DISTOGRAM-BASED STRUCTURAL LOSS: A PLUG-IN MODULE FOR PROTEIN INVERSE FOLDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein inverse folding aims to design amino acid sequences that fold into given backbone structures. However, most inverse folding models are trained primarily to recover native sequences, an objective that does not directly assess structural consistency between generated sequences and the target backbone. Here, we introduce DISCO loss, a model-agnostic auxiliary objective that incorporates pairwise distance information from the target backbone into inverse folding training. A jointly trained auxiliary module predicts residue-pair distance distributions using representations derived from inverse folding outputs, supervised by discretized distances computed from the target backbone. Combined with the sequence recovery loss, DISCO loss provides structural supervision without full 3D reconstruction or separate surrogate pretraining. On de novo protein backbones, DISCO loss improves structural self-consistency across inverse folding models with distinct generation mechanisms, with improvements also observed in long-range contact recovery. Additional evaluations show that sequence recovery performance and diversity are largely maintained. These results demonstrate the effectiveness of auxiliary supervision based on target-backbone distances in improving the structural consistency of generated sequences.

1 INTRODUCTION

Proteins perform diverse biological functions through their three-dimensional (3D) structures, which are determined by their amino acid sequences. Designing sequences that adopt desired structures and support specific functions is therefore a central goal in computational biology, with applications in enzyme engineering and drug discovery. However, the vast combinatorial space of possible amino acid sequences makes exhaustive experimental exploration infeasible, motivating computational approaches. Among these, inverse folding, the task of designing amino acid sequences that fold into a given target backbone structure, has become a core problem in structure-based protein design.

The inverse folding problem is formally defined as estimating the conditional probability distribution over amino acid sequences given a target 3D backbone structure. Due to the one-to-many nature of this mapping, inverse folding is treated as a probabilistic modeling problem rather than a deterministic prediction task. Deep learning-based methods have addressed this through architectural innovations such as graph-based structural encoders with message passing (Ingraham et al., 2019; Dauparas et al., 2022; Jing et al., 2021), improved decoding strategies, and integration of pre-trained protein language models, with data augmentation further improving generalization (Gao et al., 2023; Hsu et al., 2022; Widatalla et al., 2025; Yi et al., 2023). While these approaches differ in architecture and training strategies, they largely share the objective function of recovering native sequences from backbone structures.

This objective function is typically implemented by minimizing cross-entropy between predicted amino acid distributions and the native sequence. Although inverse folding models use backbone geometry as input, this loss supervises residue identities without explicitly assessing structural consistency between generated sequences and the target backbone. To examine whether high sequence recovery is accompanied by high structural consistency, we generated sequences with 90% sequence recovery by randomly substituting 10% of native residues and evaluated their structural consistency using ESMFold (Lin et al., 2023). Figure 1 shows substantial variation in self-consistency TM-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

score (scTM) despite identical sequence recovery, suggesting that high sequence similarity does not necessarily translate into high structural consistency with the target backbone.

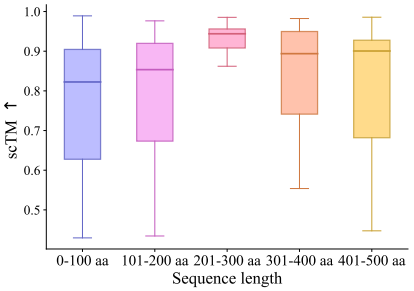

Figure 1: Distribution of scTM for sequences with 90% sequence recovery across different protein lengths.

To address this limitation of sequence recovery–based training, recent work incorporates feedback from protein folding models. DPO-based approaches fine-tune pretrained inverse folding models using preference data constructed by comparing predicted candidate structures with the target backbone (Xu et al., 2026). As an alternative, AFDistill can incorporate structural regularization into inverse folding training from scratch, but requires a separately pretrained surrogate that approximates scores from protein folding models, such as pTM and pLDDT (Melnyk et al., 2025). These approaches require additional data construction using protein folding models for structural supervision, with AFDistill further requiring separate surrogate pretraining. We instead propose an end-to-end framework that uses target-backbone residue-pair distances as supervision to jointly train the inverse folding model and an auxiliary distogram prediction module from scratch, without additional data construction or separate surrogate pretraining.

Concretely, our framework combines an inverse folding model with an auxiliary distogram prediction module. The auxiliary module predicts residue-pair distance distributions using representations derived from the inverse folding model's per-residue amino acid probabilities, supervised by discretized distance labels obtained from the ground-truth distance map of the target backbone. The total training objective combines the standard sequence recovery loss with the proposed auxiliary distogram-prediction loss, which we term DISCO (DIstogram-based Structural COnsistency) loss. This loss provides structural supervision to the inverse folding model's outputs using target-backbone distance information during training. This model-agnostic formulation allows the auxiliary module to be integrated into inverse folding training across different generation mechanisms and removed at inference.

Our contributions are threefold. First, we introduce DISCO loss, an auxiliary objective for jointly training an inverse folding model and an auxiliary module to predict target-backbone residue-pair distance distributions from the inverse folding model's outputs. This approach can be applied to inverse folding models with different generation mechanisms without full 3D reconstruction, separate surrogate pretraining, or additional inference cost. Second, we demonstrate improvements in structural self-consistency and long-range contact recovery across three inverse folding models, evaluated using multiple protein folding models. Third, through multiple complementary evaluations, we show that these structural improvements are achieved while largely maintaining sequence-level performance and diversity, and characterize the accompanying changes in physicochemical properties.

2 Related Works

2.1 Protein Design

Protein design seeks to generate amino acid sequences or structures that realize desired functions. Deep learning approaches can be broadly divided into two paradigms: decoupled approaches, where structure and sequence are addressed separately, and joint approaches, where both are addressed simultaneously. For structure generation, diffusion- and flow matching-based methods directly generate 3D protein backbones, enabling applications such as motif scaffolding and binder design (Watson et al., 2023; Wang et al., 2022a; Trippe et al., 2023). For sequence design, methods such as ProteinMPNN (Dauparas et al., 2022), PiFold (Gao et al., 2023), ESM-IF (Hsu et al., 2022), Grade-IF (Yi et al., 2023) and FAMPNN (Widatalla et al., 2025) generate sequences conditioned on a given backbone. However, these models are typically trained to maximize sequence likelihood, which does not explicitly enforce that the generated sequences fold into the target structure. Joint approaches, including Chroma (Ingraham et al., 2023), ESM3 (Hayes et al., 2025), and CarbonNovo (Ren et al., 2024), instead model the structure–sequence relationship directly. While effective, they require substantially heavier training and dedicated architectures.

2.2 Structural Similarity and Self-Consistency

Structural similarity between proteins is commonly measured by RMSD, the average positional deviation between corresponding atoms, and TM-score, a length-independent measure of topological similarity for which values above 0.5 generally indicate a shared fold (Zhang & Skolnick, 2004; 2005; Kabsch, 1976). In inverse folding, these metrics are used to assess self-consistency, which evaluates whether a generated sequence folds into the target structure by comparing the structure predicted by a protein folding model with the target backbone. However, both metrics rely on structural alignment, yielding unstable gradients that make them unsuitable as direct training objectives. To overcome this, differentiable structure-based losses such as lDDT (Abramson et al., 2024; Wang et al., 2026) and FAPE (Jumper et al., 2021) have been proposed to provide differentiable structural supervision, based on local distance preservation and frame-based positional errors, respectively. In our work, we leverage distograms as a differentiable proxy for structural consistency, enabling self-consistency to be directly incorporated as a training signal without explicit structure prediction.

3 Preliminaries

3.1 Inverse Folding Model

Protein inverse folding aims to design an amino acid sequence $\mathbf{s} = (s_1, s_2, \ldots, s_L)$ that folds into a given target backbone structure $\mathbf{X}$, where L is the sequence length and each s_i denotes one of the 20 standard amino acid types or an unknown residue. Due to the one-to-many nature of the structure-to-sequence mapping, where multiple distinct sequences can fold into the same structure, inverse folding is commonly formulated as a probabilistic modeling problem that estimates the conditional distribution $p(\mathbf{s} \mid \mathbf{X})$.

We model this distribution using a neural network with learnable parameters θ. With a position-wise factorization, the conditional distribution is written as $p(\mathbf{s} \mid \mathbf{X}; \theta) = \prod_{i=1}^{L} p(s_i \mid \mathbf{X}; \theta)$. The model outputs logits $\mathbf{Z} \in \mathbb{R}^{L \times 21}$, from which residue-wise amino acid probabilities are computed as $p(s_i \mid \mathbf{X}; \theta) = \mathrm{softmax}(\mathbf{Z}_{i,:})$. The predicted sequence is obtained by selecting the most probable amino acid at each position, $\hat{s}_i = \arg\max_s p(s_i \mid \mathbf{X}; \theta)$.

The model is trained by minimizing the cross-entropy loss between the predicted distribution and the native sequence $\mathbf{s}^*$:

$$\mathcal{L}_{\text{seq}} = -\frac{1}{L} \sum_{i=1}^{L} \log p(s_i^* \mid \mathbf{X}; \theta) \tag{1}$$

While this sequence recovery loss encourages the model to recover the ground-truth sequence, optimizing solely for this objective does not ensure that generated sequences will fold into the target structure, motivating the need for additional structural constraints in training.

3.2 Distogram-based Structure Representation

The spatial position of each residue in a protein structure is primarily characterized by its C_β atom, which is the first side-chain atom bonded to the backbone C_α. This atom encodes both the spatial orientation of the residue and the directionality of its side chain. An exception is Glycine, which lacks a side chain and thus uses C_α in place of C_β. Since the pairwise C_β distances between all residues provide a sufficient approximation of the overall three-dimensional protein structure, structural information can be effectively captured without directly operating on 3D atomic coordinates.

Building on this, the distogram is defined as the probability distribution over discretized distance bins for all residue pairs $(i, j), i, j \in \{1, 2, ..., L\}$, in a protein of sequence length L:

$$D_{ij} = p(d_{ij} \in b_k \mid X), k = 1, 2, ..., B \tag{2}$$

where d_{ij} denotes the $C_\beta - C_\beta$ distance between residues i and j, b_k is the k-th distance bin, and B is the total number of bins. This representation offers two key advantages over direct 3D coordinate representations: it is invariant to rotation and translation, and it frames distance prediction as a classification problem, which leads to more stable training. Leveraging these advantages, early protein structure prediction models adopted distograms as a key intermediate representation (Senior et al., 2020; Wang et al., 2022b; Yang et al., 2020; Du et al., 2021).

4 METHOD

4.1 OVERALL FRAMEWORK

Our goal is to provide explicit structural supervision for inverse folding models using the target backbone. One of the most straightforward approaches is to predict structures from inverse folding outputs using a protein folding model and compute a structural loss against the target backbone. We implement this approach as a baseline, referred to as the naïve approach (details and comparisons are provided in Appendix F.1). However, structure prediction and backpropagation through the folding model incur substantial computational and memory costs. We therefore use distograms to provide pairwise geometric supervision without full 3D reconstruction. Specifically, we jointly train an auxiliary module that predicts distograms using representations derived from inverse folding logits. As illustrated in Figure 2, the *Input Representation Module* combines logit-derived and pretrained protein language model features into single and pairwise representations. The *Distogram Prediction Module* processes these representations and positional encodings through a ResNet-based architecture to predict residue-pair distance distributions. DISCO loss supervises these predictions using discretized distance labels obtained from the target backbone's ground-truth distance map. We jointly optimize the inverse folding model and the auxiliary module using the combined sequence recovery and DISCO losses. Gradients from DISCO loss propagate through the logits to the inverse folding model, incorporating target-backbone geometry into training as structural supervision.

4.2 INPUT REPRESENTATION MODULE

Given the per-residue logits $\mathbf{Z} \in \mathbb{R}^{L\times 21}$ output by the inverse folding model, the Input Representation Module constructs two types of representations: a single representation and a pairwise representation, which serve as inputs to the distogram prediction module.

Temperature-Scaled Features. We first apply temperature scaling to the logits to obtain a temperature-scaled amino acid probability distribution for each residue. Specifically, for each residue i, we compute:

$$\tilde{\mathbf{z}}_i = \text{softmax}(\mathbf{Z}_{i,:}/\tau) \tag{3}$$

where $\tau > 0$ is a temperature hyperparameter. Setting τ to a value less than 1 sharpens the distribution, making it approximate a one-hot encoding by concentrating probability mass on the most likely amino acid type. The resulting temperature-scaled distributions $\tilde{\mathbf{Z}} \in \mathbb{R}^{L\times 21}$ serve as the residue type feature and are directly used as a component of the single representation.

To encode pairwise interactions between residues, we compute the outer product of the corresponding amino acid distributions for each pair of positions (i, j), $i, j \in \{1, \ldots, L\}$:

$$\mathbf{P}_{ij} = \tilde{\mathbf{Z}}_i \otimes \tilde{\mathbf{Z}}_j \in \mathbb{R}^{21\times 21} \tag{4}$$

Flattening the last two dimensions yields a pairwise feature tensor $\mathbf{P} \in \mathbb{R}^{L\times L\times 441}$, which explicitly encodes the combinatorial interactions between amino acid type distributions of each residue pair.

Enriching Representations with a Protein Language Model. To complement the logit-based representations with pretrained sequence context, we incorporate sequence representations and attention maps from ESM-2 (Lin et al., 2023), whose parameters remain frozen throughout training. These features provide the auxiliary distogram prediction module with contextual information from the predicted sequence alongside the inverse folding model's output features.

Since ESM-2 takes discrete amino acid tokens as input, we construct a sequence $\hat{\mathbf{s}}$ by selecting the most probable amino acid at each position, $\hat{s}_i = \arg\max_s p(s_i = s \mid \mathbf{X}; \theta)$, and feed it into ESM-2. We extract contextualized residue representations $\mathbf{F}_{\text{ESM}} \in \mathbb{R}^{L\times 1280}$ and linearly project them to $\mathbb{R}^{L\times 128}$. Concatenating these features with the residue-type features $\tilde{\mathbf{Z}}$ yields the *single representation* $\mathbf{H}^{\text{single}} \in \mathbb{R}^{L\times 149}$.

We additionally extract attention maps from the final layer of ESM-2, $\mathbf{A}^{\text{ESM}} \in \mathbb{R}^{L\times L\times 20}$. Protein language model attention maps provide useful information for residue contact prediction (Rao et al., 2020), and we use them as additional pairwise features for distogram prediction. Concatenating them with the outer-product features $\mathbf{P}$ yields the *pair representation* $\mathbf{H}^{\text{pair}} \in \mathbb{R}^{L\times L\times 461}$.

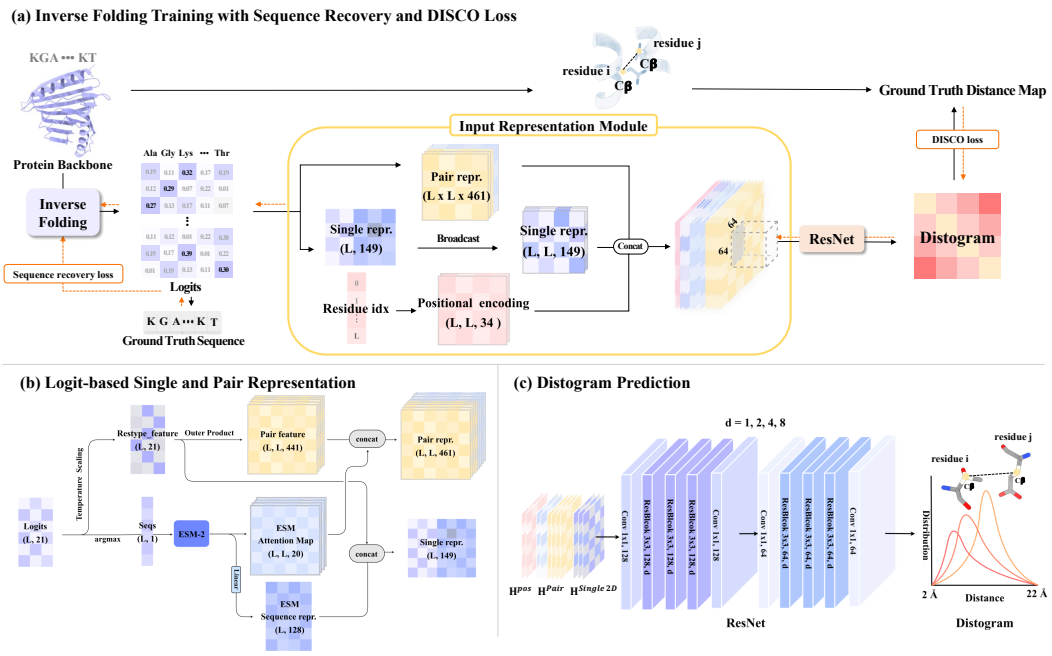

Figure 2: **Overview of training with DISCO loss.** (a) DISCO loss augments inverse folding training by predicting pairwise residue distance distributions from per-residue logits, supervising them with the ground-truth distance map, and propagating structural gradients back to the inverse folding model. (b) Single and pair representations are constructed by combining temperature-scaled logit distributions with ESM-2 sequence representations and attention maps. (c) The ResNet processes the representations through bottleneck residual blocks with cycling dilation rates $d \in \{1, 2, 4, 8\}$.

