FORMATTING INSTRUCTIONS FOR ICLR 2027 CONFERENCE SUBMISSIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding and generation are often treated as two separate paradigms in training deep neural networks, despite the fact that both are trained with closely related objectives such as denoising and masked prediction. While prior studies have shown that generative models often learn suboptimal representations for understanding tasks in vision, it is less understood whether a similar gap exists in the protein domain. In this work, we systematically investigate this question by benchmarking state-of-the-art protein generative models on widely-used protein understanding tasks, and observe that these models exhibit consistently poor performance compared to existing protein encoders. Furthermore, inspired by the Representation Alignment (REPA) framework (50), we propose to explicitly align generative protein diffusion models with pretrained protein understanding models during training. Experiments on the MotifBench demonstrate that representation alignment significantly improves functional protein generation, boosting the MotifBench score of Protpardelle-1c (29) from 39.2 to 47.1 , corresponding to a 20% relative improvement. Our results suggest that representation alignment provides a general and effective mechanism for bridging understanding and generation in protein structure modeling. Model weights and code are available in `https://anonymous.4open.science/r/reapro-AE43/`

1 INTRODUCTION

Protein engineering aims to design and manipulate protein sequences and structures to achieve desired biological functions, and has become a central application of machine learning in the life sciences (31). Broadly, machine learning approaches to protein engineering can be categorized into two core tasks: protein understanding and protein generation. Protein understanding focuses on extracting functional and structural information from existing proteins, while protein generation aims to synthesize novel protein sequences or structures with desired properties (45; 35).

Protein understanding models have achieved remarkable progress in recent years. Representative tasks include inverse folding (7), fold or structure prediction (20), and functional annotation (34), such as enzyme function prediction (51). Models such as protein language models (26) and protein structure encoders (52) have been shown to learn rich representations that are effective for downstream applications, including large-scale screening, functional annotation, and experimental validation. These understanding models are widely used as decision-making tools in protein engineering pipelines, where accurate representations are critical for selecting promising candidates.

In parallel, protein generative models have rapidly advanced, particularly with the adoption of diffusion-based and flow-based architectures for backbone and all-atom structure generation (43; 29; 12). These models have demonstrated strong performance on challenging generation benchmarks, such as motif scaffolding and de novo structure design. However, despite their empirical success, an important question remains largely unexplored: do protein generative models learn representations that are useful for protein understanding tasks? In other words, beyond producing valid structures, do these models implicit learns semantic and functional information in a way that aligns with established protein understanding objectives?

To answer this question, we conduct a systematic benchmarking study of state-of-the-art protein generative models on widely used protein understanding tasks. Our empirical results reveal a consistent gap: representations learned by generative models perform substantially worse than those

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.

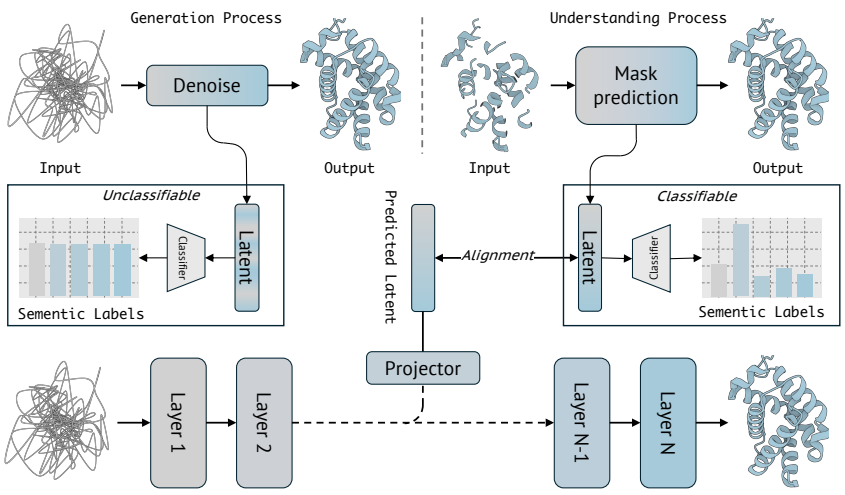

Figure 1: **Diagram of ReaPro-1c.** The Generation Process (left) employs a denoising diffusion model that maps noisy inputs to protein structures, yielding unclassifiable latent features with ambiguous semantic labels; the Understanding Process (right) utilizes a mask prediction model that reconstructs masked regions, producing classifiable latent representations with distinct semantic categories. To bridge this gap, we introduce a Projector within the intermediate layers of the generative model, aligning the denoising latents with the semantically rich understanding space.

from dedicated protein understanding models across multiple evaluation settings. This observation suggests that, similar to recent findings in vision (46), protein generative models may struggle to learn semantically meaningful representations when trained solely with generative objectives.

Motivated by this gap, we further explore whether protein generation can benefit from explicit representation-level guidance. Inspired by the REPresentation Alignment (REPA) framework (50), we propose to align the internal representations of protein diffusion models with pretrained protein understanding models during training. Experiments on the MotifBench (54) benchmark demonstrate that representation alignment significantly improves functional protein generation, increasing the MotifBench score of ProtPardelle-1c (cc58) from 39.2 to 47.1, corresponding to a 20% relative improvement. In another motif-conditional generation benchmark, the RFDiffusion benchmark, our methods overtakes baseline model on 22 out of 26 total motifs. Notably, improvement was obtained by only half training steps with regard to the baseline model, which evidences the importance of representation alignment training.

2 METHODS

Our approach, ReaPro-1c, builds upon Protpardelle-1c (29), a diffusion model for conditional protein structure generation. The key innovation is introducing representation alignment during training: we guide the generation model's intermediate representations to match those of a pretrained protein understanding model.

2.1 MODEL ARCHITECTURE

Protpardelle-1c employs a U-ViT architecture (4) that combines convolutional encoding/decoding with a central transformer. We briefly summarize its key components in the appendix C.

Diffusion objective. The model is trained with the EDM framework (21). Given a noisy structure $\mathbf{x}_t$ at noise level σ, the model predicts the denoised structure $\hat{\mathbf{x}}$. The training loss uses preconditioning weights that depend on the noise level:

$$\mathcal{L}_{\text{struct}} = \mathbb{E}_{\sigma,\mathbf{x}_0,\mathbf{c}}\left[w(\sigma)\cdot\|\hat{\mathbf{x}}(\mathbf{x}_\sigma,\sigma,\mathbf{c})-\mathbf{x}_0\|^2\right] \tag{1}$$

where $w(\sigma) = (\sigma^2 + \sigma_{\text{data}}^2)/(\sigma \cdot \sigma_{\text{data}})^2$ is the EDM loss weight, σ_{data} is the data standard deviation, and $\mathbf{c}$ represents conditioning information (e.g., motif coordinates).

We introduce representation alignment at an intermediate layer of the U-ViT backbone. Specifically, at layer k (default $k = 5$) of the central transformer, we extract the hidden state $\mathbf{h}_k \in \mathbb{R}^{L \times d}$ and project it to the embedding space of a pretrained understanding model.

