | **93.68** | **61.50** | **97.15** |
| IgBlend(inverse folding) | 96.02 | 88.15 | 94.94 | 84.88 | 93.36 | 53.35 | 96.64 |
| Antifold (inverse folding) | 87.07 | 76.73 | 88.90 | 71.53 | 88.66 | 36.27 | 91.70 |
| ESM-IF (inverse folding) | 55.69 | 50.08 | 63.43 | 46.74 | 59.41 | 20.27 | 57.96 |
| Light chains | FW-1 | CDR-1 | FW-2 | CDR-2 | FW-3 | CDR-3 | FW-4 |
| AbLang (seq-only) | 93.18 | 74.60 | 88.55 | 72.68 | 92.70 | 66.62 | 93.31 |
| AbLang2 (seq-only) | 94.06 | 75.40 | 88.79 | 72.01 | 93.01 | 68.06 | 93.54 |
| Antiberty (seq-only) | 94.05 | 75.12 | 88.63 | 72.75 | 93.01 | 68.21 | 93.63 |
| Sapiens (seq-only) | 92.94 | 72.41 | 87.25 | 69.45 | 91.58 | 63.29 | 88.45 |
| Nanobert (seq-only) | 16.15 | 7.76 | 19.27 | 05.64 | 21.12 | 06.98 | 41.97 |
| IgBlend(seq-only) | 93.97 | 74.63 | 88.43 | 73.79 | 92.86 | 67.37 | 92.32 |
| IgBlend(seq+masked struct) | 94.00 | 75.46 | 89.17 | 75.61 | 93.00 | 69.11 | 94.10 |
| IgBlend(seq+struct guided) | **95.07** | **79.16** | **91.78** | **83.70** | **94.43** | **74.66** | **96.46** |
| IgBlend(inverse folding) | 94.37 | 78.26 | 91.19 | 82.42 | 93.89 | 73.01 | 95.59 |
| Antifold (inverse folding) | 68.86 | 59.04 | 76.40 | 59.85 | 84.69 | 46.79 | 75.08 |
| ESM-IF (inverse folding) | 56.32 | 34.59 | 63.63 | 45.00 | 64.52 | 31.59 | 51.89 |
| Nanobodies | FW-1 | CDR-1 | FW-2 | CDR-2 | FW-3 | CDR-3 | FW-4 |
| AbLang (seq-only) | 87.46 | 44.83 | 60.88 | 44.84 | 78.49 | 21.69 | 87.29 |
| AbLang2 (seq-only) | 87.21 | 44.52 | 60.58 | 43.83 | 78.07 | 20.71 | 87.94 |
| Antiberty (seq-only) | 87.10 | 46.16 | 74.53 | 47.29 | 85.09 | 25.63 | 95.85 |
| Sapiens (seq-only) | 88.65 | 45.87 | 60.35 | 42.66 | 75.58 | 19.41 | 86.01 |
| Nanobert (seq-only) | 93.44 | 64.20 | 86.92 | 61.43 | 88.32 | 33.09 | 97.12 |
| IgBlend(seq-only) | 93.35 | 63.83 | 87.40 | 62.68 | 88.40 | 37.37 | 97.24 |
| IgBlend(seq+masked struct) | 94.79 | 67.77 | 88.07 | 64.23 | 88.73 | 40.05 | 97.35 |
| IgBlend(seq+struct guided) | **96.45** | **72.90** | **92.26** | **73.43** | **91.94** | **49.50** | **97.72** |
| IgBlend(inverse folding) | 96.04 | 71.49 | 91.93 | 71.33 | 91.65 | 44.42 | 97.22 |
| Antifold (inverse folding) | 87.38 | 45.48 | 64.56 | 44.40 | 80.09 | 23.50 | 87.32 |
| ESM-IF (inverse folding) | 56.83 | 31.01 | 57.67 | 41.56 | 62.43 | 16.10 | 55.13 |