Expansion of Single Representation. To enable combination with the pair representation $\mathbf{H}^{\text{pair}} \in \mathbb{R}^{L\times L\times 461}$, the single representation $\mathbf{H}^{\text{single}} \in \mathbb{R}^{L\times 149}$ is expanded into an $L \times L$ feature map via x-broadcast and y-broadcast operations, allowing the distogram prediction module to simultaneously access the individual characteristics of both residues in each pair:

$$\mathbf{H}^{x}_{ij,:} = \mathbf{H}^{\text{single}}_{i,:} \in \mathbb{R}^{149}, \quad \mathbf{H}^{y}_{ij,:} = \mathbf{H}^{\text{single}}_{j,:} \in \mathbb{R}^{149}, \quad \forall i, j \in \{1, \ldots, L\} \tag{5}$$

yielding $\mathbf{H}^{x}, \mathbf{H}^{y} \in \mathbb{R}^{L\times L\times 149}$, which are then concatenated to form:

$$\mathbf{H}^{\text{single2D}} = \text{concat}(\mathbf{H}^{x}, \mathbf{H}^{y}) \in \mathbb{R}^{L\times L\times 298} \tag{6}$$

Positional Encoding. To provide the distogram prediction module with explicit positional information about each residue pair, we incorporate three types of positional encodings. Let $N_{\text{scale}} = 100$ denote a normalization factor that scales positional values to a range comparable to typical protein lengths. First, the absolute position of residue j is encoded as $\text{pos}_j = (j - N_{\text{scale}})/N_{\text{scale}}$, which is broadcast along the i direction to form $\mathbf{P}^{\text{pos}} \in \mathbb{R}^{L\times L\times 1}$. Second, the relative sequence distance between residues i and j is encoded as $\text{offset}_{ij} = (i-j)/N_{\text{scale}}$, yielding $\mathbf{P}^{\text{offset}} \in \mathbb{R}^{L\times L\times 1}$. Third, the absolute positions of residues i and j are encoded using K-bit binary representations, where the k-th bit of position n is given by $b_k(n) = \lfloor n/2^k \rfloor \mod 2$, and broadcast to form $\mathbf{B}^{x}, \mathbf{B}^{y} \in \mathbb{R}^{L\times L\times K}$:

$$\mathbf{B}^{x}_{ij,k} = b_k(i), \quad \mathbf{B}^{y}_{ij,k} = b_k(j), \quad \forall i, j \in \{1, \ldots, L\} \tag{7}$$

These encodings are concatenated to form the final positional encoding:

$$\mathbf{H}^{\text{pos}} = \text{concat}(\mathbf{P}^{\text{pos}}, \mathbf{P}^{\text{offset}}, \mathbf{B}^{x}, \mathbf{B}^{y}) \in \mathbb{R}^{L\times L\times(2K+2)} \tag{8}$$

Crop-based Input Construction. The feature maps constructed in the previous steps all have spatial dimensions of $L \times L$. Since the protein length L varies across samples and processing the full $L \times L$ feature map would incur prohibitive computational costs, we adopt a crop-based strategy

with a fixed crop size of 64×64, where the crop position (i, j) is randomly sampled during training. Let $\mathcal{C}_{i,j}(\mathbf{H})$ denote the 64×64 crop of feature map $\mathbf{H}$ at position (i, j), i.e., $\mathbf{H}_{i:i+64,\, j:j+64,\, :}$.

For the pair representation, we extract three crops: the primary crop $\mathcal{C}_{i,j}(\mathbf{H}^{\text{pair}})$ and two diagonal crops $\mathcal{C}_{i,i}(\mathbf{H}^{\text{pair}})$ and $\mathcal{C}_{j,j}(\mathbf{H}^{\text{pair}})$, which provide additional structural context about the individual residue neighborhoods. The final input is obtained by concatenating all feature crops:

$$\mathbf{H}^{\text{input}} = \text{concat}\left(\mathcal{C}_{i,j}(\mathbf{H}^{\text{pair}}),\, \mathcal{C}_{i,i}(\mathbf{H}^{\text{pair}}),\, \mathcal{C}_{j,j}(\mathbf{H}^{\text{pair}}),\, \mathcal{C}_{i,j}(\mathbf{H}^{\text{single2D}}),\, \mathcal{C}_{i,j}(\mathbf{H}^{\text{pos}})\right) \tag{9}$$

yielding $\mathbf{H}^{\text{input}} \in \mathbb{R}^{64\times64\times C_{\text{total}}}$. To ensure sufficient coverage of the full distance map, we sample $N_{\text{crop}} \geq 2$ non-overlapping crops per protein, exposing the model to a diverse set of residue pair interactions and allowing it to incorporate richer structural information from different regions of the distance map into the learning signal.

4.3 Distogram Prediction and DISCO Loss

The distogram prediction module takes $\mathbf{H}^{\text{input}} \in \mathbb{R}^{64\times64\times C_{\text{total}}}$ as input and predicts the pairwise residue distance distribution for each crop. We adopt a ResNet-based architecture for this module, as it is well-suited for processing 2D pairwise feature maps and capturing spatial dependencies at multiple scales. Specifically, dilated convolutions within the residual blocks enable the model to capture both short- and long-range inter-residue interactions within the crop, while skip connections ensure stable gradient flow back to the inverse folding model during training. The final output is passed through a distogram head consisting of a 1×1 convolution, producing the predicted distogram $\hat{D}_{ij} \in \mathbb{R}^{64\times64\times B}$ for each crop, where $B = 64$ is the number of uniformly spaced distance bins covering the range $[2.0\text{Å}, 22.0\text{Å}]$. (Appendix A.1.2 provides further details.)

The module is trained by minimizing the cross-entropy loss between the predicted distogram and the ground-truth labels D_{ij} obtained by discretizing pairwise distances in the input backbone $\mathbf{X}$:

$$\mathcal{L}_{\text{DISCO}} = -\frac{1}{\sum_{i,j} m_{ij}} \sum_{i=1}^{H}\sum_{j=1}^{W} m_{ij} \sum_{b=1}^{B} \mathbf{1}[d_{ij} \in \text{bin}_b] \log \hat{p}_{ij,b} \tag{10}$$

where d_{ij} denotes the C_β-C_β distance between residues i and j (C_α for Glycine), $\hat{p}_{ij,b}$ is the predicted probability for bin b, and $m_{ij} = m_i \cdot m_j$ is a pairwise validity mask. The auxiliary module's input representation combines differentiable logit-based features with ESM-2 features extracted from the sequence obtained via $\arg\max$. Both contribute to distogram prediction, but $\arg\max$ blocks gradient propagation through the ESM-2 pathway, so gradients of $\mathcal{L}_{\text{DISCO}}$ reach the inverse folding model only through the logit-based pathway.

The total training loss is defined as a weighted sum of the sequence recovery loss $\mathcal{L}_{\text{seq}}$ and the DISCO loss $\mathcal{L}_{\text{DISCO}}$:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seq}} + w_{DISCO} \cdot \mathcal{L}_{\text{DISCO}} \tag{11}$$

where w_{DISCO} controls the contribution of DISCO loss and $\mathcal{L}_{\text{seq}}$ is the sequence recovery loss defined in Section 3. This combined objective jointly trains the inverse folding model and the auxiliary module to recover native sequences and predict target-backbone distances, introducing explicit geometric supervision into inverse folding training.

5 Experiments

5.1 Evaluation Settings

To evaluate DISCO loss, we train autoregressive ProteinMPNN, non-autoregressive PiFold, and diffusion-based Grade-IF with and without DISCO loss, following the settings in Appendix A.2. We assess structural self-consistency and long-range contact recovery to examine compatibility with target backbones and improvements at the residue-pair level (Sections 5.2 and 5.3). We evaluate sequence recovery and perplexity on standard benchmarks (Section 5.4) and conduct ESM ablations to assess the contribution of ESM features to the performance gains with DISCO loss (Section 5.5). The Appendix compares DISCO loss, AFDistill, and the naïve approach as alternative ways of providing structural supervision in terms of performance and computational efficiency. It also presents probes of structural information in inverse folding logits, ablations on hyperparameters and crop settings, and analyses of generated sequence properties.

Table 1: Median self-consistency on RFdiffusion-generated *de novo* backbones (grouped by sequence length L), evaluated with AlphaFold2 and ESMFold. **Bold**: models trained with $\mathcal{L}_{\text{DISCO}}$; underlined: baselines; red: improvement over baseline. Higher scTM and pLDDT (↑) and lower scRMSD (↓) indicate better performance.

AlphaFold2

Model	$L = 100$			$L = 200$			$L = 300$			$L = 400$			$L = 500$		
	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT
ESM-IF	0.506	10.363	75.77	0.361	17.288	47.08	0.322	21.165	44.20	0.271	24.694	40.88	0.259	27.425	40.88
FAMPNN	0.960	0.784	92.11	0.965	1.104	91.40	0.944	1.922	87.30	0.699	9.789	71.95	0.560	16.503	62.15
ProteinMPNN	0.953	0.866	91.32	0.964	1.119	90.16	0.936	2.110	86.03	0.736	8.556	72.14	0.575	15.783	63.39
PiFold	0.686	4.598	85.09	0.564	11.743	63.16	0.487	16.246	58.52	0.368	22.141	50.47	0.355	24.658	48.72
Grade-IF	0.323	16.041	63.53	0.214	22.411	42.35	0.194	26.473	41.47	0.184	29.615	39.70	0.168	33.760	38.90
ProteinMPNN + $\mathcal{L}_{\text{DISCO}}$	**0.952** (↓0.001)	**0.875** (↑0.009)	**91.02** (↓0.30)	**0.967** (↑0.003)	**1.038** (↓0.081)	**90.62** (↑0.46)	**0.955** (↑0.019)	**1.643** (↓0.467)	**87.39** (↑1.36)	**0.839** (↑0.103)	**5.455** (↓3.101)	**76.10** (↑3.96)	**0.618** (↑0.043)	**14.480** (↓1.303)	**64.28** (↑0.89)
PiFold + $\mathcal{L}_{\text{DISCO}}$	**0.873** (↑0.187)	**1.568** (↓3.030)	**86.59** (↑1.50)	**0.795** (↑0.231)	**4.144** (↓7.599)	**75.98** (↑12.82)	**0.685** (↑0.198)	**8.647** (↓7.599)	**67.05** (↑8.53)	**0.505** (↑0.137)	**16.342** (↓5.799)	**56.40** (↑5.93)	**0.442** (↑0.087)	**20.636** (↓4.022)	**51.43** (↑2.71)
Grade-IF + $\mathcal{L}_{\text{DISCO}}$	**0.347** (↑0.024)	**13.947** (↓2.094)	**61.75** (↓1.78)	**0.251** (↑0.037)	**19.450** (↓2.961)	**41.91** (↓0.44)	**0.241** (↑0.047)	**22.753** (↓3.720)	**39.80** (↓1.67)	**0.225** (↑0.041)	**25.293** (↓4.322)	**38.50** (↓1.20)	**0.217** (↑0.049)	**27.823** (↓5.937)	**37.50** (↓1.40)

ESMFold

Model	$L = 100$			$L = 200$			$L = 300$			$L = 400$			$L = 500$		
	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT
ESM-IF	0.514	10.909	62.06	0.355	16.261	39.22	0.342	18.659	37.48	0.280	21.678	33.83	0.277	23.551	32.09
FAMPNN	0.968	0.687	86.28	0.944	1.521	81.54	0.910	2.616	76.78	0.594	12.193	56.20	0.509	16.623	49.91
ProteinMPNN	0.958	0.798	84.85	0.953	1.324	80.03	0.904	2.837	74.87	0.605	11.703	56.46	0.513	16.599	49.97
PiFold	0.733	4.358	74.04	0.552	11.454	51.07	0.475	14.621	46.80	0.356	19.401	39.23	0.331	22.291	37.20
Grade-IF	0.331	13.580	46.76	0.234	18.706	32.74	0.222	20.877	30.86	0.216	22.871	28.18	0.214	24.182	25.49
ProteinMPNN + $\mathcal{L}_{\text{DISCO}}$	**0.959** (↑0.001)	**0.803** (↑0.005)	**90.67** (↑5.82)	**0.957** (↑0.004)	**1.248** (↓0.076)	**86.39** (↑6.36)	**0.930** (↑0.026)	**2.144** (↓0.693)	**82.28** (↑7.41)	**0.625** (↑0.020)	**10.686** (↓1.017)	**62.06** (↑5.60)	**0.533** (↑0.020)	**15.848** (↓0.751)	**55.03** (↑5.06)
PiFold + $\mathcal{L}_{\text{DISCO}}$	**0.857** (↑0.124)	**1.829** (↓2.529)	**80.92** (↑6.88)	**0.658** (↑0.106)	**8.333** (↓3.121)	**62.73** (↑11.66)	**0.585** (↑0.110)	**10.966** (↓3.655)	**54.82** (↑8.02)	**0.418** (↑0.062)	**17.242** (↓2.159)	**46.26** (↑7.03)	**0.392** (↑0.061)	**20.468** (↓1.823)	**42.61** (↑5.41)
Grade-IF + $\mathcal{L}_{\text{DISCO}}$	**0.359** (↑0.028)	**12.720** (↓0.860)	**50.07** (↑3.31)	**0.260** (↑0.026)	**18.120** (↓0.586)	**35.53** (↑2.79)	**0.259** (↑0.037)	**19.936** (↓0.941)	**33.62** (↑2.76)	**0.243** (↑0.027)	**22.067** (↓0.804)	**30.80** (↑2.62)	**0.241** (↑0.027)	**23.896** (↓0.286)	**28.91** (↑3.42)

5.2 Self-Consistency

To evaluate whether the structural supervision provided by DISCO loss translates into improved structural self-consistency, we generate 8 sequences for each of 100 backbones at each length $L \in \{100, 200, 300, 400, 500\}$. We measure structural agreement primarily using self-consistency TM-score (scTM) and RMSD (scRMSD), with pLDDT additionally reported as a measure of prediction confidence. Detailed metric definitions are provided in Appendix B.1.5.

As shown in Table 1, DISCO loss generally increases scTM and reduces scRMSD across ProteinMPNN, PiFold, and Grade-IF. PiFold improves in both metrics across all lengths under both AlphaFold2 (AF2) and ESMFold. At $L = 200$ under AF2, scTM increases from 0.564 to 0.795, while scRMSD decreases from 11.743 to 4.144 Å. ProteinMPNN improves in both metrics at $L \geq 200$ under both folding models, with an scTM increase of 0.103 and an scRMSD reduction of 3.101 Å at $L = 400$ under AF2. At $L = 100$, its changes are small and mixed. Grade-IF also improves in both metrics across all lengths under both AF2 and ESMFold. Together, these results indicate that the structural supervision provided by DISCO loss translates into improved structural self-consistency of generated sequences across inverse folding models with distinct generation mechanisms, as evaluated using both folding models. Additional comparisons with alternative structural supervision methods are provided in Appendix F.

5.3 Long-range Contact Analysis

DISCO loss supervises residue-pair distance predictions using target-backbone geometry, incorporating inter-residue spatial relationships into training. To examine whether this supervision also improves long-range contact recovery, we analyze the same generated sequences and AF2-predicted structures used in the self-consistency evaluation (Section 5.2). We measure contact F1 against the target backbone across multiple sequence separation ranges. Additional evaluation using ESMFold is provided in Appendix D.1.1.

As shown in Table 2, DISCO loss improves contact F1 across all separation ranges for all three inverse folding models, with paired bootstrap analysis supporting these improvements. At $|i - j| \geq 50$, F1 gains are 0.023 and 0.127 for ProteinMPNN and PiFold, respectively. Grade-IF also improves across all ranges, with its largest gain of 0.051 at $24 \leq |i - j| < 50$.