Alignment head. The projection is performed by an MLP with layer normalization and residual connection:

$$\mathbf{h}_{\text{align}} = \text{MLP}_{\text{align}}(\mathbf{h}_k) + \mathbf{W}_{\text{res}}(\mathbf{h}_k) \tag{2}$$

where the MLP : $\mathbb{R}^{L \times d} \to \mathbb{R}^{L \times d_{\text{align}}}$ is a 3-layer network controlled by a hyperparameter d_{proj}. The residual projection $\mathbf{W}_{\text{res}} : \mathbb{R}^d \to \mathbb{R}^{d_{\text{align}}}$ ensures dimensional compatibility between the diffusion model's hidden state d and the understanding model's embedding dimension d_{align}.

Target embedding. The alignment target is computed by feeding the *clean* ground-truth structure $\mathbf{x}_1$ into a frozen pretrained understanding model $f_{\text{understand}}$:

$$\mathbf{e}_{\text{target}} = f_{\text{understand}}(\mathbf{x}_1) \tag{3}$$

We experiment with two types of understanding models: (1) ProteinMPNN/FA-MPNN (7; 44), a structure encoder trained for inverse folding; and (2) ESM2 (26), a protein language model trained on large-scale sequence data.

2.2 Training Objective

The total training loss combines the structure denoise objective with the alignment loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{struct}} + \lambda \cdot \mathcal{L}_{\text{align}} \tag{4}$$

where λ is a hyperparameter controlling the alignment strength (default $\lambda = 2.0$).

Alignment loss. $\mathcal{L}_{\text{align}}$ is the squared ℓ_2 distance between the projected hidden state and the target embedding:

$$\mathcal{L}_{\text{align}} = \|\mathbf{h}_{\text{align}} - \mathbf{e}_{\text{target}}\|^2 \tag{5}$$

During inference, the alignment head is discarded, which guarantees no additional computational cost is incurred at test time. We provide additional implementation details in Appendix C.

3 Generative models do not learn discriminative features

In this section, we presents a comprehensive evaluation of various protein models on tasks involving the classification of protein functions.

3.1 Experiments Setup

To assess the extent to which various models capture semantic biological knowledge, we evaluated their classification performance on Gene Ontology (GO) terms (1) and Enzyme Commission (EC) numbers (3). High performance on these benchmarks requires a model to internalize complex functional attributes that transcend the mere recognition of local structural geometries. For comprehensive evaluation, we first selected two structural tokenizers, La-Proteina VAE (12) (an all-Atom encoder that compresses side chains into a continuous vector) and Kanzi VQ-VAE (8) (a backbone encoder that encodes backbone atoms to a set of discrete tokens), which are capable of encoding and decoding protein geometries with very high precision. To investigate whether the generative training objective facilitates the acquisition of discriminative biological features, we also evaluated two diffusion models, including the all-atom generative model Protpardelle-1c (29) (cc91) and the backbone generation

Table 1: Performance comparison of different protein models using F_1^{max} metric.

Category	Models	Type/Description	EC	GO-mf	GO-bp	GO-cc
Tokenizers	La-Proteina VAE	All-Atom Continuous	0.1420	0.1978	0.2723	0.2888
	Kanzi VQ-VAE	Backbone Discrete	0.0901	0.1530	0.2489	0.3183
Generation	Protpardelle-1c (cc91)	All-Atom Diffusion	0.1516	0.1946	0.2612	0.3380
	RFDiffusion	Backbone Diffusion	0.4242	0.2914	0.2753	0.3763
Understand	ESM	Protein Language Model	**0.6376**	**0.6117**	**0.4317**	**0.4821**
	ProteinMPNN	Structure Encoder	0.4528	0.4230	0.3375	0.4089

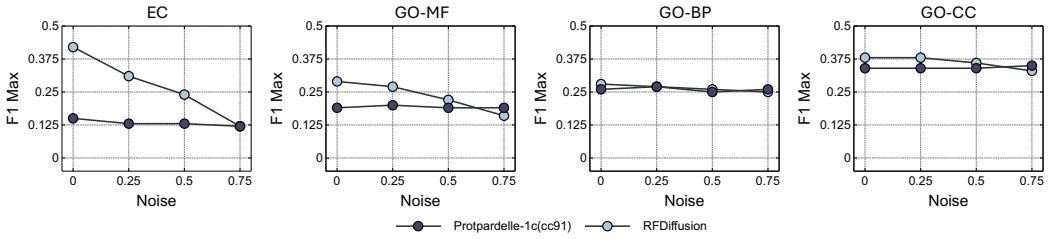

Figure 2: Functional classification performance under varying noise scales $\sigma \in [0, 0.25, 0.5, 0.75]$. Comparison of Protpardelle-1c and RFDiffusion on EC and GO tasks across varying noise levels.

model RFDiffusion (43). For Understanding models, we test the performance of the protein language model ESM (26) and the structure encoder ProteinMPNN (7) as baselines. We provide more details in Appendix B.

3.2 Results

Table 1 summarizes the classification performance of various protein models across the EC and GO benchmarks, as quantified by the $F1_{max}$ metric. Consistent with observations in computer vision, understanding models (ESM and ProteinMPNN) significantly outperform both structure tokenizers and diffusion models in all classification tasks. These results suggest that while tokenizers are capable of accurately reconstructing protein structures from a latent space, they fail to internalize high-level functional semantics during the bottleneck compression and reconstruction process. Similarly, our findings demonstrate that the internal representation of diffusion models remains inferior to that of specialized protein encoders. This gap indicates that while generative objectives excel at structural modeling, they may not naturally align with the capture of discriminative biological function without further refinement.

To investigate whether the inferior performance stems from the noisy training distribution inherent in diffusion processes, we analyze the classification performance of RFDiffusion and Protpardelle-cc91 across varying noise levels $\sigma \in [0, 0.25, 0.5, 0.75]$. As illustrated in Fig 2, functional discriminability for RFDiffusion exhibits a pronounced downward trend as noise intensity increases, whereas this sensitivity is notably weaker for Protpardelle-cc91. In Enzyme Commission (EC) number prediction, RFDiffusion significantly outperforms Protpardelle-cc91 at lower noise levels ($\sigma \leq 0.50$). However, it suffers from a sharp decay in performance as noise increases, with the F^{max} score plummeting from 0.4242 at the zero-noise limit ($\sigma = 0$) to 0.1193 at $\sigma = 0.75$. This initial superiority of RFDiffusion stems from the high-fidelity structural priors embedded within the RoseTTAFold architecture, reflecting the paradigm that structure dictates function. Nevertheless, as noise intensity scales, these structural priors become corrupted, leading to the observed degradation.