Table 4: **Sequence recovery results.** The task consists of masking randomly a proportion of residues within a sequence and asking the model to predict the masked residues. The table display the average percentage of successfully recovered masked residues in each region and for each type of chain. Bold font indicates the best result in the comparison

| | **Heavy** | | | **Light** | | | **Nanobodies** | | |
|---|---|---|---|---|---|---|---|---|---|
| **RMSD<0.5** | **AntiFold** | **ESM-IF** | **IgBend** | **AntiFold** | **ESM-IF** | **IgBend** | **AntiFold** | **ESM-IF** | **IgBend** |
| **T=1e-4** | 20.40 | 02.40 | **27.6** | 49.20 | 24.20 | **72.0** | 07.20 | 03.40 | **32.00** |
| **T=1** | 18.60 | 00.80 | **25.00** | 34.00 | 09.80 | **68.00** | 08.79 | 02.00 | **30.00** |
| **T=2** | 05.95 | 00.00 | **10.60** | 07.50 | 00.00 | **47.40** | 04.75 | 00.00 | **08.80** |
| **T=3** | 00.00 | 00.00 | **01.00** | 00.00 | 00.00 | **06.60** | 00.00 | 00.00 | **00.80** |
| **RMSD<1** | **AntiFold** | **ESM-IF** | **IgBend** | **AntiFold** | **ESM-IF** | **IgBend** | **AntiFold** | **ESM-IF** | **IgBend** |
| **T=1e-4** | 63.60 | 28.20 | **73.60** | 94.40 | 89.20 | **97.80** | 43.00 | 38.00 | **81.00** |
| **T=1** | 60.80 | 14.40 | **69.40** | 85.20 | 71.00 | **97.00** | 45.40 | 24.80 | **79.00** |
| **T=2** | 35.11 | 02.01 | **52.80** | 61.25 | 18.84 | **94.60** | 37.15 | 03.01 | **57.80** |
| **T=3** | 06.11 | 00.00 | **17.60** | 10.03 | 00.43 | **54.60** | 05.26 | 00.00 | **23.40** |
| **RMSD<1.5** | **AntiFold** | **ESM-IF** | **IgBend** | **AntiFold** | **ESM-IF** | **IgBend** | **AntiFold** | **ESM-IF** | **IgBend** |
| **T=1e-4** | 84.60 | 57.80 | **89.60** | 99.80 | 99.80 | **100.0** | 62.60 | 67.60 | **95.60** |
| **T=1** | 81.20 | 42.00 | **86.60** | 99.80 | 95.80 | **100.0** | 72.20 | 56.60 | **94.00** |
| **T=2** | 66.17 | 11.85 | **78.40** | 92.50 | 46.69 | **99.40** | 62.85 | 17.27 | **80.60** |
| **T=3** | 23.40 | 00.65 | **51.00** | 27.96 | 04.49 | **83.40** | 24.44 | 00.43 | **51.20** |
| **RMSD<2** | **AntiFold** | **ESM-IF** | **IgBend** | **AntiFold** | **ESM-IF** | **IgBend** | **AntiFold** | **ESM-IF** | **IgBend** |
| **T=1e-4** | 94.00 | 77.80 | **96.40** | 100.0 | 99.80 | **100.0** | 78.20 | 84.80 | **98.20** |
| **T=1** | 92.40 | 66.00 | **95.40** | 100.0 | 99.00 | **100.0** | 84.20 | 75.80 | **97.80** |
| **T=2** | 84.68 | 27.91 | **90.80** | 98.12 | 68.34 | **99.60** | 77.97 | 34.14 | **90.00** |
| **T=3** | 47.12 | 02.16 | **71.60** | 48.02 | 11.32 | **93.00** | 39.85 | 02.59 | **69.80** |

Table 5: **Inverse folding results:** The sequence if fully masked, and the model attempts to recover it from the structure. The table displays the percentage of sequences generated by each method with a RMSD below a given threshold and for different temperatures. Higher is better.