Table 2: Long-range contact F1 on AF2-predicted structures by sequence separation $|i-j|$. Scores are averaged over eight sequences per backbone, then across backbones. ΔF1 denotes the change from baseline with DISCO loss. Percentile 95% CIs use 10,000 paired backbone bootstrap resamples, stratified by length ($n_{\text{bb}} = 500$; $n_{\text{bb}} = 485$ for $|i-j| \geq 50$, excluding backbones without such contacts).

Model	$\lvert i-j\rvert$	Baseline	$+\mathcal{L}_{\text{DISCO}}$	ΔF1 [95% CI]
ProteinMPNN	12–24	0.601	**0.612**	$+\mathbf{0.012}$ [+0.007, +0.016]
	24–50	0.631	**0.644**	$+\mathbf{0.013}$ [+0.008, +0.018]
	≥ 50	0.464	**0.487**	$+\mathbf{0.023}$ [+0.015, +0.031]
PiFold	12–24	0.436	**0.519**	$+\mathbf{0.083}$ [+0.071, +0.095]
	24–50	0.430	**0.529**	$+\mathbf{0.099}$ [+0.086, +0.113]
	≥ 50	0.189	**0.315**	$+\mathbf{0.127}$ [+0.111, +0.143]
Grade-IF	12–24	0.098	**0.145**	$+\mathbf{0.048}$ [+0.042, +0.054]
	24–50	0.064	**0.116**	$+\mathbf{0.051}$ [+0.046, +0.057]
	≥ 50	0.013	**0.026**	$+\mathbf{0.013}$ [+0.010, +0.016]

We further examine the relationship between these contact recovery gains and improvements in structural self-consistency using Spearman correlations between backbone-level ΔF1 and ΔscTM. At $|i-j| \geq 50$, the correlations are 0.761, 0.760, and 0.407 for ProteinMPNN, PiFold, and Grade-IF, respectively, indicating that backbones with larger contact recovery gains tend to show larger scTM improvements. Detailed analyses are provided in Appendix D.1.2.

5.4 Sequence Recovery & Perplexity

Table 3: Median sequence recovery (%) and perplexity on CATH 4.2, TS50, and TS500. **Bold values** mark the best result per column. ProteinMPNN, PiFold, and Grade-IF are evaluated under our setup with and without $\mathcal{L}_{\text{DISCO}}$; other results are taken from (Gao et al., 2023).

	CATH 4.2		TS50		TS500	
Model	**Perp.** (↓)	**Rec. %** (↑)	**Perp.** (↓)	**Rec. %** (↑)	**Perp.** (↓)	**Rec. %** (↑)
GCA	6.05	37.64	5.09	47.02	4.72	47.74
GVP	5.36	39.47	4.71	44.14	4.20	49.14
AlphaDesign	6.30	41.31	5.25	48.36	4.93	49.23
ProteinMPNN	5.41	45.97	5.12	44.31	4.62	47.65
PiFold	4.55	51.66	3.86	58.72	3.44	60.42
Grade-IF	**4.35**	**52.05**	4.03	53.74	3.71	56.85
ProteinMPNN $+\mathcal{L}_{\text{DISCO}}$	5.42	42.09	5.22	42.31	4.60	47.85
PiFold $+\mathcal{L}_{\text{DISCO}}$	4.48	51.91	**3.60**	**58.83**	**3.29**	**61.32**
Grade-IF $+\mathcal{L}_{\text{DISCO}}$	4.41	51.76	4.08	53.09	3.79	55.77

We also evaluate the extent to which performance on the original sequence learning task is maintained when training with DISCO loss. Table 3 reports perplexity and sequence recovery on CATH 4.2, TS50, and TS500. Training with DISCO loss improves both metrics for PiFold across all three benchmarks, while Grade-IF shows perplexity increases of 0.05–0.08 and recovery decreases of 0.29–1.08 percentage points. ProteinMPNN largely retains its perplexity, although recovery decreases by 3.88 and 2.00 percentage points on CATH 4.2 and TS50, respectively, while increasing slightly on TS500. Higher native sequence recovery does not necessarily imply higher structural self-consistency (Ye et al., 2025): Grade-IF achieves higher CATH 4.2 recovery than ProteinMPNN both with and without DISCO loss, but lower scTM on de novo backbones.

5.5 Ablation Studies

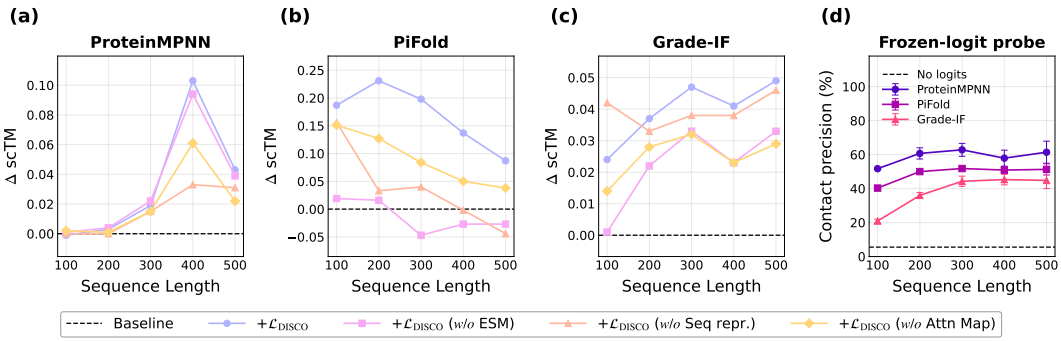

Figure 3: ESM feature ablation for **(a)** ProteinMPNN, **(b)** PiFold, and **(c)** Grade-IF. *w/o* ESM, Seq repr., and Attn Map remove both features, sequence representations, and attention maps, respectively. AF2-based ΔscTM is measured relative to baselines trained without DISCO loss (dashed lines). **(d)** Long-range contact precision (top-$L/5$) of frozen-logit probes on RFdiffusion backbones, with the dashed line indicating the positional control.

Section 5.2 shows that DISCO loss improves structural self-consistency. The auxiliary distogram prediction module uses ESM sequence representations and attention maps alongside logit-derived representations. We assess their contributions to these improvements by comparing the full configuration with variants that remove the ESM sequence representations, the attention maps, or both.

Figure 3(a–c) shows that ESM features contribute differently across models. For PiFold, adding ESM increases scTM from 0.580 to 0.795 at $L = 200$, while the variant without ESM features falls below baseline at $L \geq 300$. Retaining sequence representations alone outperforms retaining attention maps alone at $L \geq 200$, and combining both yields the highest scTM across all lengths. Grade-IF benefits from ESM across all evaluated lengths, with scTM increasing from 0.201 to 0.217 at $L = 500$. ProteinMPNN retains substantial improvements without ESM; at $L = 400$, adding ESM further increases scTM from 0.830 to 0.839.

ProteinMPNN retains substantial self-consistency gains without ESM, suggesting that its logits may already contain useful structural information. We examine this possibility using frozen baselines trained with their original sequence-learning objectives. For each baseline, we train a newly initialized probe with the DISCO auxiliary architecture, excluding ESM features. All probes are trained using only DISCO loss under a common protocol.

On RFdiffusion backbones, long-range top-$L/5$ contact precision, evaluated on the $L/5$ highest-probability long-range pairs, is $58.9 \pm 3.7\%$ for ProteinMPNN, $48.9 \pm 1.2\%$ for PiFold, and $38.2 \pm 2.2\%$ for Grade-IF across three probe seeds, compared with $5.5 \pm 1.3\%$ for the positional control (Figure 3(d)). ProteinMPNN's higher precision indicates that its logits support more accurate contact prediction without ESM. This provides evidence that its logits already contain useful structural cues for the auxiliary module, consistent with the substantial self-consistency gains retained without ESM features. ProteinMPNN also achieves the highest probe contact precision on natural protein structures. Additional results and probe details are provided in Appendix C.

6 Conclusions

We propose DISCO loss, a distogram-based auxiliary objective that complements sequence recovery with structural supervision. A jointly trained auxiliary module predicts residue-pair distance distributions using logit-derived representations, supervised by discretized target-backbone distances. This provides structural supervision without full 3D reconstruction, separate surrogate pretraining, or additional inference cost. DISCO loss improves structural self-consistency and long-range contact recovery across distinct inverse folding paradigms on de novo backbones, while largely maintaining sequence recovery performance and diversity. Lower Rosetta energies and improved amino acid composition further suggest gains in generated sequence quality beyond structural similarity. Experimental validation of protein stability and function remains a direction for future work.

AI USE STATEMENT

We used generative AI tools to assist with translation, language editing, LaTeX and table formatting, and discussions of experimental analyses and result interpretation.

ETHICS STATEMENT

This work aims to improve training methods for inverse folding models to generate protein sequences consistent with target structures. Structural self-consistency and other computational metrics do not establish the biological safety or functionality of generated sequences. Any downstream experimental use requires appropriate validation and biosafety assessment.

REPRODUCIBILITY STATEMENT

Training settings for the inverse folding models are provided in Appendix A.2, and the structural self-consistency evaluation protocol is described in Appendix B. Details of hyperparameter selection and ablation studies are provided in Appendix C.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Rebecca F Alford, Andrew Leaver-Fay, Jeliazko R Jeliazkov, Matthew J O'Meara, Frank P DiMaio, Hahnbeom Park, Maxim V Shapovalov, P Douglas Renfrew, Vikram K Mulligan, Kalli Kappel, et al. The Rosetta all-atom energy function for macromolecular modeling and design. *Journal of Chemical Theory and Computation*, 13(6):3031–3048, 2017.

Nicholas Bhattacharya, Neil Thomas, Roshan Rao, Justas Dauparas, Peter K Koo, David Baker, Yun S Song, and Sergey Ovchinnikov. Single layers of attention suffice to predict protein contacts. *Biorxiv*, pp. 2020–12, 2020.

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.

Zongyang Du, Hong Su, Wenkai Wang, Lisha Ye, Hong Wei, Zhenling Peng, Ivan Anishchenko, David Baker, and Jianyi Yang. The trrosetta server for fast and accurate protein structure prediction. *Nature protocols*, 16(12):5634–5651, 2021.

Zhangyang Gao, Cheng Tan, and Stan Z. Li. Pifold: Toward effective and efficient protein inverse folding. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=oMsN9TYwJ0j.

Thomas 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. *Science*, 387(6736):850–858, 2025.

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*, pp. 8946–8970. PMLR, 2022.

John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

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*, 2021. URL https://openreview.net/forum?id=1YLJDvSx6J4.

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*, 32(5):922–923, 1976.

Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

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.

Igor Melnyk, Aurélie Lozano, Payel Das, and Vijil Chenthamarakshan. Alphafold distillation for inverse protein design. *Scientific Reports*, 15(1):21743, 2025.

Lucas G Nivon, Rocco Moretti, and David Baker. A Pareto-optimal refinement method for protein design scaffolds. *PLoS ONE*, 8(4):e59004, 2013.

Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *Biorxiv*, pp. 2020–12, 2020.

Milong Ren, Tian Zhu, and Haicang Zhang. Carbonnovo: Joint design of protein structure and sequence using a unified energy-based model. In *Forty-first international conference on machine learning*, 2024.

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.

Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

A Shrake and JA Rupley. Environment and exposure to solvent of protein atoms. Lysozyme and insulin. *Journal of Molecular Biology*, 79(2):351–371, 1973.

Jaspreet Singh, Thomas Litfin, Jaswinder Singh, Kuldip Paliwal, and Yaoqi Zhou. Spot-contact-lm: improving single-sequence-based prediction of protein contact map using a transformer language model. *Bioinformatics*, 38(7):1888–1894, 2022.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, pp. 369–386. SPIE, 2019.

Matthew Z Tien, Austin G Meyer, Dariya K Sydykova, Stephanie J Spielman, and Claus O Wilke. Maximum allowed solvent accessibilites of residues in proteins. *PLoS ONE*, 8(11):e80635, 2013.

Brian L Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations*, 2023.

Michael D Tyka, Daniel A Keedy, Ingemar André, Frank DiMaio, Yifan Song, David C Richardson, Jane S Richardson, and David Baker. Alternate states of proteins revealed by detailed energy landscape mapping. *Journal of Molecular Biology*, 405(2):607–618, 2011.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jue Wang, Sidney Lisanza, David Juergens, Doug Tischer, Joseph L Watson, Karla M Castro, Robert Ragotte, Amijai Saragovi, Lukas F Milles, Minkyung Baek, et al. Scaffolding protein functional sites using deep learning. *Science*, 377(6604):387–394, 2022a.

Wenkai Wang, Zhenling Peng, and Jianyi Yang. Single-sequence protein structure prediction using supervised transformer protein language models. *Nature Computational Science*, 2(12):804–814, 2022b.

Yuyang Wang, Jiarui Lu, Navdeep Jaitly, Joshua M. Susskind, and Miguel Ángel Bautista. Simplefold: Folding proteins is simpler than you think. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0j0MmK7EMA.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Talal Widatalla, Richard W. Shuai, Brian Hie, and Possu Huang. Sidechain conditioning and modeling for full-atom protein sequence design with FAMPNN. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EE8Z8Se5Te.

Junde Xu, Zijun Gao, Xinyi Zhou, hujie, Xingyi Cheng, Le Song, Guangyong Chen, Pheng-Ann Heng, and Jiezhong Qiu. Protein inverse folding from structure feedback. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=ORsrbGTXQB.

Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, 117(3):1496–1503, 2020.

Fei Ye, Zaixiang Zheng, Dongyu Xue, Yuning Shen, Lihao Wang, Yiming Ma, Yan Wang, Xinyou Wang, Xiangxin Zhou, and Quanquan Gu. ProteinBench: A holistic evaluation of protein foundation models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Kai Yi, Bingxin Zhou, Yiqing Shen, Pietro Liò, and Yuguang Wang. Graph denoising diffusion for inverse protein folding. *Advances in Neural Information Processing Systems*, 36:10238–10257, 2023.

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.

Yang Zhang and Jeffrey Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33(7):2302–2309, 2005.

APPENDIX

CONTENTS

A Implementation and Training Details **14**

A.1 DISCO Loss 14

A.2 Training Details 15

B Evaluation Protocols **16**

B.1 Structural Evaluation 16

B.2 Sequence Evaluation 18

C Ablation Studies and Analysis **19**

C.1 Hyperparameter Sensitivity 19

C.2 Effects of Crop Number and Composition 20

C.3 ESM Feature Ablation 21

C.4 Probing Structural Information in Inverse Folding Logits 22

D Additional Structural Evaluation **23**

D.1 Long-Range Contact Analysis 24

D.2 Secondary Structure Evaluation 25

E Sequence Performance and Diversity **26**

E.1 CATH 4.2 Benchmark 26

E.2 Sequence Diversity 27

F Comparison with Alternative Structural Supervision **28**

F.1 Naïve Approach to Structural Supervision 28

F.2 Structural Supervision with AFDistill 30

F.3 Comparison of Performance and Computational Cost 30

G Amino Acid Composition and Energy Analyses **31**

G.1 Amino Acid Composition Analysis 31

G.2 Rosetta Energy Evaluation 33

H Code Availability **34**

A Implementation and Training Details

This section details the ESM-2 features, distogram prediction module, and training configurations used in our experiments.

A.1 DISCO Loss

A.1.1 ESM-2 Features

We use ESM-2 (Lin et al., 2023) to enrich the input representation of the distogram prediction module with pretrained evolutionary information. Specifically, we use the `esm2_t33_650M_UR50D` model (33 layers, 650M parameters, hidden dimension 1280) pretrained on UniRef50 (2021_04).

We extract attention maps from the final (33rd) layer only. Prior work (Rao et al., 2020; Bhattacharya et al., 2020) reports only marginal performance differences between using attention maps from all layers and from the final layer alone; we therefore adopt the latter for efficiency.

Although ESM-2 remains frozen, its features are recomputed at each training step using the discrete sequence obtained by applying $\arg\max$ to the inverse folding logits. Its sequence representations and attention maps therefore reflect the model's current sequence prediction. Since $\arg\max$ is non-differentiable, gradients from $\mathcal{L}_{\text{DISCO}}$ reach the inverse folding model only through the continuous amino acid probability pathway, not through the ESM-2 branch.