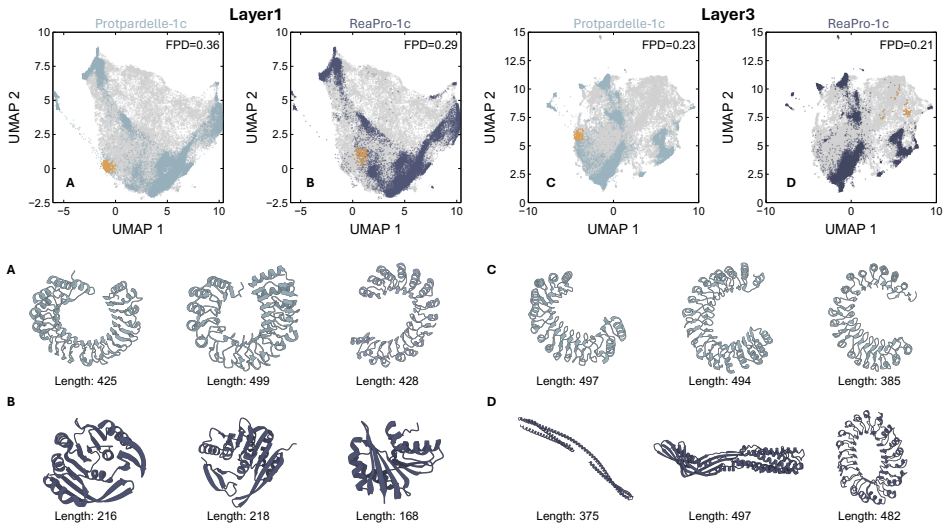

Figure 3: **Distribution coverage comparison between Protpardelle-1c and ReaPro-1c.** UMAP visualization of ProteinMPNN embeddings at Layer 1 and Layer 3. Gray points represent CATH 4.4-S40 reference structures; blue points represent generated samples. Orange points highlight regions where one model shows higher sampling density than the other. FPD scores indicate that ReaPro-1c achieves better alignment with the reference distribution at both layers. (**A**) Visualization of structures where Protpardelle-1c shows higher sampling density than ReaPro-1c; (**B**) Visualization structures where Protpardelle-1c shows higher sampling density than ReaPro-1c. (**C, D**) Additional examples from Layer 3 analysis.

4 Understanding model helps unconditional generation.

In this section, we evaluate the performance of ReaPro-1c on unconditional protein structure generation.

4.1 Experiment Setup

We use CATH 4.4-S40 (36) as the reference dataset to evaluate the distributional coverage of generated structures. This dataset contains 601,328 protein domains with less than 40% sequence identity, representing a diverse set of naturally occurring protein structures. To ensure a fair comparison, we adopt a length-matched sampling strategy. Specifically, we exclude proteins with lengths over 500 and compute the length distribution of proteins in CATH 4.4-S40, resulting in 28,449 structures. For each length L that appears N_L times in the reference set, we generate N_L samples of length L from both the baseline model and our model. This approach ensures that any observed differences in distribution are attributable to the models' generation capabilities rather than length bias. The sampling hyperparameters are kept identical for both models (see Appendix D for details).

We adopt the SHAPES framework (28) to assess the coverage of protein structure space. SHAPES quantifies distributional similarity using Fréchet Protein Distance (FPD), analogous to Fréchet Inception Distance (FID) in image generation. Specifically, we extract structural embeddings using ProteinMPNN at different layers (Layer 1 and Layer 3) and compute the FPD between the generated samples and the reference CATH structures. Lower FPD values indicate better alignment with the natural protein structure distribution.

4.2 Results

Fig. 3 presents the UMAP visualization of ProteinMPNN embeddings for both models alongside the CATH reference set. The quantitative FPD scores reveal that ReaPro-1c consistently achieves better distributional alignment with natural protein structures. The improvement is more pronounced at Layer 1 (0.29 vs 0.36), which captures local structural features, suggesting that the understanding

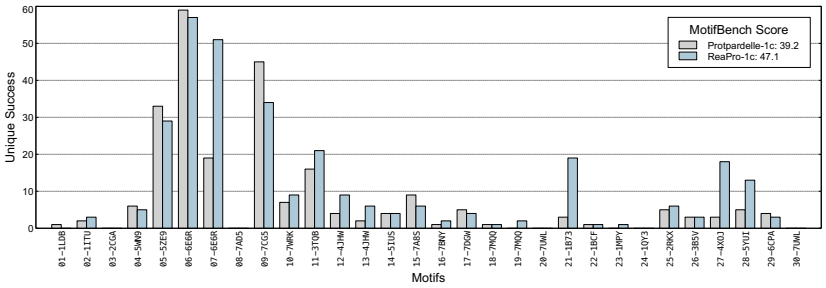

Figure 4: MotifBench results comparing Protpardelle-1c and ReaPro-1c. The bar plot shows the number of unique successful designs per motif. ReaPro-1c achieves a total score of 47.1 compared to Protpardelle-1c's 39.2, successfully designing 25 out of 30 motifs versus 23 for the baseline.

model guidance particularly enhances the generation of chemically plausible local geometries. At Layer 3, which encodes more global structural information, both models perform comparably (0.21 vs 0.23), with ReaPro-1c maintaining a slight advantage.

To gain deeper insights into the differences between the two models, we randomly select regions in the embedding space where one model exhibits significantly higher sampling density than the other (Orange dots in Fig. 3). We then extract the structures falling into these differentially sampled regions and cluster them using Foldseek (41). The cluster centers are visualized to characterize the structural motifs that distinguish each model's generation behavior. As shown in Fig. 3A, regions where Protpardelle-1c exhibits higher sampling density are dominated by large α-helical bundles and ring-like architectures (e.g., lengths 425, 499, 428). This suggests that the baseline model may have a bias toward generating highly regular, designable structures with prominent secondary structure content. In contrast, Fig. 3B shows that regions where ReaPro-1c exhibits higher sampling density feature more diverse structural classes, including β-sheet rich folds (lengths 216, 218) and mixed α/β structures (length 168). The Layer 3 analysis (Fig. 3D) further reveals that ReaPro-1c generates more extended and irregular structures, including elongated helical assemblies (length 375) and complex multi-domain arrangements (lengths 497, 482).

5 Understanding model helps conditional generation.

In this section, we evaluate the effectiveness of our approach on conditional protein generation tasks, specifically focusing on motif scaffolding. Motif scaffolding is a protein design task where the goal is to generate a protein structure that preserves a given functional motif (a set of residues with specific 3D coordinates) while designing the surrounding scaffold. This task is critical for designing proteins with desired functional properties, such as binding sites or catalytic centers.

5.1 MotifBench

We evaluate on MotifBench (54), a standardized benchmark containing 30 diverse motif-scaffolding problems. For each target motif, we generate 100 structures using the same sampling hyperparameters as the baseline. Following the evaluation protocol of Protpardelle-1c (29), we use ProteinMPNN (7) to design 8 amino acid sequences for each generated backbone, then fold these sequences with ESMFold (26). A design is considered successful if the folded N, C_α, and C of motif structure achieves an RMSD of less than 1Å to the target motif, and the C_α of scaffold achieves an RMSD of less than 2Å.

As shown in Figure 4, ReaPro-1c achieves substantial improvements over the baseline. The motifbench score increases from 39.2 to 47.1, representing a 20% improvement. More importantly, ReaPro-1c successfully designs scaffolds for 25 out of 30 motifs, compared to 23 for Protpardelle-1c. This demonstrates that incorporating understanding model guidance not only increases the diversity of successful designs but also expands the range of motifs that can be designed.