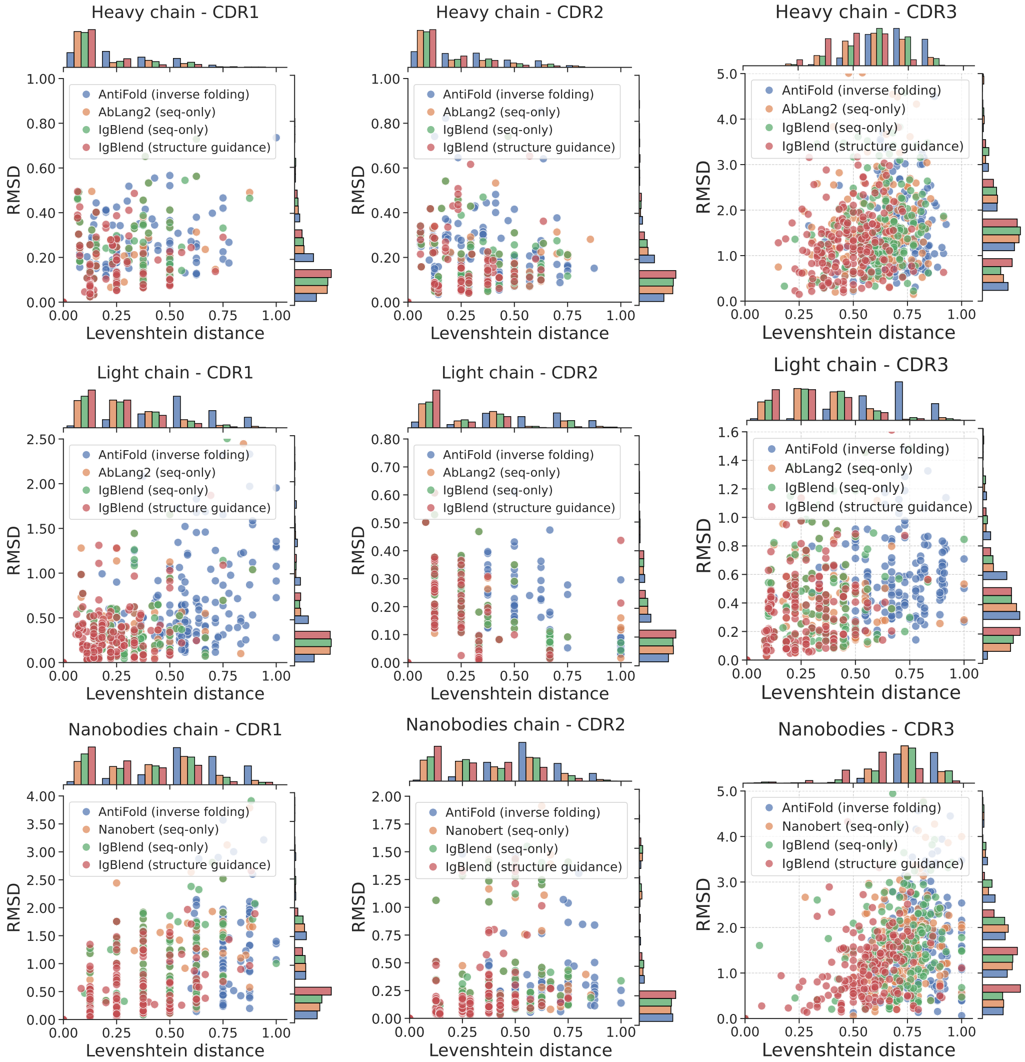


Figure 5: **CDR recovery results:** One series of aminco acid of the sequence is fully masked (one CDR), and the model attemps to recover it. AntiFold only uses the structural information. IgBlend (structure guidance) uses the masked sequence and the structure information. The distances (both Levenshtein and RSME) are only computed in the masked CDR regions. The x-axis displays the Levensthein distance of the generated sequences to the original one and the y-axis reports the RMSE of the generated sequence with regards to the original structure.

## NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims match the empirical results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.

- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The percentage in each regions are reported, suggesting that the algorithms are not efficient in CDR3 regions. Moreover, limitations are highlighted in the conclusion and future work session.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [NA]

   Justification: There are no theoretical results

   Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: All the datasets are open sourced and the data processing is described in the paper. Moreover, the hyper-parameter of the architecture and training details are also reported.

   Guidelines:

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

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

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

   Answer: [No]

   Justification: The models parameters are not open-sourced

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: Yes, they are given in the appendix

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [No]

   Justification: For sequence recovery, only the average is reported.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

Answer: [Yes]

Justification: The details are provided in the training details

Guidelines:

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

9. **Code Of Ethics**

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

Answer: [Yes]

Justification: the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics

Guidelines:

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

10. **Broader Impacts**

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

Answer: [No]

Justification: Societal impact is not discussed

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [No]

    Justification: No, safeguards are not dicsussed

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All the contributors are credited

    Guidelines:

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

13. **New Assets**

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

    Answer: [NA]

    Justification: there is no asset

    Guidelines:

    - The answer NA means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: not applicable