A.1.2 Distogram Prediction Module

The distogram prediction module follows a 2D dilated ResNet architecture commonly used in distogram and contact prediction (Senior et al., 2020; Singh et al., 2022). The module predicts residue-pair distance distributions using logit-derived features. The prediction loss against discretized target-backbone distances provides pairwise structural supervision to the inverse folding model. Following AlphaFold1, we adopt a crop-based training scheme and a ResNet-style architecture. As the module is used only for auxiliary supervision, it is designed to be compact. The full set of hyperparameters is summarized in Table A1.

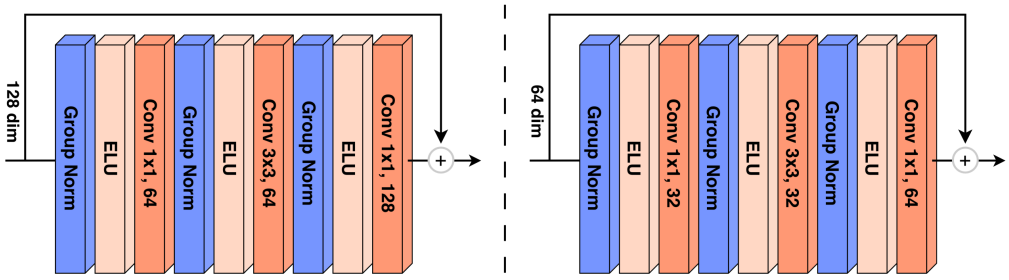

Figure A1: Bottleneck residual blocks in the distogram prediction module. Each block follows a pre-activation design: GroupNorm $\rightarrow$ ELU $\rightarrow$ Conv 1×1 $\rightarrow$ GroupNorm $\rightarrow$ ELU $\rightarrow$ Conv 3×3 (with dilation) $\rightarrow$ GroupNorm $\rightarrow$ ELU $\rightarrow$ Conv 1×1, with a residual skip connection. (a) First block group: 128-dim input, with the bottleneck reducing to 64 channels before restoring to 128. (b) Second block group: 64-dim input, with the bottleneck reducing to 32 channels before restoring to 64.

Two-group design. The network consists of two block groups, with progressively reduced channel dimensions. The first group compresses the high-dimensional input into a smaller representation space: a 1×1 convolution projects the 1715 input channels to $C_1 = 128$, followed by 12 bottleneck residual blocks and a final 1×1 convolution. The second group further compresses the representation: a 1×1 convolution projects the 128 channels to $C_2 = 64$, followed by another 12 residual blocks and a final 1×1 convolution. In total, the network comprises 24 residual blocks. Unlike conventional ResNet designs, the channel dimension decreases with depth. Given the auxiliary role of

the module, this design reduces computational cost while retaining sufficient capacity to propagate the supervision signal.

Bottleneck residual block. Each group contains 12 bottleneck residual blocks, preceded and followed by 1×1 projection layers. Each block follows a pre-activation design (Figure A1): a 1×1 convolution reduces the channel dimension by half, a 3×3 convolution extracts spatial features, and a final 1×1 convolution restores the original dimension. The dilation rate d cycles through $\{1, 2, 4, 8\}$ across blocks, enabling multi-scale receptive fields. We use GroupNorm (32 groups) instead of InstanceNorm to mitigate instability arising from varying protein lengths and batch sizes. ELU is used as the activation function.

Distogram head. The output of the second block group is passed through a final 1×1 convolution to produce logits over $B = 64$ distance bins, defined by uniformly partitioning the range $[2.0, 22.0]$ Å.

Crop sampling. During training, four 64×64 crops are randomly sampled at each step and used jointly, balancing memory efficiency with broad coverage of pairwise residue regions.

Table A1: Hyperparameters of the distogram prediction module. The module consists of two block groups with progressively reduced channel dimensions ($128 \rightarrow 64$) and is used only during training; it is removed at inference.

Hyperparameter	Value
Channel dimension (C_1 / C_2)	128 / 64
Residual blocks per group	12
Total residual blocks	24
Single block structure	1×1, 3×3, 1×1
Crop size	64×64
Crops per step	4
Number of distance bins (B)	64
Distance range	$[2.0, 22.0]$ Å
Dilation cycle	$\{1, 2, 4, 8\}$
Normalization	GroupNorm (32 groups)
Activation	ELU

A.2 Training Details

A.2.1 Training datasets

For a fair comparison with the baselines (ProteinMPNN, PiFold, and Grade-IF), we follow the training datasets used in their original papers.

CATH 4.2 For training PiFold and Grade-IF, we use the S40 split of CATH 4.2. CATH is a structural classification database of protein domains derived from the Protein Data Bank (PDB). The S40 split (Ingraham et al., 2019) excludes domains with pairwise sequence identity above 40% and restricts sequence lengths to 60–500 residues.

PDB For training ProteinMPNN, we use biological assembly (biounit) data extracted from the Protein Data Bank (PDB). Each biounit consists of multiple chains and their corresponding rigid-body transformations (rotation and translation), representing the functional oligomeric state of the protein. The dataset includes structures released before August 2, 2021, with redundancy removed by clustering sequences at 30% identity. Train/validation/test splits are constructed at the cluster level to avoid overlap across splits. During training, a representative chain is randomly sampled from each cluster at every epoch to increase data diversity and prevent overfitting.

A.2.2 Training Hyperparameters

This section describes the additional hyperparameters used to train the distogram module. We consider three hyperparameters: the logit temperature τ, the distogram loss weight λ, and the number of crops per batch. These values are selected separately for ProteinMPNN, PiFold, and Grade-IF through model-specific hyperparameter sweeps. In particular, τ is tuned per model to account for differences in their output logit distributions.

ProteinMPNN We follow the original training configuration of ProteinMPNN. Adam (Kingma & Ba, 2015) is used as the optimizer, with NoamOpt (Vaswani et al., 2017) as the learning rate scheduler (warmup steps = 4,000, factor = 2, $d_{\text{model}} = 128$). The batch size is 10,000 tokens, and the model is trained for 150 epochs. For the distogram module, we set $\tau = 1.0$, $\lambda = 0.5$, and the number of crops per batch to 4. On a single NVIDIA RTX PRO 6000 GPU, training takes 27.8 hours for the baseline and 54.6 hours with DISCO loss.

PiFold We follow the original training configuration of PiFold. Adam is used as the optimizer, with OneCycleLR (Smith & Topin, 2019) as the learning rate scheduler (maximum learning rate = 1×10^{-3}). The batch size is 8 structures, and the model is trained for 100 epochs. For the distogram module, we set $\tau = 0.1$, $\lambda = 1.0$, and the number of crops per batch to 4. On a single NVIDIA RTX PRO 6000 GPU, training takes about 15 hours for the baseline and 16.5 hours with DISCO loss.

Grade-IF We follow the original Grade-IF configuration with Adam ($\beta = (0.9, 0.99)$, weight decay = 10^{-5}), a learning rate of 5×10^{-4} reduced tenfold halfway through training, and global gradient norm clipping at 1.0. Training uses 200 epochs, batches of 64 structures, 500 diffusion steps, and the BLOSUM transition matrix. EMA weights (decay = 0.995) are used for evaluation. The distogram module uses $\tau = 1.0$, $\lambda = 1.0$, and 4 crops per batch. On a single NVIDIA RTX PRO 6000 GPU, training takes 2.4 hours for the baseline and 22.3 hours with DISCO loss.

B Evaluation Protocols

This section details the protocols for structural and sequence evaluation, including datasets, evaluation settings, metrics, and statistical analyses.

B.1 Structural Evaluation

Designing sequences compatible with newly generated backbones is a practical task in *de novo* protein design. We therefore evaluate structural self-consistency on RFdiffusion-generated backbones to assess whether DISCO loss improves the ability of inverse folding models to generate sequences whose predicted structures match the intended designs.

B.1.1 De Novo Backbone Generation

De novo backbones are generated using the base checkpoint and default parameters from the official RFdiffusion repository. We follow the unconditional monomer protocol, fixing sequence lengths to $\{100, 200, 300, 400, 500\}$ via contig specifications. We generate 100 backbones for each length, yielding 500 backbones in total.

B.1.2 Sequence Sampling

For each backbone, we generate eight sequences at a fixed sampling temperature of 0.1, yielding 4,000 sequences per model configuration across 500 backbones. The same sampling settings are used for models trained with and without DISCO loss.

B.1.3 Structure Prediction Models

We use two protein structure prediction models with different architectures, AlphaFold2 and ESMFold, to assess the consistency of the evaluation results.

AlphaFold2. AlphaFold2 is a high-accuracy protein structure prediction model, which we use via ColabFold for inference. Since no MSA is constructed, structures are predicted in single-sequence mode; all parameters, including the number of recycling steps, follow the default settings.

ESMFold. ESMFold is an ESM-2-based, MSA-free protein structure prediction model that enables fast inference, making it suitable for large-scale self-consistency evaluation. We use the official ESMFold-v1 implementation with default settings.

B.1.4 Comparison Models

We additionally evaluate ESM-IF and FAMPNN using the default checkpoints and settings from their official implementations.

ESM-IF. An inverse folding model trained with augmentation from AlphaFold2-predicted structures. Sequences are generated using the `sample_sequences.py` script.

FAMPNN. A full-atom inverse folding model that jointly models sequence and side-chain structure. Sequences are generated using the default checkpoint.

B.1.5 Self-Consistency Evaluation

For each generated sequence, we compute scTM, scRMSD, and mean pLDDT, and then take the median across the eight sequences generated for each backbone. scTM and scRMSD are computed using the official TM-score implementation (Zhang & Skolnick, 2004), while pLDDT is obtained directly from the protein folding model. For each sequence length, we report the median of the backbone-level scores.

scTM. The TM-score between the original backbone and the structure predicted from the generated sequence, defined as

$$\text{TM-score} = \max \left[\frac{1}{L_{\text{target}}} \sum_{i=1}^{L_{\text{common}}} \frac{1}{1 + (d_i / d_0(L_{\text{target}}))^2} \right],$$

where L_{target} is the length of the target structure, L_{common} is the number of residues appearing in both structures, d_i is the C_α–C_α distance between the i-th aligned residue pair, and $d_0(L_{\text{target}}) = 1.24 \sqrt[3]{L_{\text{target}} - 15} - 1.8$ is a length-dependent distance scale. The maximum is taken over all possible structural superpositions. Values range from 0 to 1, with values above 0.5 generally indicating that the two structures share the same topology.

scRMSD. The C_α root-mean-square deviation between the original backbone and the predicted structure after structural superposition, defined as

$$\text{scRMSD} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}_i^{\text{pred}} - \mathbf{x}_i^{\text{ref}}\|^2},$$

where $\mathbf{x}_i^{\text{pred}}$ and $\mathbf{x}_i^{\text{ref}}$ denote the C_α coordinates of residue i in the predicted and reference structures, and N is the number of residues. Lower values indicate greater structural similarity.

pLDDT. pLDDT is a per-residue confidence score (0–100) output by the protein folding model. A predicted structure may exhibit high similarity to the target backbone while receiving a low pLDDT, in which case the structural similarity should be interpreted with caution. Conversely, a high pLDDT may indicate a confident but off-target fold. We therefore report the mean pLDDT as a complementary confidence metric alongside scTM and scRMSD:

$$\text{pLDDT} = \frac{1}{N} \sum_{i=1}^{N} \text{pLDDT}_i,$$

where pLDDT_i denotes the confidence score assigned to residue i.

B.1.6 LONG-RANGE CONTACT RECOVERY

A long-range contact is defined as a residue pair with a C_β–C_β distance (C_α for glycine) ≤ 8 Å and sequence separation $|i-j| \geq 12$. Let C_{true} denote the set of long-range contacts in the RFdiffusion backbone and C_{pred} the corresponding set in the structure predicted by AlphaFold2 or ESMFold. Precision, recall, and F1-score are defined as:

$$\text{Precision} = \frac{|C_{\text{pred}} \cap C_{\text{true}}|}{|C_{\text{pred}}|}, \quad \text{Recall} = \frac{|C_{\text{pred}} \cap C_{\text{true}}|}{|C_{\text{true}}|}, \quad \text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Precision, recall, and F1 are set to zero whenever their respective denominators are zero. We evaluate contacts separately for sequence separations $12 \leq |i-j| < 24$, $24 \leq |i-j| < 50$, and $|i-j| \geq 50$, restricting both contact sets to the corresponding range. For each backbone, F1 is averaged over the eight generated sequences, and the resulting backbone-level scores are averaged across backbones.

B.1.7 STATISTICAL ANALYSIS

Paired bootstrap. We estimate uncertainty in contact F1 improvements using paired bootstrap resampling at the backbone level. For each model and sequence separation range, we compute the F1 difference between models trained with and without DISCO loss for each backbone. We perform 10,000 bootstrap resamples, sampling backbones with replacement within each length group while preserving the original group sizes and the pairing between the two training conditions. For each resample, we compute the mean F1 difference across sampled backbones. The 2.5th and 97.5th percentiles of the resulting distribution define the 95% confidence interval.

Correlation analysis. We examine whether improvements in contact recovery are associated with improvements in structural self-consistency. For each backbone b, we define

$$\Delta\text{F1}_b = \text{F1}_b^{+\mathcal{L}_{\text{DISCO}}} - \text{F1}_b^{\text{baseline}}, \qquad \Delta\text{scTM}_b = \text{scTM}_b^{+\mathcal{L}_{\text{DISCO}}} - \text{scTM}_b^{\text{baseline}},$$

where the superscripts denote models trained with and without DISCO loss, respectively. Spearman's rank correlation is computed between these backbone-level differences separately for each model and sequence separation range. We report correlations across all lengths and within each length group to examine whether the associations persist among backbones of the same length. Confidence intervals are obtained from 10,000 bootstrap resamples, jointly resampling each backbone's paired differences and recomputing the correlation. Resampling is stratified by length for the pooled analysis and performed within the corresponding group for length-specific analyses. We report percentile 95% confidence intervals.

B.2 SEQUENCE EVALUATION

B.2.1 SEQUENCE BENCHMARK DATASETS

To evaluate sequence recovery and perplexity on natural proteins, we use the CATH 4.2 test set, TS50, and TS500.

CATH 4.2 test set. We use the test split of the S40 dataset described in Appendix A.2.1. Results are reported on three subsets: Short (sequence length ≤ 100 residues), Single-chain (proteins with a single chain), and All (the full test set).

TS50. TS50 consists of 50 single-chain proteins curated from the PDB and provides an additional benchmark for sequence recovery and perplexity.

TS500. TS500 contains 500 single-chain proteins and provides a larger benchmark for the same evaluation.

B.2.2 SEQUENCE RECOVERY AND PERPLEXITY

Both sequence recovery and perplexity are computed per protein. We report their medians across proteins in each test set.

Sequence recovery. Sequence recovery measures the percentage of evaluated residues at which the designed amino acid matches the native amino acid:

$$\text{Recovery} = \frac{100}{N} \sum_{i=1}^{N} \mathbb{I}\left[\hat{s}_i = s_i^{\text{native}}\right],$$

where N is the number of evaluated residues, $\hat{s}_i$ is the designed amino acid, and s_i^{native} is the native amino acid at position i.

Perplexity. Perplexity is computed as the exponential of the mean negative log-probability assigned to the native amino acids over evaluated residues:

$$\text{Perplexity} = \exp\left(-\frac{1}{N} \sum_{i=1}^{N} \log p_i(s_i^{\text{native}})\right),$$

where $p_i(s_i^{\text{native}})$ denotes the probability assigned to the native amino acid at position i under the model's evaluation procedure. Lower values indicate higher probability assigned to the native sequence.

C Ablation Studies and Analysis

We examine the sensitivity of DISCO loss to the softmax temperature and loss weight, the effects of varying the number of crops and composition, and the contribution of pretrained ESM features. We additionally use frozen-logit probes to assess structural information accessible from inverse folding logits and explore its relation to differences in the contribution of ESM features across models.