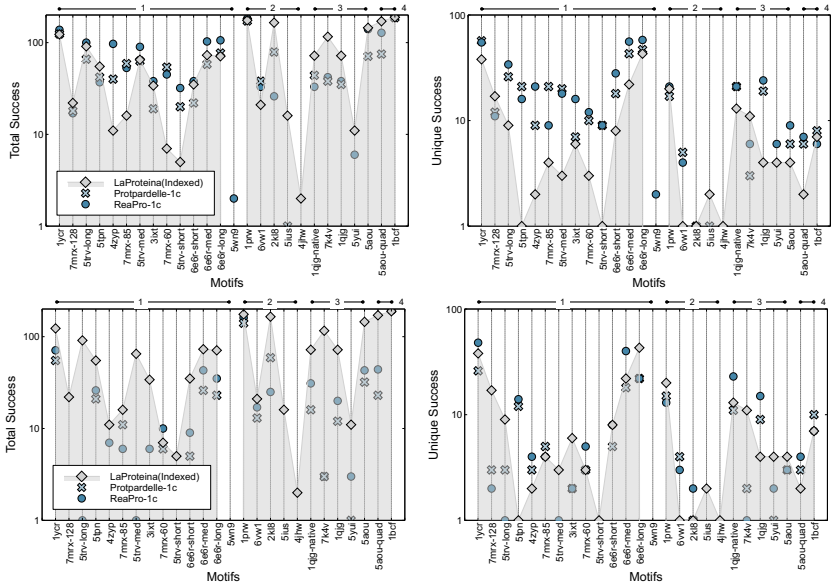

Figure 5: **Top: RFDiffusion backbone generation benchmark results.** (**Left**) Total success count per motif. (**Right**) Unique success count after clustering. Top annotations indicate the number of motif segments (1–4). **Bottom: RFDiffusion all-atom generation results.** (**Left**) Total success count. (**Right**) Unique success count.

5.2 RFDiffusion Benchmark

We further evaluate on the RFDiffusion motif benchmark (43), which contains 26 challenging motif-scaffolding problems with varying complexity (1–4 motif segments). We compare against both Protpardelle-1c and La-Proteina (12), a state-of-the-art flow matching approach for joint sequence-structure generation. For each of the 26 motifs, we generate 200 backbone structures and design a single sequence per structure using ProteinMPNN. Success is determined by ESMFold folding with motif N, C_α, and C RMSD less than 1Å, and the C_α of the scaffold achieves an RMSD of less than 2Å. We report both the total success count (all successful designs) and unique success count (clustered at TM-score < 0.5 using Foldseek (41)).

Figure 5 presents the comparison results. ReaPro-1c demonstrates strong performance compared with Protpardelle-1c and La-Proteina. For total success, ReaPro-1c matches or exceeds La-Proteina on 15 out of 26 motifs, and outperforms Protpardelle-1c on 22 out of 26 motifs. The advantage becomes more pronounced when measuring structural diversity. ReaPro-1c matches or exceeds La-Proteina on 21 out of 26 motifs, and outperforms Protpardelle-1c on 17 out of 26 motifs. This result suggests that the understanding model guidance enables ReaPro-1c to generate a more diverse set of successful scaffolds, even when the total success counts are comparable. The improvement is particularly notable for multi-segment motifs (2–4 segments), where the structural constraints are more complex.

5.3 All-Atom Generation

To evaluate the extension of our approach to all-atom generation, we train a variant following the CC91 setting of protpardelle-1c (29). This model generates full atomistic structures, including side chains. We use the same evaluation protocol as Protpardelle-1c, sample 200 structures per motif, and redesign a single sequence per structure using ProteinMPNN.

As shown in Figure 5, the all-atom generation task presents greater challenges. In total success, ReaPro-1c matches or exceeds La-Proteina on 6 out of 26 motifs, but significantly outperforms Protpardelle-1c on 23 out of 26 motifs. In unique success, ReaPro-1c matches or exceeds La-Proteina on 12 out of 26 motifs, and outperforms Protpardelle-1c on 19 out of 26 motifs. While there remains a performance gap to La-Proteina (which employs a specialized partially latent flow matching

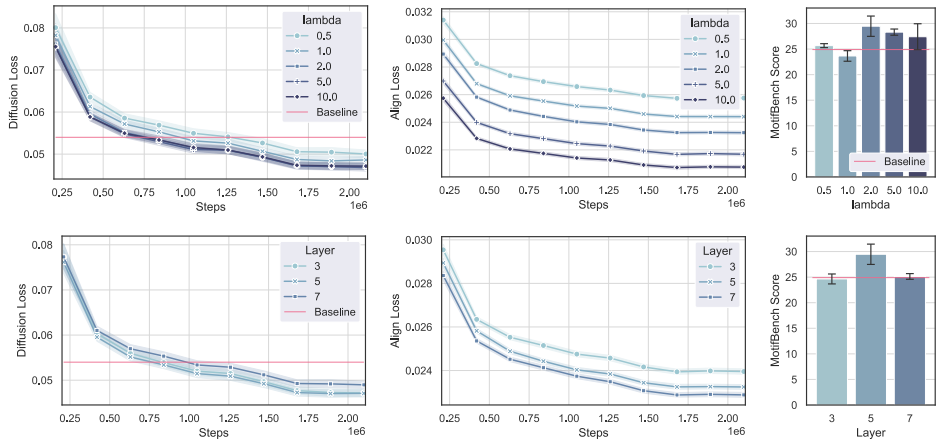

Figure 6: **Ablation on alignment loss weight (λ) and alignment layers.** (**Left**) Diffusion loss curves during training. (**Middle**) Alignment loss curves. (**Right**) MotifBench scores for different λ and layers.

framework for all-atom generation), ReaPro-1c substantially narrows the gap between Protpardelle-1c and La-Proteina. This demonstrates that the understanding model guidance transfers effectively to the more complex all-atom generation setting, even without architectural modifications specifically tailored for atomistic modeling.

6 Ablation Studies

In this section, we conduct ablation studies to investigate the key design choices in our approach: (1) the choice of understanding model for alignment, (2) the weight of the alignment loss, and (3) the layer at which alignment is performed. All ablations are conducted on the MotifBench benchmark following the evaluation protocol described in Section 5.

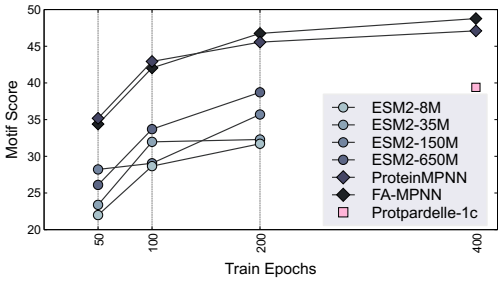

Figure 7: **Ablation on understanding model choice.** MotifBench scores of models trained with different understanding models for representation alignment. All models are trained for 200 epochs with identical hyperparameters except for the alignment target. The baseline (Protpardelle-1c) is trained for 416 epochs.