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: not applicable

    Guidelines:

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

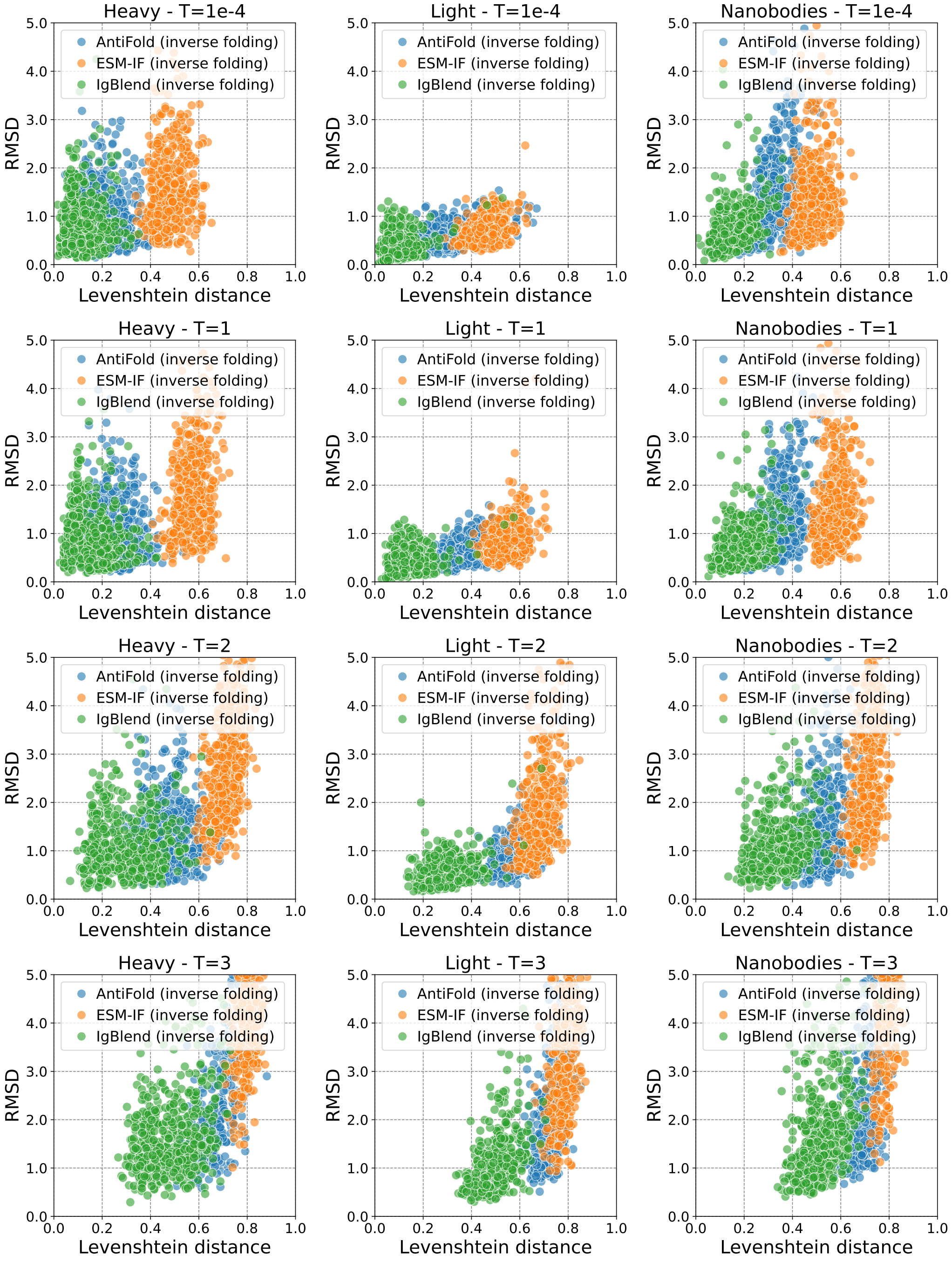


Figure 6: **Inverse folding results:** The sequence if fully masked, and the model attempts to recover it from the structure. **Top:** the graph displays the normalized Levenshtein distance of the generated sequences to the original sequences associated with the input structure and the y-axis reports the RMSD of the folded structure of the generated sequences with regards to the original structure set as input. For both metrics, lower is better.