C.1 Hyperparameter Sensitivity

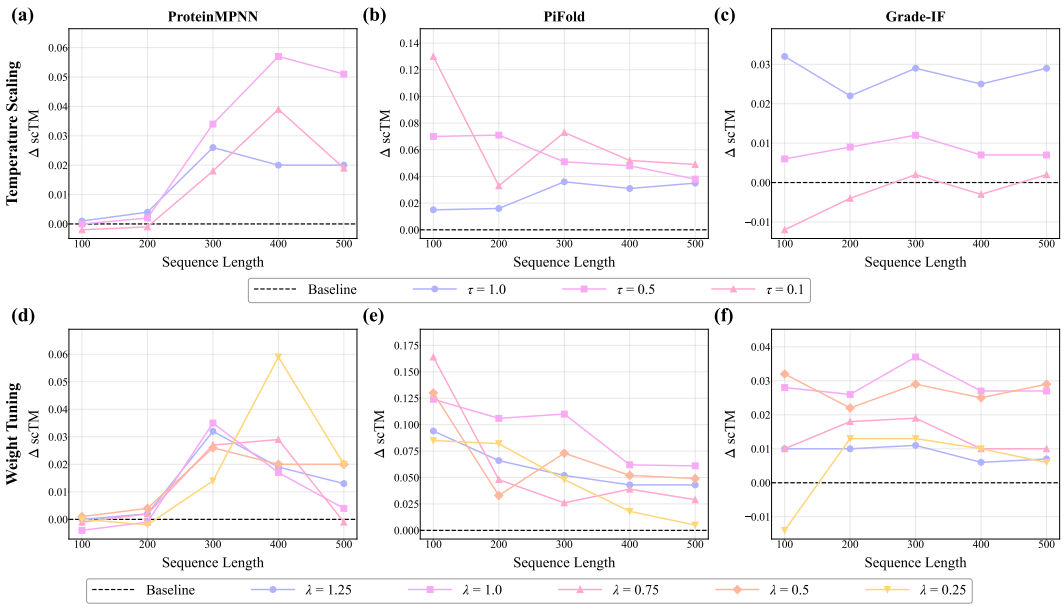

Figure A2: Hyperparameter sensitivity of $\mathcal{L}_{\text{DISCO}}$. Top row **(a–c)**: softmax temperature τ; bottom row **(d–f)**: loss weight λ. Columns correspond to ProteinMPNN, PiFold, and Grade-IF, respectively. During each sweep, the other hyperparameter is fixed: $\lambda = 0.5$ in **(a–c)**, $\tau = 1.0$ in **(d, f)**, and $\tau = 0.1$ in **(e)**. ΔscTM is measured relative to each model's baseline trained without DISCO loss (dashed line), using ESMFold across sequence lengths following the evaluation protocol in Appendix B.1.5.

We analyze the sensitivity of DISCO loss to the softmax temperature τ and loss weight λ in ProteinMPNN, PiFold, and Grade-IF. We sweep $\tau \in \{0.1, 0.5, 1.0\}$ and $\lambda \in \{0.25, 0.5, 0.75, 1.0, 1.25\}$.

During the temperature sweep, λ is fixed at 0.5 for all models. During the loss-weight sweep, τ is fixed at 1.0 for ProteinMPNN and Grade-IF and 0.1 for PiFold. Figure A2 reports ΔscTM relative to each model's baseline across sequence lengths, evaluated using ESMFold following the protocol in Section 5.2.

Temperature scaling. The effect of temperature varies across models and sequence lengths. ProteinMPNN achieves the highest scTM at $\tau = 0.5$ for $L \geq 300$, with gains of 0.057 and 0.051 at $L = 400$ and $L = 500$, respectively. However, its scTM at $L = 100$ matches the baseline, whereas $\tau = 1.0$ yields improvements across all lengths, albeit small gains for short sequences. We therefore use $\tau = 1.0$ to favor improvements across the full length range. PiFold achieves the highest scTM at $\tau = 0.1$ for all lengths except $L = 200$; at $L = 100$, ΔscTM increases from 0.015 at $\tau = 1.0$ to 0.130 at $\tau = 0.1$. Grade-IF achieves the highest scTM at $\tau = 1.0$ across all lengths, with gains of 0.022–0.032 over the baseline. Lowering the temperature to $\tau = 0.5$ reduces these gains, while $\tau = 0.1$ yields scTM values close to or below the baseline.

Loss weight. No single loss weight gives the highest scTM across all lengths for ProteinMPNN. The largest gain at $L = 400$ is 0.059 with $\lambda = 0.25$, whereas at $L = 300$ it is 0.035 with $\lambda = 1.0$. With $\lambda = 0.5$, scTM improves over the baseline at every length. PiFold achieves the highest scTM at $\lambda = 1.0$ for $L \geq 200$, with gains of 0.106 and 0.110 at $L = 200$ and $L = 300$, respectively. At $L = 100$, however, $\lambda = 0.75$ gives the largest gain of 0.164. For Grade-IF, $\lambda = 0.5$ achieves the highest scTM at $L = 100$ and $L = 500$, while $\lambda = 1.0$ performs best at $L = 200$, $L = 300$, and $L = 400$. We therefore use $\lambda = 1.0$, which performs best at three of the five lengths and remains within 0.004 of the best scTM at the remaining two lengths.

Based on these results, we use $(\tau, \lambda) = (1.0, 0.5)$ for ProteinMPNN, $(0.1, 1.0)$ for PiFold, and $(1.0, 1.0)$ for Grade-IF in our main experiments.

C.2 Effects of Crop Number and Composition

We examine how the number of crops and the inclusion of primary pair and diagonal context crops affect the performance of DISCO loss. All experiments use PiFold with $\tau = 0.1$ and $\lambda = 1.0$.

Table A2: Ablation study on PiFold. All non-baseline configurations use $\mathcal{L}_{\text{DISCO}}$ with $\tau = 0.1$ and $\lambda = 1.0$. The full configuration uses four crops, and all pair-representation crop ablations retain this number of crops. Full results are repeated in the second group for reference. Best values for each metric and length within each ablation group are in bold.

	L=100			L=200			L=300			L=400			L=500		
Configuration	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT
Baseline	0.733	4.358	74.04	0.552	11.454	51.07	0.475	14.621	46.80	0.356	19.401	39.23	0.331	22.291	37.20
Number of crops															
Crop = 1	0.793	2.327	79.10	0.610	9.361	60.65	0.548	12.766	55.20	0.398	18.401	45.93	0.388	**20.372**	42.61
Crop = 2	0.763	2.871	79.67	0.588	10.211	60.05	0.494	14.053	52.16	0.373	19.271	42.99	0.350	21.628	41.11
Crop = 3	0.869	1.678	81.16	**0.704**	**6.524**	62.01	0.578	11.428	54.80	0.413	17.845	44.35	0.370	20.560	41.39
Crop = 4	0.857	1.829	80.92	0.658	8.333	**62.73**	**0.585**	**10.966**	54.82	**0.418**	**17.242**	**46.26**	**0.392**	20.468	42.61
Crop = 5	**0.883**	**1.511**	**82.67**	0.621	9.968	61.46	0.547	12.802	**56.20**	0.389	18.288	46.20	0.355	21.538	**44.56**
Pair-representation crop ablation (4 crops)															
Full	**0.857**	**1.829**	**80.92**	**0.658**	**8.333**	**62.73**	**0.585**	**10.966**	**54.82**	**0.418**	**17.242**	**46.26**	**0.392**	**20.468**	**42.61**
w/o diagonal context crops	0.836	1.897	79.05	0.573	10.340	57.11	0.499	14.451	53.30	0.371	18.917	43.75	0.363	21.172	41.40
w/o primary pair crop	0.776	2.632	78.95	0.585	10.442	57.37	0.492	13.708	51.38	0.380	19.089	40.92	0.352	21.667	39.21

Number of crops. We compare one to five crops using PiFold with $\tau = 0.1$ and $\lambda = 1.0$ fixed throughout. As shown in Table A2, all configurations improve scTM and scRMSD over the baseline across all lengths. Five crops achieve the highest scTM and lowest scRMSD at $L = 100$, while three crops perform best on both metrics at $L = 200$. Four crops achieve the highest scTM at $L \geq 300$ and the lowest scRMSD at $L = 300$ and $L = 400$. Increasing the crop count from four to five reduces scTM at every length except $L = 100$. Based on these results, we use four crops as the default setting, considering its performance on longer backbones.

Pair-representation crop components. The primary pair crop provides features for the residue pairs to be predicted, while the diagonal context crops provide within-segment pair features for the

corresponding residue segments. Using the same hyperparameters and four crops, we compare configurations that remove either component. The full configuration achieves higher scTM and pLDDT and lower scRMSD than both removal variants across all lengths. At $L = 300$, removing the diagonal context crops reduces scTM from 0.585 to 0.499 and increases scRMSD from 10.966 to 14.451 Å. Removing the primary pair crop yields an scTM of 0.492 and an scRMSD of 13.708 Å. Both variants retain improvements in scTM and scRMSD over the baseline, but their gains are smaller than those of the full configuration. The performance decrease upon removing either component suggests that combining target-pair features with context within the corresponding segments supports more effective structural supervision and improved self-consistency.

C.3 ESM Feature Ablation

DISCO loss combines representations derived from inverse folding logits with ESM features for distogram prediction. Although ESM parameters remain frozen during training, its contextual features depend on the sequences derived from the model's current logits. We assess their contribution by excluding sequence representations, attention maps, or both, while retaining the hyperparameters selected in Appendix C.1 and all other training settings. Evaluation uses AF2 and follows the same backbone set and protocol as Section 5.2.

Table A3: ESM feature ablation evaluated with AF2 using the self-consistency protocol in Section 5.2. Values are medians by length L; scRMSD is in Å. *w/o* ESM, Seq repr., and Attn Map exclude both ESM features, sequence representations, and attention maps, respectively. **Bold**: best values within each model and length (higher scTM/pLDDT; lower scRMSD), including ties.

	$L = 100$			$L = 200$			$L = 300$			$L = 400$			$L = 500$		
Configuration	**scTM**	**scRMSD**	**pLDDT**	**scTM**	**scRMSD**	**pLDDT**	**scTM**	**scRMSD**	**pLDDT**	**scTM**	**scRMSD**	**pLDDT**	**scTM**	**scRMSD**	**pLDDT**
ProteinMPNN															
Baseline	0.953	0.866	91.32	0.964	1.119	90.16	0.936	2.110	86.03	0.736	8.556	72.14	0.575	15.783	63.39
$+\mathcal{L}_{\text{DISCO}}$	0.952	0.875	91.02	0.967	1.038	90.62	0.955	1.643	87.39	**0.839**	**5.455**	**76.10**	**0.618**	**14.480**	64.28
$+\mathcal{L}_{\text{DISCO}}$ (w/o ESM)	0.954	0.859	91.29	**0.968**	**1.027**	**91.02**	**0.958**	**1.627**	**88.38**	0.830	6.393	75.44	0.614	14.498	**64.72**
$+\mathcal{L}_{\text{DISCO}}$ (w/o Seq repr.)	0.953	0.862	91.30	0.964	1.100	90.98	0.951	1.779	88.17	0.769	7.498	73.58	0.606	15.053	64.47
$+\mathcal{L}_{\text{DISCO}}$ (w/o Attn Map)	**0.955**	**0.842**	**91.36**	0.965	1.114	90.38	0.951	1.752	87.75	0.797	7.222	74.30	0.597	15.012	64.33
PiFold															
Baseline	0.686	4.598	85.09	0.564	11.743	63.16	0.487	16.246	58.52	0.368	22.141	50.47	0.355	24.658	48.72
$+\mathcal{L}_{\text{DISCO}}$	**0.873**	**1.568**	**86.59**	**0.795**	**4.144**	**75.98**	**0.685**	**8.647**	**67.05**	**0.505**	**16.342**	**56.40**	**0.442**	**20.636**	**51.43**
$+\mathcal{L}_{\text{DISCO}}$ (w/o ESM)	0.705	4.035	80.22	0.580	11.644	62.59	0.440	18.217	55.54	0.341	25.202	51.69	0.328	28.011	50.07
$+\mathcal{L}_{\text{DISCO}}$ (w/o Seq repr.)	0.842	2.087	85.54	0.597	10.034	64.46	0.527	14.610	59.02	0.366	22.146	49.70	0.311	26.096	46.63
$+\mathcal{L}_{\text{DISCO}}$ (w/o Attn Map)	0.837	2.000	86.08	0.691	6.918	68.68	0.571	12.266	60.73	0.418	19.181	49.89	0.393	21.996	46.63
Grade-IF															
Baseline	0.323	16.041	**63.53**	0.214	22.411	42.35	0.194	26.473	**41.47**	0.184	29.615	**39.70**	0.168	33.760	**38.90**
$+\mathcal{L}_{\text{DISCO}}$	0.347	13.947	61.75	**0.251**	19.450	41.91	**0.241**	22.753	39.80	**0.225**	**25.293**	38.50	**0.217**	**27.823**	37.50
$+\mathcal{L}_{\text{DISCO}}$ (w/o ESM)	0.324	14.464	57.49	0.236	20.094	38.48	0.227	23.775	36.33	0.207	27.384	35.12	0.201	30.769	34.45
$+\mathcal{L}_{\text{DISCO}}$ (w/o Seq repr.)	**0.365**	**13.669**	62.77	0.247	**19.409**	**42.43**	0.232	**22.740**	40.62	0.222	25.755	38.91	0.214	27.958	38.13
$+\mathcal{L}_{\text{DISCO}}$ (w/o Attn Map)	0.337	13.727	60.62	0.242	19.727	40.25	0.226	23.261	38.26	0.207	26.452	36.38	0.197	28.999	35.54

Table A3 reports the complete self-consistency results, including scRMSD and pLDDT alongside scTM. For PiFold, at $L = 200$, excluding ESM increases scRMSD from 4.144 to 11.644 Å and decreases pLDDT from 75.98 to 62.59. Retaining sequence representations yields better scTM and scRMSD than retaining only attention maps at $L \geq 200$. For Grade-IF, combining both features improves scTM and scRMSD over excluding ESM across all lengths; at $L = 500$, scRMSD decreases from 30.769 to 27.823 Å. In contrast to PiFold, retaining attention maps alone yields better scTM and scRMSD than retaining sequence representations alone across all lengths. For ProteinMPNN, the benefit of ESM is most apparent on longer backbones: using both features yields the best scTM and scRMSD at $L = 400$ and $L = 500$, whereas excluding both gives the best values at $L = 200$ and $L = 300$. These results indicate that ESM improves scTM and scRMSD across lengths in PiFold and Grade-IF, while its benefits are concentrated on longer backbones in ProteinMPNN. The relative contributions of sequence representations and attention maps also differ across models. In Appendix C.4, we investigate whether these differences are related to structural information accessible from inverse folding logits using frozen-logit probes.

C.4 Probing Structural Information in Inverse Folding Logits

DISCO loss uses an auxiliary module that combines logit-derived representations with ESM features for distogram prediction. In the ESM feature ablation, ESM substantially improves self-consistency for PiFold and provides additional gains for Grade-IF, whereas ProteinMPNN retains substantial improvements without ESM. Motivated by this observation, we investigate whether ProteinMPNN's logits already provide useful structural cues for distance prediction. We freeze the three baseline models trained without DISCO loss and separately train distogram probes using their logits without ESM features under a common training protocol.

C.4.1 Experimental Setup

Baseline models and probes. We trained ProteinMPNN, PiFold, and Grade-IF from scratch without DISCO loss, following each model's original settings and objective. We froze the resulting baselines and kept them in evaluation mode while training randomly initialized DISCO auxiliary modules without ESM features, using only DISCO loss. For ProteinMPNN, all input residues were replaced with unknown tokens during both probe training and evaluation to prevent access to native sequence information through teacher forcing.

Probe training. All probes were trained on the CATH 4.2 training split for 50 epochs, selecting the checkpoint with the lowest validation DISCO loss. We used Adam with OneCycleLR, a maximum learning rate of 10^{-3}, zero weight decay, temperature $\tau = 1.0$, and a DISCO loss weight of 1.0 across all probes. We repeated probe training with three random seeds and report means and standard deviations.

Positional control. We trained the same probe architecture with all-zero logits, retaining only positional information.

C.4.2 Evaluation Protocol

Evaluation datasets. We evaluated the trained probes on 608 structures from the CATH 4.2 validation split and 500 RFdiffusion-generated backbones, with 100 backbones at each length $L \in \{100, 200, 300, 400, 500\}$. We compared probe performance across models on the CATH validation split using the same checkpoint selection procedure. RFdiffusion backbones, used for neither probe training nor checkpoint selection, provided a separate evaluation of generalization to de novo structures.

Distogram reconstruction and contact definition. Deterministic tiling covered all residue pairs. Predicted probabilities in overlapping regions were averaged and symmetrized to construct the full distogram. Contact scores were computed by summing probabilities over the 19 distance bins with centers below 8 Å. Ground-truth contacts were defined by distances below 8 Å, using Cβ coordinates for CATH and Cα for glycine. For RFdiffusion backbones, which have placeholder poly-G sequences, virtual Cβ coordinates were constructed from backbone atoms for all residues.