6.1 Choice of Understanding Model

We first investigate which understanding model provides the most effective guidance for generation. We compare ESM2 models of varying scales (8M, 35M, 150M, 650M parameters) (26) against ProteinMPNN (7), a structure encoder specifically designed for protein inverse folding.

As shown in Figure 7, we observe a clear scaling trend with ESM2 models: larger models consistently yield better generation performance. ESM2-650M achieves a MotifBench score of 38.7 at 200 epochs, approaching the baseline performance (39.4) despite the baseline being trained for 416 epochs. This demonstrates that alignment with stronger sequence understanding models leads to faster convergence and better final performance.

Notably, ProteinMPNN achieves the highest score (47.1), significantly outperforming all ESM2 variants. We hypothesize that while ESM2 excels at protein understanding tasks (e.g., fitness prediction, contact prediction), there exists an inherent modality gap between sequence representations and 3D structures. ProteinMPNN bridges this gap by operating directly on atomic coordinates. To verify this, we conduct an experiments that employ a different structure encoder FA-MPNN (44) to check whether this trend persists. The result of FA-MPNN suggest that structure-aware understanding

models provide more effective guidance for structure generation, even when sequence-based models like ESM2 are scaled up.

6.2 Alignment Loss Weight

Next, we examine the effect of the alignment loss weight λ in Equation 4. We vary $\lambda \in \{0.5, 1.0, 2.0, 5.0, 10.0\}$ while keeping all other hyperparameters fixed. For the motif benchmark, we report the motif score of 3 successive epochs of each setting: 199, 200, and 201, and we design 1 sequence for each structure.

Figure 6 (upper) shows the diffusion loss and alignment loss curves during training. Higher λ values lead to lower alignment loss, indicating a stronger alignment with the understanding model. Notably, stronger alignment also shows lower diffusion loss, evidence that alignment with an understanding model can facilitate generative ability. Over-aggressive alignment makes the improvement of diffusion loss marginal; this trend also coincides with the motifscore, where the performance starts to decline for $\lambda > 2.0$. This suggests an optimal balance: insufficient alignment fails to provide meaningful guidance, while excessive alignment may constrain the generator's exploration of the structure space. We use $\lambda = 2.0$ for all main experiments.

6.3 Alignment Layer Selection

Finally, we investigate at which layer of the understanding model the alignment should be performed. We test aligning at layers 3, 5, and 7 of our model.

As shown in Figure 6 (lower), aligning at layer 5 achieves the best MotifBench performance. Layer 3 (early features) and layer 7 (late features) both underperform, suggesting that mid-level representations strike the best balance between capturing local structural details and global architectural patterns. The alignment loss curves (middle panel) show that layer 7 achieves the lowest alignment loss, likely because late-layer features are more compressed and easier to match. However, this does not translate to better generation performance, indicating that aligning to overly abstract representations may lose structural details critical for generation. We align at layer 5 for all main experiments.

7 Discussion

In this work, we reveal a consistent gap between protein understanding and generation models: state-of-the-art generative models learn suboptimal representations for discriminative tasks. To bridge this gap, we propose ReaPro-1c, which aligns protein diffusion models with pretrained understanding models via a lightweight projection head. Experiments demonstrate that this simple alignment yields substantial improvements across benchmarks—boosting MotifBench scores by 20%, improving distribution coverage on CATH, and accelerating convergence. Our ablations further reveal that structure encoders (ProteinMPNN) provide more effective guidance than sequence encoders (ESM2), highlighting the importance of matching modalities between the understanding target and the generation task.

Limitations and future work. First, we currently align at a single layer; multi-layer alignment could provide richer guidance. Second, extending representation alignment to other protein design tasks, such as binder design and enzyme engineering, and exploring alignment with multiple complementary understanding models are promising avenues for future research. More broadly, recent studies on unifying understanding and generation models have achieved impressive results (RAE (53)), suggesting the potential of directly generating proteins from semantically rich representations, such as those learned by AlphaFold (20).

References

[1] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-

embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15619–15629, 2023.

[3] Amos Bairoch. The enzyme database in 2000. *Nucleic acids research*, 28(1):304–305, 2000.

[4] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22669–22679, 2023.

[5] Avishek Joey Bose, Tara Akhound-Sadegh, Guillaume Huguet, Kilian Fatras, Jarrid Rector-Brooks, Cheng-Hao Liu, Andrei Cristian Nica, Maksym Korablyov, Michael Bronstein, and Alexander Tong. Se (3)-stochastic flow matching for protein backbone generation. *arXiv preprint arXiv:2310.02391*, 2023.

[6] Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. *arXiv preprint arXiv:2401.14404*, 2024.

[7] 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.

[8] Rohit Dilip, Ayush Varshney, Evan Zhang, and David Ashley Van Valen. Flow autoencoders are effective protein tokenizers. *bioRxiv*, pages 2025–10, 2025.

[9] Ada Fang, Zaixi Zhang, Andrew Zhou, and Marinka Zitnik. Atomica: Learning universal representations of intermolecular interactions. *bioRxiv*, pages 2025–04, 2025.

[10] Zhangyang Gao, Cheng Tan, Jue Wang, Yufei Huang, Lirong Wu, and Stan Z Li. Foldtoken: Learning protein language via vector quantization and beyond. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 219–227, 2025.

[11] Benoit Gaujac, Jérémie Donà, Liviu Copoiu, Timothy Atkinson, Thomas Pierrot, and Thomas D Barrett. Learning the language of protein structure. *arXiv preprint arXiv:2405.15840*, 2024.

[12] Tomas Geffner, Kieran Didi, Zhonglin Cao, Danny Reidenbach, Zuobai Zhang, Christian Dallago, Emine Kucukbenli, Karsten Kreis, and Arash Vahdat. La-proteina: Atomistic protein generation via partially latent flow matching. *arXiv preprint arXiv:2507.09466*, 2025.

[13] 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.

[14] Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Milot Mirdita, Martin Steinegger, and Burkhard Rost. Bilingual language model for protein sequence and structure. *NAR Genomics and Bioinformatics*, 6(4):lqae150, 2024.

[15] Pedro Hermosilla and Timo Ropinski. Contrastive representation learning for 3d protein structures. *arXiv preprint arXiv:2205.15675*, 2022.

[16] Po-Ssu Huang, Scott E Boyken, and David Baker. The coming of age of de novo protein design. *Nature*, 537(7620):320–327, 2016.

[17] Guillaume Huguet, James Vuckovic, Kilian Fatras, Eric Thibodeau-Laufer, Pablo Lemos, Riashat Islam, Chenghao Liu, Jarrid Rector-Brooks, Tara Akhound-Sadegh, Michael Bronstein, et al. Sequence-augmented se (3)-flow matching for conditional protein generation. *Advances in neural information processing systems*, 37:33007–33036, 2024.

[18] 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.

[19] Arian R Jamasb, Alex Morehead, Chaitanya K Joshi, Zuobai Zhang, Kieran Didi, Simon Mathis, Charles Harris, Jian Tang, Jianlin Cheng, Pietro Liò, et al. Evaluating representation learning on the protein structure universe. *ArXiv*, pages arXiv–2406, 2024.

[20] 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.

[21] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[23] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18262–18272, 2025.

[24] Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2206–2217, 2023.

[25] Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2. *arXiv preprint arXiv:2405.15489*, 2024.

[26] 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.

[27] Jiarui Lu, Xiaoyin Chen, Stephen Zhewen Lu, Chence Shi, Hongyu Guo, Yoshua Bengio, and Jian Tang. Structure language models for protein conformation generation. *arXiv preprint arXiv:2410.18403*, 2024.

[28] Tianyu Lu, Melissa Liu, Yilin Chen, Jinho Kim, and Po-Ssu Huang. Assessing generative model coverage of protein structures with shapes. *Cell Systems*, 16(8), 2025.

[29] Tianyu Lu, Richard Shuai, Petr Kouba, Zhaoyang Li, Yilin Chen, Akio Shirali, Jinho Kim, and Po-Ssu Huang. Conditional protein structure generation with protpardelle-1c. *bioRxiv*, 2025.

[30] Aleksandr Medvedev, Karthik Viswanathan, Praveenkumar Kanithi, Kirill Vishniakov, Prateek Munjal, Clément Christophe, Marco AF Pimentel, Ronnie Rajan, and Shadab Khan. Biotoken and biofm–biologically-informed tokenization enables accurate and efficient genomic foundation models. *bioRxiv*, pages 2025–03, 2025.

[31] Pascal Notin, Nathan Rollins, Yarin Gal, Chris Sander, and Debora Marks. Machine learning for functional protein design. *Nature biotechnology*, 42(2):216–228, 2024.

[32] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1150–1160, 2020.

[33] Jiezhong Qiu, Junde Xu, Jie Hu, Hanqun Cao, Liya Hou, Zijun Gao, Xinyi Zhou, Anni Li, Xiujuan Li, Bin Cui, et al. Instructplm: Aligning protein language models to follow protein structure instructions. *bioRxiv*, pages 2024–04, 2024.

[34] Jae Yong Ryu, Hyun Uk Kim, and Sang Yup Lee. Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. *Proceedings of the National Academy of Sciences*, 116(28):13996–14001, 2019.

[35] Robert Schmirler, Michael Heinzinger, and Burkhard Rost. Fine-tuning protein language models boosts predictions across diverse tasks. *Nature Communications*, 15(1):7407, 2024.

[36] Ian Sillitoe, Nicola Bordin, Natalie Dawson, Vaishali P Waman, Paul Ashford, Harry M Scholes, Camilla SM Pang, Laurel Woodridge, Clemens Rauer, Neeladri Sen, et al. Cath: increased structural coverage of functional space. *Nucleic acids research*, 49(D1):D266–D273, 2021.

[37] Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. *BioRxiv*, pages 2023–10, 2023.

[38] Shengbang Tong, Boyang Zheng, Ziteng Wang, Bingda Tang, Nanye Ma, Ellis Brown, Jihan Yang, Rob Fergus, Yann LeCun, and Saining Xie. Scaling text-to-image diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2601.16208*, 2026.

[39] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[40] Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Foldseek: fast and accurate protein structure search. *Biorxiv*, pages 2022–02, 2022.

[41] Michel Van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature biotechnology*, 42(2):243–246, 2024.

[42] Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Dplm-2: A multimodal diffusion protein language model. *arXiv preprint arXiv:2410.13782*, 2024.

[43] 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.

[44] Talal Widatalla, Richard W Shuai, Brian L Hie, and Po-Ssu Huang. Sidechain conditioning and modeling for full-atom protein sequence design with fampnn. *Proceedings of machine learning research*, 267:66746, 2025.

[45] Adam Winnifrith, Carlos Outeiral, and Brian L Hie. Generative artificial intelligence for de novo protein design. *Current Opinion in Structural Biology*, 86:102794, 2024.

[46] Weilai Xiang, Hongyu Yang, Di Huang, and Yunhong Wang. Denoising diffusion autoencoders are unified self-supervised learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15802–15812, 2023.

[47] Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Regina Barzilay, Tommi Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

[48] Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023.

[49] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

[50] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

[51] Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. *Science*, 379(6639):1358–1363, 2023.

[52] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

[53] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.

[54] Zhuoqi Zheng, Bo Zhang, Kieran Didi, Kevin K Yang, Jason Yim, Joseph L Watson, Hai-Feng Chen, and Brian L Trippe. Motifbench: A standardized protein design benchmark for motif-scaffolding problems. *arXiv preprint arXiv:2502.12479*, 2025.

A Preliminaries and Related Works

A.1 Protein Understanding

Protein Structure determines how protein performs their biological function (16; 31). In computational settings, protein structures are commonly represented as sets of 3D atomic coordinates. In backbone-centric settings, we model the rigid frames or coordinates of the backbone atoms (N, C_α, C, O). Formally, the backbone coordinates are denoted as $\mathbf{X}^{\text{bb}} \in \mathbb{R}^{L\times 4\times 3}$, where L is the sequence length. For more detailed modeling, a common standardized representation is *Atom37*, which stores up to 37 atom positions per residue: $\mathbf{X}^{37} \in \mathbb{R}^{L\times 37\times 3}$ with a residue-dependent mask selecting the valid subset (43).

To learn from structure, proteins are frequently represented as graphs $G = (V, E)$ where nodes correspond to residues (or atoms) and edges encode spatial proximity or chemical bonds (15). Representative methods like GearNet (52), utilize graph contrastive learning (49; 32) to learn from two augmented structures. ATOMICA (9) learns complex representations by recovering the original atom coordinates and types from a corrupted input. However, due to the scarcity of protein structures, these methods are less competitive than large-scale pre-trained sequence models (19).

Protein Sequence are represented as strings of amino acid tokens from a vocabulary of size 20 (standard amino acids) plus special tokens. Formally, a sequence is denoted as $\mathbf{s} = (s_1, s_2, \ldots, s_L)$ where $s_i \in \mathcal{V}$ and $\mathcal{V}$ is the amino acid vocabulary.

Benefiting from the large-scale protein sequence data, protein language models (PLMs) have become the dominant approach for protein understanding. ESM2 (26) demonstrates that scaling model size and training data leads to emergent capabilities in structure prediction and functional understanding. Recently, there has been growing interest in integrating structural information into PLMs. SaProt (37) and Prot-T5 (14) introduce a structure-aware vocabulary (40) that encodes both sequence and local structural context. This structure-aware vocabulary is small and can only reconstruct rough structures. To eliminate the potential reconstruction bottleneck of structure vocabulary, ESM3 (13), DPLM2 (42) using a VQ-VAE (39), served as a structure tokenizer to first compress structures into a discrete token, then joint training with sequence data (30).