Contact precision. Our primary metric was long-range contact precision at top-$L/5$, following Rao et al. (2021), with long-range defined as $|i-j| \geq 24$ for this probe analysis rather than $|i-j| \geq 12$ as in Appendix B.1.6. We ranked valid residue pairs satisfying $i < j$ and $|i-j| \geq 24$ by their contact scores and selected $\lfloor L/5 \rfloor$ pairs, bounded below by one and above by the number of candidates, where L is the number of valid residues. Precision was calculated as the fraction of selected pairs that were true contacts. We also evaluated long-range precision at top-$L/2$ and top-L, and short- and medium-range precision at top-$L/5$ using $6 \leq |i-j| < 12$ and $12 \leq |i-j| < 24$, respectively. All metrics followed the same selection procedure, with precision computed per chain and averaged across chains.

Distance prediction error. We additionally reported expected-distance mean absolute error (MAE). Predicted distances were calculated as $\hat{d}_{ij} = \sum_b p_{ij,b} c_b$, where the bin centers c_b range from approximately 2.16 to 21.84 Å. MAE was evaluated only for valid pairs with ground-truth dis-

Table A4: Frozen-logit probe results on RFdiffusion-generated *de novo* backbones ($n = 500$) and natural protein structures from CATH 4.2 ($n = 608$). Values are means ± standard deviations over three seeds. Contact precision is reported in %, and expected-distance MAE in Å. The positional control uses all-zero logits. **Bold** indicates the best value in each row (higher precision; lower MAE).

Metric	ProteinMPNN	PiFold	Grade-IF	Positional control
De Novo Backbones				
Long-range top-$L/5$ precision	$\mathbf{58.9 \pm 3.7}$	48.9 ± 1.2	38.2 ± 2.2	5.5 ± 1.3
Long-range top-$L/2$ precision	$\mathbf{47.7 \pm 3.5}$	38.2 ± 0.8	29.7 ± 1.0	5.8 ± 0.7
Long-range top-L precision	$\mathbf{37.6 \pm 2.9}$	29.6 ± 0.7	22.5 ± 0.5	6.0 ± 0.3
Medium-range top-$L/5$ precision	$\mathbf{64.3 \pm 1.6}$	48.8 ± 0.1	26.6 ± 1.5	3.9 ± 0.1
Short-range top-$L/5$ precision	$\mathbf{68.7 \pm 0.9}$	55.3 ± 0.4	33.6 ± 0.9	5.0 ± 0.2
Expected-distance MAE	$\mathbf{2.7 \pm 0.1}$	3.2 ± 0.1	3.2 ± 0.0	3.7 ± 0.1
Natural Protein Structures				
Long-range top-$L/5$ precision	$\mathbf{32.7 \pm 0.2}$	22.6 ± 0.6	21.9 ± 0.2	2.8 ± 0.2
Long-range top-$L/2$ precision	$\mathbf{25.6 \pm 0.2}$	18.3 ± 0.5	17.4 ± 0.1	2.7 ± 0.2
Long-range top-L precision	$\mathbf{20.0 \pm 0.1}$	14.9 ± 0.3	14.0 ± 0.1	2.7 ± 0.1
Medium-range top-$L/5$ precision	$\mathbf{46.9 \pm 0.4}$	34.6 ± 1.1	34.0 ± 0.2	4.1 ± 0.3
Short-range top-$L/5$ precision	$\mathbf{60.3 \pm 0.4}$	45.6 ± 0.8	43.8 ± 0.5	7.3 ± 0.3
Expected-distance MAE	$\mathbf{3.0 \pm 0.0}$	3.4 ± 0.0	3.4 ± 0.0	3.7 ± 0.1

tances $2 < d_{ij} < 22$ Å, without a sequence-separation restriction. Unlike contact precision, MAE was computed by pooling absolute errors over all eligible residue pairs across chains.

C.4.3 Probe Results

Table A4 reports contact precision and distance prediction errors for probes trained on each model's frozen logits, evaluated on de novo backbones and natural protein structures. We examine whether model differences persist across sequence-separation ranges and selection cutoffs, and compare against a positional control.

De Novo Backbones. On RFdiffusion backbones, long-range top-$L/5$ precision is 58.9% for ProteinMPNN, 48.9% for PiFold, and 38.2% for Grade-IF, all exceeding the positional control at 5.5%. The same model ranking holds at top-$L/2$ and top-L, indicating consistent differences across selection cutoffs. This ranking also holds for short- and medium-range contacts, with medium-range top-$L/5$ precision reaching 64.3%, 48.8%, and 26.6%, respectively. ProteinMPNN achieves the lowest expected-distance MAE of 2.7 Å, compared with 3.2 Å for both PiFold and Grade-IF and 3.7 Å for the positional control. These results show that ProteinMPNN logits support more accurate contact and distance prediction under the same probe training settings.

Natural Protein Structures. On CATH structures, long-range top-$L/5$ precision is 32.7% for ProteinMPNN, 22.6% for PiFold, and 21.9% for Grade-IF, all exceeding the positional control at 2.8%. The ProteinMPNN–PiFold gap is 10.1 percentage points, similar to the 10.0-point gap on RFdiffusion backbones. In contrast, PiFold and Grade-IF differ by only 0.7 points, indicating that their separation depends on the evaluation dataset. ProteinMPNN achieves the highest precision across the remaining contact metrics and the lowest expected-distance MAE of 3.0 Å, compared with 3.4 Å for both PiFold and Grade-IF. These consistent findings across both datasets indicate that ProteinMPNN logits provide useful structural cues for distogram prediction, consistent with the substantial self-consistency gains retained without ESM features.

D Additional Structural Evaluation

This section extends the structural evaluations in the main text with long-range contact recovery on ESMFold-predicted structures, correlations between contact recovery and self-consistency improvements, and secondary structure recovery. Evaluation protocols and contact definitions are provided

in Appendix B. Unless otherwise specified, analyses use the AF2-predicted structures from the self-consistency evaluation in Section 5.2.

D.1 Long-Range Contact Analysis

D.1.1 Long-Range Contact Recovery on ESMFold Predictions

To examine whether the contact recovery improvements observed with AF2 also hold with ESMFold, we evaluate the same generated sequences using ESMFold-predicted structures. Contact definitions and aggregation procedures follow Appendix B.1.6.

Table A5: Long-range contact F1 on ESMFold-predicted structures by sequence separation $|i-j|$. Scores are averaged over eight sequences per backbone, then across backbones. ΔF1 denotes the change from baseline with DISCO loss. Percentile 95% CIs use 10,000 paired backbone bootstrap resamples, stratified by length ($n_{\text{bb}} = 500$; $n_{\text{bb}} = 485$ for $|i-j| \geq 50$).

Model	$\lvert i-j \rvert$	Baseline	$+\mathcal{L}_{\text{DISCO}}$	ΔF1 [95% CI]
ProteinMPNN	12–24	0.564	**0.571**	$+\mathbf{0.007}$ [+0.002, +0.012]
	24–50	0.560	**0.568**	$+\mathbf{0.008}$ [+0.001, +0.014]
	≥ 50	0.423	**0.436**	$+\mathbf{0.013}$ [+0.005, +0.022]
PiFold	12–24	0.422	**0.472**	$+\mathbf{0.050}$ [+0.040, +0.060]
	24–50	0.375	**0.436**	$+\mathbf{0.061}$ [+0.049, +0.074]
	≥ 50	0.185	**0.269**	$+\mathbf{0.084}$ [+0.070, +0.098]
Grade-IF	12–24	0.094	**0.138**	$+\mathbf{0.044}$ [+0.039, +0.049]
	24–50	0.056	**0.107**	$+\mathbf{0.051}$ [+0.045, +0.056]
	≥ 50	0.016	**0.026**	$+\mathbf{0.010}$ [+0.007, +0.013]

Table A5 shows that DISCO loss improves contact F1 across all separation ranges for all three models, with all 95% confidence intervals excluding zero. As with AF2, gains for ProteinMPNN and PiFold increase with sequence separation, reaching 0.013 and 0.084 at $|i-j| \geq 50$, respectively (AF2: 0.023 and 0.127). Grade-IF shows larger gains in the two shorter separation ranges, with improvements of 0.044, 0.051, and 0.010 across the three ranges. These results show improvements in contact recovery across all three inverse folding models under both AF2 and ESMFold evaluation.

D.1.2 Correlation with Self-consistency

Table A6: Contact recovery gains and their association with self-consistency gains on AF2-predicted structures. ΔF1 is the mean backbone-level change with DISCO loss, and ρ is the Spearman correlation between backbone-level ΔF1 and ΔscTM. Brackets indicate percentile 95% CIs for ρ from 10,000 backbone bootstrap resamples stratified by length ($n_{\text{bb}} = 500$; $n_{\text{bb}} = 485$ for $|i-j| \geq 50$).

$\lvert i-j \rvert$	ProteinMPNN		PiFold		Grade-IF	
	ΔF1	ρ [95% CI]	ΔF1	ρ [95% CI]	ΔF1	ρ [95% CI]
12–24	+0.012	0.543 [0.469, 0.610]	+0.083	0.593 [0.528, 0.653]	+0.048	0.447 [0.370, 0.518]
24–50	+0.013	0.647 [0.581, 0.707]	+0.099	0.713 [0.660, 0.758]	+0.051	0.667 [0.606, 0.720]
≥ 50	+0.023	0.761 [0.710, 0.807]	+0.127	0.760 [0.715, 0.799]	+0.013	0.407 [0.324, 0.485]

We quantify the association between contact recovery gains and overall structural self-consistency gains with DISCO loss using Spearman correlations between backbone-level ΔF1 and ΔscTM. Table A6 reports mean F1 gains and the corresponding correlations across sequence separation ranges. All three models show positive mean F1 gains and positive correlations across all separation ranges, with all correlation confidence intervals excluding zero. For ProteinMPNN and PiFold, both mean F1 gains and correlations increase with sequence separation. At $|i-j| \geq 50$, their mean F1 gains

reach 0.023 and 0.127, with correlations of $\rho = 0.761$ and 0.760, respectively. Grade-IF shows its largest mean F1 gain and strongest correlation at $24 \leq |i - j| < 50$ (ΔF1 $= 0.051$, $\rho = 0.667$), compared with 0.013 and 0.407 at $|i - j| \geq 50$. Together, these results show that contact recovery improves on average and that backbones with larger contact recovery gains tend to exhibit larger scTM gains.

D.2 Secondary Structure Evaluation

We compute the fraction and recall of helix, sheet, and loop on the AlphaFold2-predicted structures used in the self-consistency evaluation. This allows us to assess how accurately the sequences generated by inverse folding models reproduce the secondary structure of the original backbones.

D.2.1 DSSP Protocol

Secondary structure analysis is performed using the DSSP algorithm (Kabsch & Sander, 1983). DSSP classifies each residue into one of eight states based on backbone hydrogen-bonding patterns, which are collapsed into three states: helix (H, G, I), sheet (E, B), and loop (T, S, C). DSSP is applied to both the original RFdiffusion backbones and the predicted structures from the generated sequences.

D.2.2 Metrics

Metrics are averaged over eight sequences per backbone and then summarized by the median within each length group, excluding undefined recall values for classes absent from the target backbone.

SS Fraction. For each structure, we compute the fraction of residues assigned to helix, sheet, and loop:

$$F_c = \frac{|\{i : \mathrm{DSSP}(i) = c\}|}{N}, \quad c \in \{\mathrm{helix}, \mathrm{sheet}, \mathrm{loop}\},$$

where N is the total number of residues and $F_{\text{helix}} + F_{\text{sheet}} + F_{\text{loop}} = 1$. Fractions are computed for both the original backbone and the predicted structure, and their distributions are compared to assess preservation of secondary structure composition.

SS Recall. Using the DSSP states of the original backbone as ground truth, we compute the per-class recall:

$$\mathrm{Recall}_c = \frac{|\{i : \hat{s}_i = c,\ s_i = c\}|}{|\{i : s_i = c\}|}.$$

Here, s_i and $\hat{s}_i$ denote the DSSP assignments for residue i in the original backbone and the predicted structure, respectively. Higher values indicate better recovery of the original secondary structure at the residue level.

D.2.3 Secondary Structure Fraction and Recall

We analyze how structural self-consistency improvements manifest at the secondary-structure level by comparing the fractions and per-residue recall of α-helices, β-sheets, and loops between target backbones and AF2-predicted structures (Figure A3). This comparison characterizes preservation across secondary-structure types. In particular, β-sheets can bring residues that are distant in sequence into spatial proximity, making their recovery informative for understanding the effects of pairwise distance supervision from DISCO loss.

The PiFold and Grade-IF baselines show lower β-sheet fractions and higher loop fractions than the target backbones at $L \geq 200$. With DISCO loss, both fractions move closer to the target values, while β-sheet recall increases across all lengths. At $L = 400$, β-sheet recall increases from 0.588 to 0.774 for PiFold and from 0.121 to 0.257 for Grade-IF. ProteinMPNN shows little change in β-sheet fraction, while recall improves at $L \geq 200$ but decreases at $L = 100$. α-Helix recall remains largely unchanged for ProteinMPNN and PiFold and increases across all lengths for Grade-IF, reaching 0.779 at $L = 500$. Loop recall also improves across all lengths for all three models.

These results show that the structural improvements with DISCO loss extend to secondary-structure composition and residue-level recovery. In particular, improved β-sheet recovery, together with

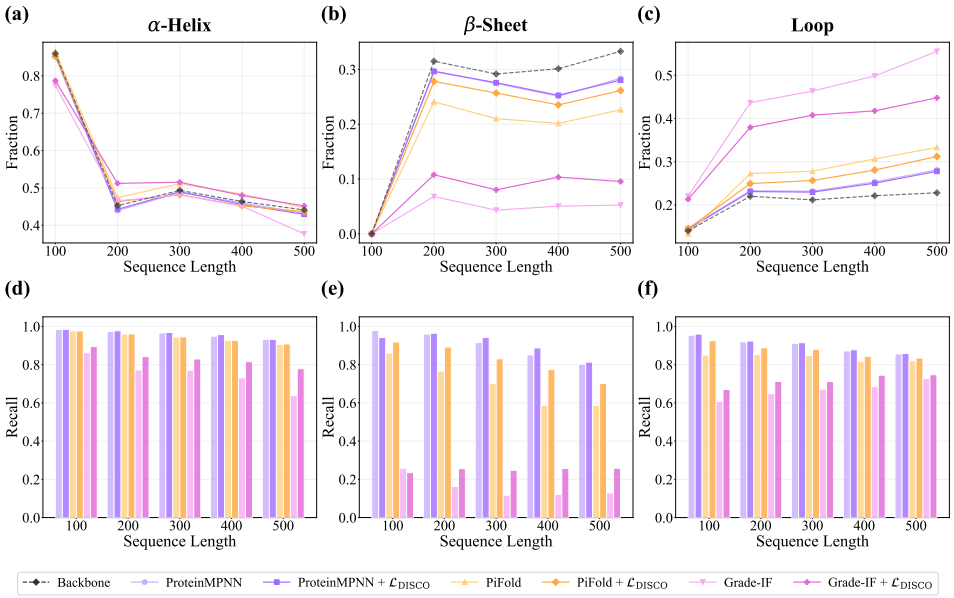

Figure A3: Secondary structure fraction and recall on AF2-predicted structures for ProteinMPNN, PiFold, and Grade-IF, with and without DISCO loss. Backbone lengths are 100, 200, 300, 400, and 500 residues. (a)–(c): fractions of α-helix, β-sheet, and loop residues, respectively. Dashed black lines indicate target backbone fractions. (d)–(f): corresponding per-class recall against the target backbone.

the long-range contact F1 results in Section 5.3, supports the interpretation that DISCO loss helps recover target structural relationships between residues that are distant in sequence.

E Sequence Performance and Diversity

This section reports detailed sequence recovery and perplexity results on the CATH 4.2 test subsets and analyzes sequence diversity with and without DISCO loss. The CATH 4.2 evaluation dataset and sequence performance metrics are described in Appendix B.2.

E.1 CATH 4.2 Benchmark

Table A7: Median sequence recovery (%) and perplexity across the Short, Single-chain, and All subsets of the CATH 4.2 test set. Bold values indicate the best result in each column.

	CATH 4.2					
	Perplexity (↓)			Recovery % (↑)		
Model	**Short**	**Single-chain**	**All**	**Short**	**Single-chain**	**All**
GCA	7.09	7.49	6.05	32.62	31.10	37.64
GVP	7.23	7.84	5.36	30.60	28.95	39.47
AlphaDesign	7.32	7.63	6.30	34.16	32.66	41.31
ProteinMPNN	9.81	9.42	5.41	27.13	26.45	45.97
PiFold	6.04	6.31	4.55	39.84	38.53	51.66
Grade-IF	**5.18**	6.16	**4.35**	**46.33**	**41.67**	**52.05**
ProteinMPNN $+\mathcal{L}_{\text{DISCO}}$	9.55	9.44	5.42	26.99	27.12	42.09
PiFold $+\mathcal{L}_{\text{DISCO}}$	7.09	7.13	4.48	37.25	38.71	51.91
Grade-IF $+\mathcal{L}_{\text{DISCO}}$	5.20	**6.13**	4.41	45.41	**41.67**	51.76

Table A7 provides detailed results on the Short, Single-chain, and All subsets of CATH 4.2, complementing the overall benchmark results in Table 3. The effects of DISCO loss vary across models and subsets. On the All subset, PiFold improves in both metrics, with recovery increasing from 51.66% to 51.91% and perplexity decreasing from 4.55 to 4.48. ProteinMPNN shows a recovery decrease of 3.88 percentage points with little change in perplexity, while Grade-IF shows a small recovery decrease from 52.05% to 51.76% and a perplexity increase from 4.35 to 4.41. On the Single-chain subset, recovery increases for ProteinMPNN and PiFold but remains unchanged at 41.67% for Grade-IF, whose perplexity improves from 6.16 to 6.13. On the Short subset, recovery decreases for all three models, with the largest decrease in PiFold (39.84% to 37.25%). These results highlight variation in native sequence recovery across subsets; as discussed in Section 5.4, this metric measures agreement with the native sequence and does not directly measure the structural quality of generated sequences.

E.2 Sequence Diversity

We evaluate sequence diversity alongside structural self-consistency using the eight sequences generated per backbone, with identical sampling settings for models trained with and without DISCO loss.

E.2.1 Diversity metric.

We measure sequence diversity as the average pairwise mismatch ratio. Sequences generated for the same backbone share residue positions and length, so no alignment is required. For two sequences s_i and s_j of length L, the mismatch ratio is

$$D_{i,j} = \frac{1}{L} \sum_{l=1}^{L} \mathbb{I}[s_{i,l} \neq s_{j,l}],$$

where $s_{i,l}$ denotes the amino acid at position l in sequence i. Diversity for each backbone is then averaged over all unordered sequence pairs:

$$\text{Div} = \frac{2}{m(m-1)} \sum_{1 \leq i < j \leq m} D_{i,j}, \qquad m = 8.$$

Higher values indicate greater sequence diversity.

Table A8: scTM and sequence diversity with and without DISCO loss on the 500 RFdiffusion-generated backbones used in the self-consistency evaluation. Diversity is the mean pairwise mismatch ratio among the eight sequences designed for each backbone, reported as mean ± standard deviation across backbones. ΔDiversity is the mean paired per-backbone change with DISCO loss relative to the baseline.

Model	scTM	Diversity	ΔDiversity
ProteinMPNN	0.787	0.4048 ± 0.0365	–
ProteinMPNN $+\mathcal{L}_{\text{DISCO}}$	**0.820**	0.3852 ± 0.0303	-0.0196
PiFold	0.512	0.1612 ± 0.0234	–
PiFold $+\mathcal{L}_{\text{DISCO}}$	**0.657**	0.1687 ± 0.0234	$+0.0074$
Grade-IF	0.296	0.4507 ± 0.0593	–
Grade-IF $+\mathcal{L}_{\text{DISCO}}$	**0.340**	0.5602 ± 0.0542	$+0.1095$

E.2.2 Results.

Table A8 reports scTM and sequence diversity for all three models. With DISCO loss, scTM increases from 0.787 to 0.820 for ProteinMPNN, from 0.512 to 0.657 for PiFold, and from 0.296 to 0.340 for Grade-IF. Diversity decreases from 0.4048 to 0.3852 for ProteinMPNN, but increases from 0.1612 to 0.1687 for PiFold and from 0.4507 to 0.5602 for Grade-IF. Thus, the diversity response varies across models, and improvements in structural self-consistency do not necessarily entail reduced sequence diversity.

F Comparison with Alternative Structural Supervision

This section compares DISCO loss with two alternative forms of structural supervision. We first describe a naïve approach that we implement to use the discrepancy between a protein folding model's predicted structure and the target backbone as a training loss (Appendix F.1). We also present the settings used to apply AFDistill, an existing method that provides structural regularization through a pretrained surrogate model (Appendix F.2). We then compare the structural self-consistency and computational costs of all three approaches (Appendix F.3).

F.1 Naïve Approach to Structural Supervision

An intuitive approach to structural supervision is to incorporate a protein folding model used for self-consistency evaluation into training. We implement this naïve approach by predicting structures from inverse folding outputs and using FAPE against the target backbone as an auxiliary loss. The folding model remains frozen, while structural gradients propagate to the inverse folding model to encourage structural consistency alongside native sequence recovery.

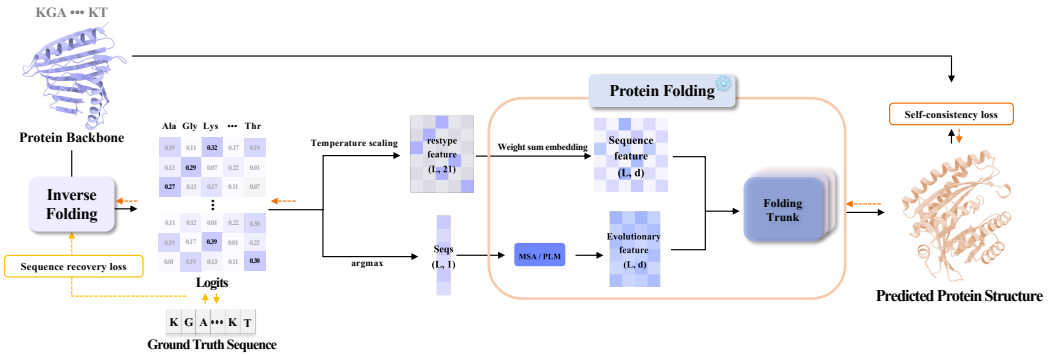

Figure A4: **Overview of the naïve approach.** The inverse folding model produces logits, which are converted into a discrete sequence via argmax (for the MSA/PLM input) and a soft probability distribution via temperature scaling (for gradient flow). Both are fed into a frozen folding trunk to predict the structure, which is supervised by the self-consistency loss against the input backbone.

F.1.1 Method

The naïve approach couples the inverse folding model with a frozen protein folding model and uses frame-aligned point error (FAPE) between the predicted structure and target backbone as auxiliary supervision. However, converting inverse folding logits into discrete residue identities is non-differentiable, blocking gradients through the discrete sequence pathway.

We therefore construct two inputs from the logits z: a discrete sequence $\hat{s}$ obtained by argmax and a continuous amino acid distribution p obtained by temperature-scaled softmax. The discrete sequence supplies sequence-dependent features, while p replaces one-hot residue inputs in the embedding operation through a weighted sum of amino acid embeddings. Gradients from the FAPE loss propagate through this continuous pathway to the inverse folding model.

Logit Processing Given an input backbone X of length L, the inverse folding model produces logits $z \in \mathbb{R}^{L\times 21}$ over the residue alphabet A. We construct a discrete sequence using argmax:

$$\hat{s}_i = \arg\max_{a\in A} z_{i,a}.$$

The discrete sequence is processed by the frozen ESM-1b model to obtain sequence features for OpenFold SoloSeq.

We also apply temperature-scaled softmax to obtain a continuous amino acid distribution:

$$p_{i,a} = \frac{\exp(z_{i,a}/\tau)}{\sum_{b\in A}\exp(z_{i,b}/\tau)},$$

where τ controls the sharpness of the distribution. This distribution replaces one-hot residue inputs in the weighted-sum embedding described below, providing a differentiable pathway from the FAPE loss to the inverse folding logits. The argmax pathway remains non-differentiable.

Weighted-Sum Embedding The residue embedding operation maps a one-hot amino acid vector $x_i \in \{0,1\}^{21}$ to an embedding:

$$e_i = x_i W,$$

where $W \in \mathbb{R}^{21\times d}$ is the pretrained residue embedding matrix. We replace x_i with the continuous distribution p_i defined in Appendix F.1.1:

$$\tilde{e}_i = p_i W = \sum_{a\in A} p_{i,a} W_a,$$

where W_a is the embedding vector for residue type a. This produces a probability-weighted sum of amino acid embeddings while retaining the pretrained embedding weights.

The resulting embeddings are combined with ESM-1b features from the argmax sequence and passed through the folding model to predict 3D coordinates. Because p_i is differentiable with respect to the inverse folding logits, the FAPE loss backpropagates through the folding model and weighted-sum embeddings to update the inverse folding model. The folding model and ESM-1b parameters remain frozen.

FAPE Loss We use frame-aligned point error (FAPE) to measure the discrepancy between the predicted structure $\hat{X}$ and the target backbone X. FAPE compares corresponding atom positions expressed in their respective residue-local coordinate frames, making the loss invariant to global rotation and translation without requiring structural alignment.

We compute the loss over the backbone atoms (N, C_α, C, O):

$$\mathcal{L}_{\text{FAPE}} = \frac{\sum_{i=1}^{L}\sum_{j=1}^{4L} m_i^{\text{frame}} m_j^{\text{atom}} \min\left(\left\|\hat{T}_i^{-1}\hat{\mathbf{x}}_j - T_i^{-1}\mathbf{x}_j\right\|_2, d_{\text{clamp}}\right)}{d_{\text{scale}} \sum_{i=1}^{L}\sum_{j=1}^{4L} m_i^{\text{frame}} m_j^{\text{atom}}}.$$

Here, $\hat{T}_i$ and T_i denote the local frames of residue i in the predicted and target structures, respectively, each constructed from the corresponding (N, C_α, C) atoms. The coordinates $\hat{\mathbf{x}}_j$ and $\mathbf{x}_j$ refer to corresponding backbone atoms in the two structures. The binary masks m_i^{frame} and m_j^{atom} indicate valid residue frames and atom positions, respectively.

Errors are clipped at $d_{\text{clamp}} = 10\,\text{Å}$ to limit the influence of large deviations, averaged over valid frame–atom pairs, and normalized by $d_{\text{scale}} = 10\,\text{Å}$. The resulting FAPE loss provides auxiliary structural supervision for the inverse folding model.

Training Objective We combine the sequence cross-entropy loss with the FAPE loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seq}} + \omega_{\text{FAPE}}\mathcal{L}_{\text{FAPE}},$$

where ω_{FAPE} controls the contribution of structural supervision. The sequence loss $\mathcal{L}_{\text{seq}}$ encourages recovery of the native sequence given the input backbone, while $\mathcal{L}_{\text{FAPE}}$ penalizes deviations of the predicted structure from the target backbone. During training, only the inverse folding model is updated; the protein folding model remains frozen.

F.1.2 Implementation and Training Details

We implement the naïve approach using ProteinMPNN and PiFold, following their respective original training settings. For structure prediction, we use OpenFold in single-sequence mode (SoloSeq), which uses ESM-1b (Rives et al., 2021) embeddings instead of MSAs.

Due to GPU memory constraints, we randomly select one sample with sequence length $L \leq 500$ from each batch for structure prediction and compute the FAPE loss only for that sample. Both OpenFold and ESM-1b remain frozen, while gradients from the FAPE loss propagate through OpenFold and the weighted-sum embedding pathway to the inverse folding model. We set the softmax temperature to $\tau = 0.1$ and the FAPE loss weight to $\omega_{\text{FAPE}} = 0.5$.

On a single NVIDIA RTX PRO 6000 GPU, training takes approximately 112 hours for ProteinMPNN over 150 epochs and 145 hours for PiFold over 100 epochs. The corresponding training times with DISCO loss are 54.6 and 16.5 hours, respectively. Its additional computation includes structure prediction with OpenFold and backpropagation through the folding model, even though its parameters remain frozen.

F.2 Structural Supervision with AFDistill

AFDistill provides structural regularization through a surrogate model pretrained to estimate AlphaFold confidence metrics, pTM and pLDDT, from amino acid sequences alone using knowledge distillation. These estimates serve as auxiliary training signals for inverse folding models, providing structural supervision without full 3D structure prediction at each training step. The surrogate is used only during training, leaving the inverse folding model's inference procedure unchanged.

F.2.1 Implementation and Training Details

To apply AFDistill to ProteinMPNN and PiFold, we follow the data preprocessing and surrogate pretraining procedures provided in the official implementation. We pass inverse folding logits at valid residue positions to the trained surrogate, using the residue mask to exclude invalid positions and mapping each model's amino acid vocabulary to the AFDistill vocabulary. We add the resulting regularization term to the original sequence objective and train both inverse folding models from scratch under their respective baseline training settings.

In an environment equipped with a single NVIDIA RTX PRO 6000 GPU, data preprocessing takes 13.5 hours and surrogate pretraining takes 24 hours. Subsequent inverse folding training takes 35.9 hours for ProteinMPNN over 150 epochs and 15.95 hours for PiFold over 100 epochs. Including preprocessing and surrogate pretraining, the total time is 73.4 hours for ProteinMPNN and 53.45 hours for PiFold.

F.3 Comparison of Performance and Computational Cost

We compare DISCO loss, AFDistill, and the FAPE-based naïve approach on ProteinMPNN and PiFold, first evaluating structural self-consistency and then computational cost, including preprocessing and surrogate pre-training where applicable.

F.3.1 Self-Consistency

Table A9: Structural self-consistency on AF2-predicted structures, following the evaluation protocol in Section 5.2. All training variants of each inverse folding model use the same number of epochs. Bold indicates the best value for each metric and length within each inverse folding model.

	$L = 100$			$L = 200$			$L = 300$			$L = 400$			$L = 500$		
Model	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT	scTM	scRMSD	pLDDT
ProteinMPNN	0.953	0.866	91.32	0.964	1.119	90.16	0.936	2.110	86.03	0.736	8.556	72.14	0.575	15.783	63.39
ProteinMPNN + AFDistill	0.952	**0.860**	91.30	0.962	1.144	**90.90**	0.950	1.807	**87.80**	0.830	6.167	**76.35**	0.609	14.834	**65.25**
ProteinMPNN $+\mathcal{L}_{\text{FAPE}}$	**0.955**	0.868	**91.57**	**0.967**	1.066	90.55	0.952	1.667	87.65	0.817	6.599	74.25	0.609	14.674	63.63
ProteinMPNN $+\mathcal{L}_{\text{DISCO}}$	0.952	0.875	91.02	**0.967**	**1.038**	90.62	**0.955**	**1.643**	87.39	**0.839**	**5.455**	76.10	**0.618**	**14.480**	64.28
PiFold	0.686	4.598	85.09	0.564	11.743	63.16	0.487	16.246	58.52	0.368	22.141	50.47	0.355	24.658	48.72
PiFold + AFDistill	0.826	2.100	85.42	0.756	5.399	69.45	0.655	10.416	64.29	0.470	18.584	52.35	0.411	21.536	47.05
PiFold $+\mathcal{L}_{\text{FAPE}}$	0.794	2.285	82.18	0.614	10.765	63.55	0.527	15.061	57.24	0.395	20.489	49.75	0.352	24.469	47.47
PiFold $+\mathcal{L}_{\text{DISCO}}$	**0.873**	**1.568**	**86.59**	**0.795**	**4.144**	**75.98**	**0.685**	**8.647**	**67.05**	**0.505**	**16.342**	**56.40**	**0.442**	**20.636**	**51.43**

Table A9 compares DISCO loss with AFDistill and the FAPE-based naïve approach. For PiFold, DISCO loss achieves the highest scTM and pLDDT and the lowest scRMSD at every length. At length 300, its scTM is 0.685, compared with 0.655 for AFDistill and 0.527 for FAPE, while its scRMSD is 8.647, compared with 10.416 and 15.061, respectively. Both DISCO loss and AFDistill improve scTM and scRMSD over the baseline across all lengths.