A.2 Protein Generation

From backbone to inverse-folding. As different amino acids have different atoms, a popular design pipeline is using the diffusion model to generate the backbone atoms (N, C, Ca, O) first, including RFDiffusion (43), Protpardelle (29), etc. (48; 18; 47; 5; 17), or Ca-only methods (Proteina and Genie 2 (12; 25)). Followed by an inverse folding method (7; 33) to predict the label of the side chain. Similar to the understanding model, another line of work adopts a structure tokenizer to compress protein structures into a discrete token first, then trains a GPT-style auto-regressive model to generate novel protein structures (27; 11; 10; 8).

Co-design. Recent work increasingly emphasizes co-design, i.e., jointly generating sequence and structure, potentially at the all-atom level. La-Proteina (12) proposes a partially latent flow matching framework for joint generation of protein sequence and fully atomistic structure: it maintains explicit backbone variables while encoding per-residue sequence and side-chain details into fixed-size continuous latents. This hybrid design aims to avoid mixed continuous–categorical modeling complexity and improve scalability for long proteins.

A.3 Alignment of Understanding and Generation

A recurring observation across domains is that generative denoising objectives can induce discriminative representations (24; 6; 2), yet these representations often lag behind those learned by specialized self-supervised encoders. This motivates approaches that explicitly inject or align semantic representations to improve generative modeling. Representation Alignment (REPA) proposes a simple regularization that aligns hidden states of a denoising network operating on noisy inputs with representations computed by an external pretrained encoder on the corresponding clean inputs (50; 23). Instead of adding an auxiliary alignment loss, representation autoencoders(RAEs) replace the conventional VAE encoder in latent diffusion with a frozen pretrained representation

encoder (e.g., DINO/SigLIP/MAE) paired with a lightweight decoder, forming a latent space that is both reconstruction-capable and semantically rich (53). This direction argues that pretrained semantic representations can directly serve as effective latents for diffusion transformers and can yield faster convergence and improved generation quality (38).

B Implement Details of Classification Tasks

B.1 la-proteina

The implementation utilized the publicly available codebase `https://github.com/NVIDIA-Digital-Bio/la-proteina`. For the input pdb, we perform `CenterStructureTransform` and `ChainBreakPerResidueTransform`. Result $L \times 16$ embeddings.

B.2 Kanzi

The implementation utilized the publicly available publicly-released codebase `https://github.com/rdilip/kanzi`. Only support 256, we extend the max length to 1024. For each input pdb, we use the discrete embeddings corresponding to the encoded cod Result $L \times 256$ embeddings.

B.3 ProtPardelle-1c

The implementation utilized the publicly available codebase `https://github.com/ProteinDesignLab/protpardelle-1c/`. Specifically, varying levels of noise ($0.0, 0.25, 0.50$, and 0.75) were added to the protein structures to extract corresponding protein embeddings across these different noise regimes. We extracted the hidden state representations from the U-ViT module, yielding feature dimensions of $L \times 256$.

B.4 RFDiffusion

The implementation utilized the publicly available codebase `https://github.com/RosettaCommons/RFdiffusion`. During the forward pass of the diffusion process, the state_prev embeddings were extracted across four noise regimes ($0.0, 0.25, 0.50$, and 0.75). This process resulted in feature representations with dimensions of $L \times 64$.

B.5 ESM2

For the sequence-based baseline, we utilized the ESM-2 650M parameter model (`esm2_t33_650M_UR50D`). Protein representations were obtained by extracting the embeddings from the 33rd layer (the final hidden layer), resulting $L \times 1280$ embeddings.

B.6 ProteinMPNN

The implementation followed the standard ProteinMPNN architecture for structure-based encoding. We extracted the output embeddings directly from the structural encoder to serve as the fixed-length representation for the downstream functional classification tasks. Result $L \times 128$ embeddings.

B.7 Classification head

To aggregate the residue-level embeddings into a fixed-size protein-level representation, we employ an attention-based pooling mechanism. Let $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_L] \in \mathbb{R}^{L \times D}$ denote the sequence of embeddings for a protein of length L. The attention score e_i for the i-th residue is computed as:

$$e_i = \mathbf{w}^\top \tanh(\mathbf{W}_a \mathbf{h}_i + \mathbf{b}_a) + b_w$$

where $\mathbf{W}_a \in \mathbb{R}^{H \times D}$ and $\mathbf{w} \in \mathbb{R}^H$ are learnable parameters of the hidden and scoring layers, respectively. To ensure the model focuses only on valid residues, we apply a mask M such that

$e_i = -\infty$ for padding tokens. The normalized attention weights α_i are then calculated via the Softmax function:

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{L} \exp(e_j)}$$

The final aggregated protein representation $\mathbf{z} \in \mathbb{R}^D$ is obtained as the weighted sum:

$$\mathbf{z} = \sum_{i=1}^{L} \alpha_i \mathbf{h}_i$$

The global vector $\mathbf{z}$ is subsequently passed through a readout network Φ to generate the final predictions $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = \mathbf{W}_2 \cdot \text{Dropout}(\text{ReLU}(\mathbf{W}_1 \mathbf{z} + \mathbf{b}_1)) + \mathbf{b}_2$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ represent the weight matrices of the two-layer Multi-Layer Perceptron (MLP). This architecture allows the model to adaptively highlight functional motifs within the protein sequence for classification.

C Training Details of ReaPro-1c

C.1 Model Architecture

We adopt the same model architecture as the original Protpardelle-1c (29).

Input representation. The model operates on all-atom protein coordinates $\mathbf{X} \in \mathbb{R}^{L \times A \times 3}$, where L is the sequence length and A is the number of atoms per residue (37 for Atom37 representation).

U-ViT backbone. The architecture consists of: (1) downsampling convolutional blocks that process atomic coordinates into patch embeddings; (2) a central Time-Conditional Transformer with rotary positional embeddings; and (3) upsampling convolutional blocks that reconstruct the full-resolution output. The transformer contains 10 layers by default, with each layer comprising time-conditional self-attention and feed-forward modules.

We use cc58 and cc91 as the base models for backbone design and all-atom design, respectively. For the alignment head, we use a 3-layer MLP to project the original hidden state $\mathbf{h}_k \in \mathbb{R}^{L \times d}$ into the space of the understanding hidden state $\mathbf{h}_{\text{align}}$. The pseudo code of the alignment head is shown in Algorithm 1.

Tab.2 shows the detailed model parameters we used in the main results.

Name	Value
layers	10
n_channel	256
dim_head	32
rope_dim	32
n_blocks_per_layer	2
n_heads	8
patch_size	1

Table 2: Model parameters of ReaPro-1c.

C.2 Dataset

Following the same setting of Protpardelle-1c, we adopt the augmented Ingraham CATH dataset for training. The dataset includes 337,936 structures.