For ProteinMPNN, differences are small for short backbones: FAPE achieves the highest scTM and pLDDT at length 100, while AFDistill achieves the lowest scRMSD. At lengths of 200 or greater, DISCO loss achieves the lowest scRMSD and the highest or tied-highest scTM. At length 400, its

scTM is 0.839, compared with 0.830 for AFDistill and 0.817 for FAPE. However, AFDistill achieves the highest pLDDT at lengths of 200 or greater, consistent with its use of a surrogate pretrained to approximate folding-model scores, including pLDDT. This illustrates that prediction confidence and agreement with the target backbone do not always rank methods identically. These comparisons demonstrate that DISCO loss effectively improves structural self-consistency through supervision based on target-backbone residue-pair distances, without full 3D reconstruction or separate surrogate pretraining.

F.3.2 Computational Cost

Table A10: Computational costs (hours) on a single NVIDIA RTX PRO 6000 GPU for ProteinMPNN (150 epochs) and PiFold (100 epochs). Total time includes inverse folding training and, for AFDistill, one-time preprocessing and surrogate pre-training. Original pre-training costs of reused protein folding and language models are excluded. Bold indicates the lowest total time for each model.

Model	Training Time	Preprocessing	Pre-training	Total Time
ProteinMPNN $+\mathcal{L}_{\text{DISCO}}$	54.6 hr	–	–	**54.6 hr**
ProteinMPNN $+\mathcal{L}_{\text{FAPE}}$	112 hr	–	–	112 hr
ProteinMPNN + AFDistill	35.9 hr	13.5 hr	24 hr	73.4 hr
PiFold $+\mathcal{L}_{\text{DISCO}}$	16.5 hr	–	–	**16.5 hr**
PiFold $+\mathcal{L}_{\text{FAPE}}$	145 hr	–	–	145 hr
PiFold + AFDistill	15.95 hr	13.5 hr	24 hr	53.45 hr

Training with DISCO loss takes 54.6 hours for ProteinMPNN and 16.5 hours for PiFold (Table A10). The naïve approach takes 112 and 145 hours, approximately $2.1\times$ and $8.8\times$ as long, due to structure prediction and backpropagation through the folding model.

AFDistill requires less inverse folding training time (35.9 and 15.95 hours), but preprocessing and surrogate pre-training add 37.5 hours. Its total times are therefore 73.4 and 53.45 hours, approximately $1.34\times$ and $3.24\times$ those of DISCO loss.

Including initial preparation, DISCO loss achieves the lowest total time while providing strong self-consistency (Table A9): it performs best across all lengths and metrics for PiFold and achieves the best or tied-best scTM and scRMSD for ProteinMPNN at lengths of 200 or greater.

G Amino Acid Composition and Energy Analyses

The preceding self-consistency evaluation compared structures predicted from the final discrete sequences with the target backbones, examining whether structural supervision applied to continuous logits translates into improvements in the generated sequences. Here, we evaluate amino acid composition and Rosetta energy to characterize the physicochemical changes accompanying these structural improvements. This analysis examines whether improved agreement with the target backbone is accompanied by compositional biases or unfavorable changes in energy, providing an additional assessment of the physicochemical plausibility of the generated sequences.

G.1 Amino Acid Composition Analysis

We examine how DISCO loss affects amino acid composition in sequences generated by ProteinMPNN, PiFold, and Grade-IF. Using AF2-predicted structures, we classify each residue into core, boundary, or surface regions based on relative solvent-accessible surface area (rSASA). Amino acids are grouped into hydrophobic (A, V, L, I, M, F, W, Y, C), hydrophilic (R, K, D, E, N, Q, H, S, T), and neutral (G, P) categories.

We compare whole-sequence and region-specific composition between models trained with and without DISCO loss, and examine individual amino acid profiles to identify the residues contributing to these changes (Appendix G.1.1).

G.1.1 Region-Based Amino Acid Composition

Soluble proteins generally exhibit hydrophobic enrichment in the core and hydrophilic enrichment on the surface. We examine whether sequences generated with and without DISCO loss exhibit these compositional patterns across core, boundary, and surface regions in their AF2-predicted structures.

Region Definitions We compute residue-level solvent-accessible surface area (SASA) on AF2-predicted structures using the Shrake–Rupley algorithm (Shrake & Rupley, 1973). Relative SASA (rSASA) is obtained by normalizing each residue's SASA by the theoretical maximum for its amino acid type (Tien et al., 2013).

Based on rSASA, residues are classified into core ($\mathrm{rSASA} < 5\%$), boundary ($5\% \leq \mathrm{rSASA} < 25\%$), and surface ($\mathrm{rSASA} \geq 25\%$) regions (Dauparas et al., 2022).

For each structure, the region fraction is the proportion of all residues assigned to a given region. Within each region, composition is calculated as the proportion of residues classified as hydrophobic (Hb), hydrophilic (Hi), or neutral (Ne).

Table A11: Amino acid composition by structural region in AF2-predicted structures. All values are percentages. **All** denotes whole-protein composition, and **Frac** denotes the fraction of residues assigned to each region. **Hb**, **Hi**, and **Ne** denote hydrophobic, hydrophilic, and neutral residues, respectively; their regional percentages are calculated within each region.

	All			Core				Boundary				Surface			
Model	**Hb**	**Hi**	**Ne**	**Frac**	**Hb**	**Hi**	**Ne**	**Frac**	**Hb**	**Hi**	**Ne**	**Frac**	**Hb**	**Hi**	**Ne**
ProteinMPNN	45.7	47.4	6.8	22.9	85.8	11.6	2.6	26.5	57.3	37.3	5.4	50.6	21.6	68.9	9.5
ProteinMPNN $+\mathcal{L}_{\mathbf{DISCO}}$	42.9	50.9	6.2	**23.9**	85.0	12.8	2.2	26.2	55.5	**39.3**	5.2	**49.9**	**16.2**	**75.3**	8.5
PiFold	49.6	41.9	8.4	11.5	88.7	8.2	3.1	24.2	66.3	27.5	6.1	64.3	36.4	53.4	10.2
PiFold $+\mathcal{L}_{\mathbf{DISCO}}$	44.2	49.1	6.6	**16.1**	88.0	9.3	2.7	25.9	62.4	**31.6**	5.9	**58.0**	**24.0**	**68.0**	8.0
Grade-IF	44.0	30.1	25.9	1.1	73.2	3.8	23.0	8.2	63.3	11.6	25.1	90.7	41.9	32.1	26.0
Grade-IF $+\mathcal{L}_{\mathbf{DISCO}}$	46.5	38.5	14.9	**2.5**	80.2	6.1	13.8	14.8	68.8	**16.7**	14.5	**82.8**	**41.5**	**43.4**	15.1

Hb/Hi/Ne Composition by Region Table A11 summarizes changes in amino acid composition across structural regions with DISCO loss. All three models show an increased core fraction and a decreased surface fraction, while retaining hydrophobic enrichment in the core.

For ProteinMPNN and PiFold, the hydrophobic fraction within surface regions decreases from 21.6% to 16.2% and from 36.4% to 24.0%, respectively, accompanied by an increased hydrophilic fraction. Meanwhile, the hydrophobic fraction within the core remains high at 85.0% and 88.0%, respectively.

For Grade-IF, the surface fraction decreases from 90.7% to 82.8%, while the hydrophilic fraction within surface regions increases from 32.1% to 43.4%. The surface hydrophobic fraction decreases only slightly, from 41.9% to 41.5%, whereas the neutral fraction decreases substantially, from 26.0% to 15.1%. Across the whole sequence, the neutral category comprising glycine and proline decreases from 25.9% to 14.9%. Thus, in ProteinMPNN and PiFold, DISCO loss reduces surface hydrophobicity while maintaining a highly hydrophobic core, consistent with amino acid compositions suited to solvent-exposed and buried environments. In Grade-IF, the main changes are increased surface hydrophilicity and a reduced glycine/proline fraction.

Amino Acid Composition by Region To identify the amino acids contributing to the changes in Table A11, we compare individual amino acid fractions. ProteinMPNN and PiFold show reduced alanine fractions across all regions, particularly at the surface, from 8.8% to 3.6% and from 16.5% to 8.5%, respectively. Together with increased surface glutamate and lysine, these changes contribute to the decreased hydrophobic and increased hydrophilic fractions **(d), (h)**.

Within the core, the overall hydrophobic fraction remains relatively stable, but its composition changes: ProteinMPNN shows decreased valine and increased isoleucine, whereas PiFold shows decreased leucine and increased phenylalanine **(b), (f)**. These shifts are not apparent from the grouped Hb/Hi/Ne analysis.

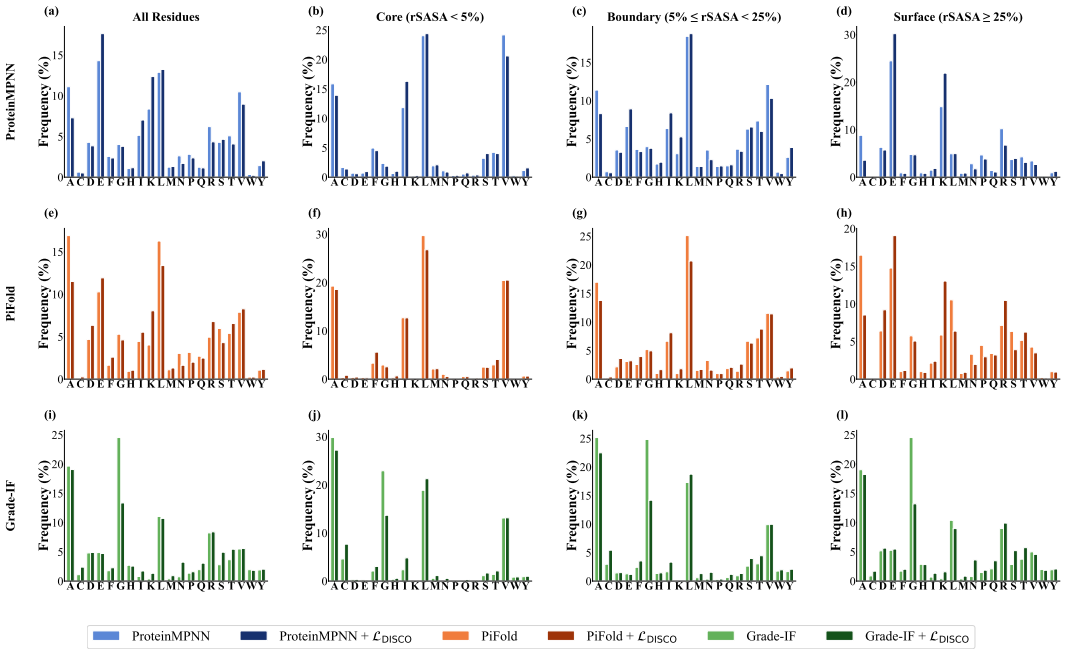

Figure A5: Amino acid composition (%) for ProteinMPNN **(a)–(d)**, PiFold **(e)–(h)**, and Grade-IF **(i)–(l)**. Columns show All, Core, Boundary, and Surface, with regions defined by rSASA on AF2-predicted structures. Lighter and darker bars denote the baseline and $+\mathcal{L}_{\text{DISCO}}$, respectively.

Grade-IF shows decreased glycine fractions across all regions, with the whole-sequence fraction declining from 24.5% to 13.3% **(i)–(l)**. This decrease largely accounts for the reduction in the neutral fraction from 25.9% to 14.9%. At the surface, glycine decreases from 24.5% to 13.2%, while asparagine, serine, and threonine increase from 0.8% to 3.6%, from 2.8% to 5.2%, and from 3.7% to 5.7%, respectively **(l)**. Within the core, cysteine, isoleucine, and leucine increase as glycine decreases **(j)**. Overall, DISCO loss shifts surface amino acid composition toward hydrophilic residues across all three models while maintaining high hydrophobic residue fractions in the cores of ProteinMPNN and PiFold. These changes are consistent with the distinct physicochemical environments of the surface and core.

G.2 Rosetta Energy Evaluation

We compare per-residue Rosetta energies of structures corresponding to sequences generated by ProteinMPNN, PiFold, and Grade-IF, with and without DISCO loss. This analysis complements the structural self-consistency and amino acid composition evaluations by assessing the generated designs under the Rosetta energy function.

G.2.1 Evaluation Dataset

We evaluate Rosetta energies on the AF2-predicted structures used in the self-consistency evaluation. For each backbone length $L \in \{100, 200, 300, 400, 500\}$, we randomly select 50 backbones without replacement using a fixed random seed and without performance-based filtering. The resulting 250 backbones are shared across ProteinMPNN, PiFold, and Grade-IF, with and without DISCO loss, enabling paired comparisons at the backbone level. We evaluate all eight designed sequences per backbone, yielding 2,000 structures per model and training condition.

G.2.2 Rosetta Energy

Rosetta energies are computed using PyRosetta with the `beta_nov16` score function (Alford et al., 2017; Chaudhury et al., 2010). Each AF2-predicted structure undergoes FastRelax (Tyka et al., 2011; Nivon et al., 2013) with coordinate constraints referenced to its initial backbone coordinates,

enabled by setting `constrain_relax_to_start_coords(True)`. These constraints restrain deviations from the starting coordinates during relaxation and are removed before final scoring.

We evaluate both the initial energy of each AF2-predicted structure and its energy after FastRelax using the same score function without coordinate constraints. Energies are normalized by the number of residues and reported in Rosetta energy units per residue (REU/residue), with lower values indicating more favorable scores under the selected energy function.

G.2.3 Length-wise Energy Comparison

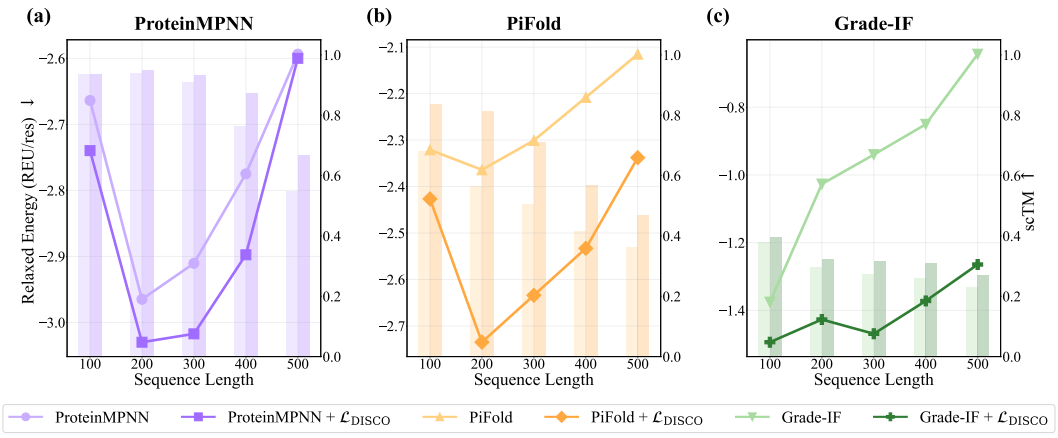

Figure A6: Per-residue Rosetta energy after FastRelax and structural self-consistency across backbone lengths for ProteinMPNN, PiFold, and Grade-IF. Bars show scTM, and lines show Rosetta energy (REU/residue), comparing models trained with and without DISCO loss.

Across the full evaluation set, all three models trained with DISCO loss achieve higher structural self-consistency and lower per-residue Rosetta energy after FastRelax than their baselines. Paired comparisons on the same 250 backbones show scTM increases of 0.035, 0.144, and 0.044 for ProteinMPNN, PiFold, and Grade-IF, respectively, accompanied by decreases in energy after FastRelax of 0.071, 0.227, and 0.441 REU/residue. The 95% confidence intervals estimated by bootstrap resampling stratified by length exclude zero for all six comparisons. These results indicate that improved agreement with the target backbone is accompanied by more favorable Rosetta energy scores.

Energy decreases significantly across all length bins for all three models. Significant scTM increases are observed at every length for PiFold and Grade-IF and at lengths of 300 or greater for ProteinMPNN. For ProteinMPNN at lengths 100 and 200, the confidence intervals for scTM differences include zero, although energy decreases significantly. Thus, improvements in both metrics are observed overall, while energy reductions can occur without a statistically significant increase in scTM for short backbones.

Grade-IF shows the largest energy reduction, but its overall median energy after training with DISCO loss remains higher at -1.391 REU/residue than those of ProteinMPNN (-2.875) and PiFold (-2.537). Although differences in final scores remain across models, the overall increases in scTM and decreases in energy support the benefits of DISCO loss from the complementary perspectives of structural agreement and Rosetta energy. Rosetta total scores also include composition-dependent residue reference energies, so lower scores should not be attributed solely to improved spatial interactions.

H Code Availability

Our code is available at `https://anonymous.4open.science/r/Inverse-Folding-with-Distogram-F1A4/`