C.3 Training Parameters

We use Adam (22) optimizer with a batch size of 32. We start the training process with a 1000-step linear warm-up, increase the learning rate to 1×10^{-4}, and using an cosine annealing to decrease the

Algorithm 1 Alignment Head Architecture

```
class AlignmentHead(nn.Module):
    def __init__(self, hidden_size, projector_dim, z_dim):
        super().__init__()
        # Residual projection for dimensional matching
        self.residue_proj = nn.Linear(hidden_size, z_dim)

        # 3-layer MLP with SiLU activation
        self.mlp = nn.Sequential(
            nn.LayerNorm(hidden_size),
            nn.Linear(hidden_size, projector_dim),
            nn.SiLU(),
            nn.Linear(projector_dim, projector_dim),
            nn.SiLU(),
            nn.Linear(projector_dim, z_dim),
            nn.LayerNorm(z_dim)
        )

    def forward(self, h_k):
        # MLP pathway
        out = self.mlp(h_k)

        # Residual connection with projection
        residue = self.residue_proj(h_k)

        return out + residue
```

learning rate to 1×10^{-6} in 2,000,000 steps. We use the same crop config as Protpardelle-1c. We train the model for 401 epochs. For the all-atom model, we use the same setting but train for 201 epochs. All experiments are done on a single Nvidia A100 GPU.

D Evaluation Details of ReaPro-1c

D.1 Motifbench

To get the final score, we first generate 100 structures and design 1 sequence per structure with ProteinMPNN. We use models of 399, 400, and 401 epochs. Then we choose models with the best score and generate 100 structure deign 8 sequences per structure to get the final results. We use the same sampling parameters as original Protpardelle-1c, with SCHURNS = 200 and STEP_SCALES = 1.2.

D.2 RFDiffusion Motif Benchmark

For the backbone design task, we use the same model and the same sampling parameters as the motifbench to better test the generalization ability on different datasets. For the all-atom design task, we follow the same protocol as motifbench. We first generate a small set (100 sequences for each motif) to select a checkpoint between 199, 200, and 201 epochs. Then we use the best checkpoint to obtain the final result. Note that the baseline model was trained for 383 epochs, which is significantly longer than our model.

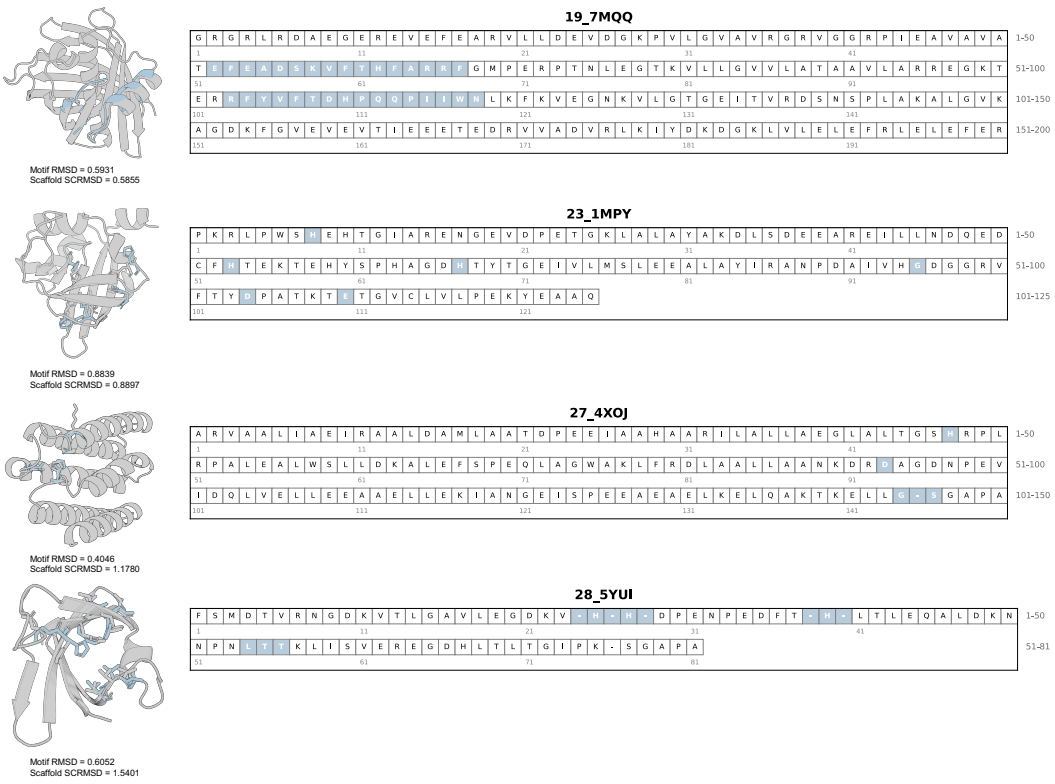

Figure 8: Successfully designed case of MotifBench with ReaPro-1c

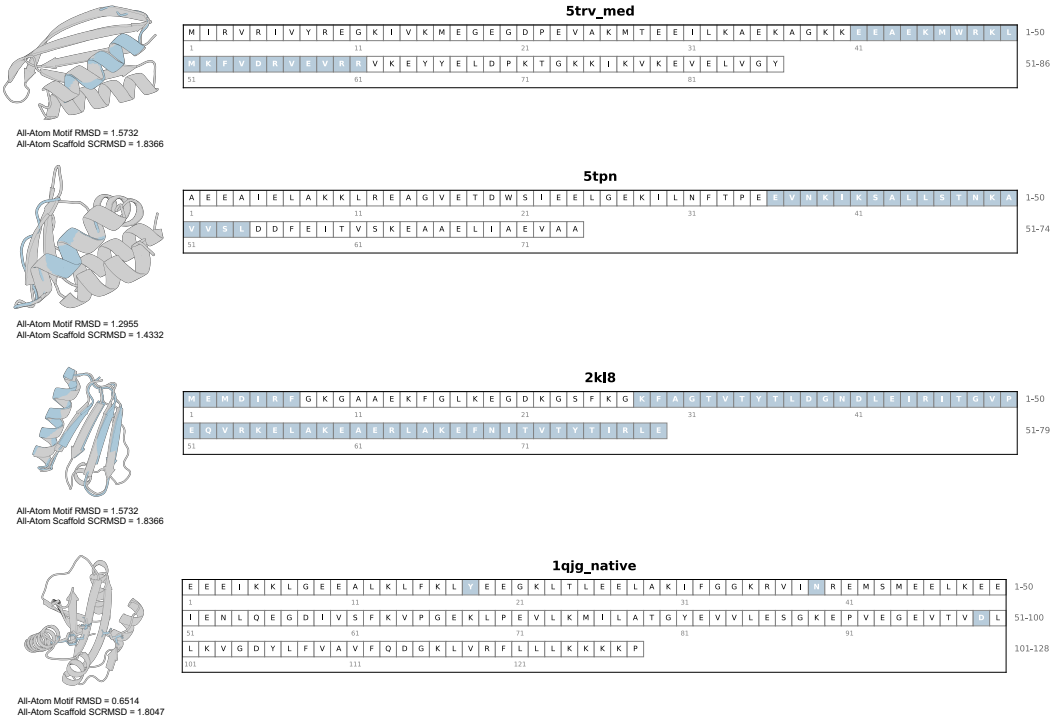

Figure 9: Successfully designed case of RFDiffusion benchmark with ReaPro-1c