HiGenIF: Hierarchical Structural Context Modeling for Protein Inverse Folding

Anonymous authors
Paper under double-blind review

Abstract

Protein inverse folding aims to design amino-acid sequences compatible with a target backbone. Protein function emerges from coordinated structural environments spanning local, regional, and protein-wide scales. However, regional relationships often remain implicit in residue-level representations, leaving the connection between structural organization and position-specific biochemical preferences insufficiently explicit. We propose **HiGenIF**, which explicitly organizes protein-wide, regional, and residue-level structural context to predict position-specific amino-acid preferences. The resulting complete sequence draft and continuous structural states jointly condition ESM2-650M, which incorporates structural information at an intermediate layer and predicts residual corrections to the initial logits. On the CATH 4.2 topology split, HiGenIF achieves 59.37% amino-acid recovery and 3.40 perplexity across 1,120 test chains, establishing the highest recovery within our focused comparison of inverse-folding methods that incorporate pretrained sequence knowledge. Controlled experiments further demonstrate the effectiveness of explicit structural-region modeling and pretrained sequence refinement.

1 Introduction

Protein design aims to create amino-acid sequences with desired structural and functional properties, supporting applications in enzyme engineering, therapeutics, and biotechnology (Watson et al., 2023; Wang et al., 2022). Protein function emerges from coordinated structural environments: local neighborhoods shape residue interactions, regional organization brings these interactions together, and the overall fold provides the scaffold that supports them (Halabi et al., 2009; Wang et al., 2022). Protein inverse folding focuses on one aspect of this broader goal: predicting amino-acid sequences compatible with a prescribed three-dimensional backbone.

Recent advances have substantially improved protein inverse folding through geometric representation learning and sequence modeling. Structure-based methods, including GVP, ProteinMPNN, and PiFold, encode backbone geometry to predict amino-acid preferences using graph neural networks and autoregressive or parallel decoders (Jing et al., 2021; Dauparas et al., 2022; Gao et al., 2022). Generative approaches further explore the space of compatible sequences through iterative sampling (Yi et al., 2023; Bai et al., 2025). Alongside these developments, methods such as LM-Design and Bridge-IF incorporate pretrained protein language models to combine structural information with sequence regularities learned from large protein databases (Zheng et al., 2023; Zhu et al., 2024). These advances improve geometric representations and sequence priors, while leaving open how structural regions can be explicitly represented and integrated into residue-level sequence prediction.

A specific challenge is to connect local residue environments with protein-wide context through explicit regional representations. Residue-level message passing and attention can capture relationships across a structure, but do not necessarily organize them into identifiable structural regions. Likewise, incorporating a pretrained sequence model does not by itself provide this intermediate structural representation. This motivation has a biological basis: surrounding scaffolds provide structural support for functional sites (Wang et al., 2022), while residue networks spanning multiple secondary-structure elements can collectively contribute to protein function (Halabi et al., 2009). These observations motivate an explicit regional representation, whose utility we evaluate here through backbone-conditioned sequence prediction.

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.

To address this challenge, we propose **HiGenIF**, a protein inverse-folding framework that combines explicit hierarchical structural modeling with pretrained sequence refinement (Figure 1). To make regional organization explicit, its structural predictor represents contiguous secondary-structure segments as regions and integrates their representations with local residue geometry and protein-wide context. This hierarchy provides each residue with structural context at multiple scales for predicting initial amino-acid preferences. To complement these structure-derived preferences with learned sequence regularities, the refinement stage conditions ESM2-650M (Lin et al., 2023) on the complete predicted sequence draft and continuous structural states. The draft provides an initial sequence hypothesis, while the structural states supply information beyond the discrete amino-acid choices. An intermediate-layer interface incorporates this structural context into ESM2, which predicts residual corrections to the initial logits.

Our contributions are as follows:

- **Explicit hierarchical structural modeling.** We develop a structural predictor that integrates protein-wide context, regional organization, and local residue environments to predict position-specific amino-acid preferences.
- **Structure-conditioned sequence refinement.** We combine the complete predicted draft with continuous hierarchical structural representations in a pretrained protein language model, enabling sequence refinement informed by both structural context and sequence knowledge.
- **Empirical effectiveness and extensibility.** HiGenIF achieves 59.37% amino-acid recovery and 3.40 perplexity on CATH 4.2, establishing the highest recovery within our focused comparison of inverse-folding methods that incorporate pretrained sequence knowledge (Table 1). Evaluation on TS50 and TS500 supports cross-dataset transfer, while controlled experiments demonstrate the benefits of regional organization and pretrained refinement. HiGenIF also complements external generators, raising recovery from 55.32% to 68.15% for DPLM-2 Bit (Hsieh et al., 2025) and from 45.98% to 61.58% for ADFLIP at flow time $t = 0$ (Yi et al., 2025).

2 Related work

Backbone-conditioned inverse folding. Graph-based models combine geometric representations with autoregressive decoding (Ingraham et al., 2019; Dauparas et al., 2022), one-shot prediction (Gao et al., 2022), or denoising diffusion (Yi et al., 2023). GVP introduces scalar–vector representations (Jing et al., 2021), while ESM-IF1 scales training with predicted structures (Hsu et al., 2022). These methods primarily organize structural computation around residues. HiGenIF explicitly incorporates intermediate regions into a hierarchy linking protein-wide context to position-specific amino-acid preferences.

Multiscale structural context. Functional scaffolding, protein sectors, and structure-based sequence design motivate modeling coordinated residue environments (Wang et al., 2022; Halabi et al., 2009; Ding et al., 2024; Krapp et al., 2024). ProRefiner incorporates global graph attention (Zhou et al., 2023), and SurfPro hierarchically encodes continuous protein surfaces (Song et al., 2024). FractalGen composes atomic generative modules recursively across resolutions (Li et al., 2025). HiGenIF transfers this cross-resolution principle to protein inverse folding by organizing backbone-derived context through interactions within and between contiguous secondary-structure regions, integrating protein-wide information, and using coarse biochemical preferences to condition amino-acid prediction. These backbone-derived regions provide explicit computational units for regional structural modeling.

Pretrained sequence priors and multimodal generators. LM-Design adapts a protein language model through a structural adapter and iterative refinement (Zheng et al., 2023). DPLM-2 Bit models sequences and tokenized structures (Hsieh et al., 2025), while ADFLIP uses discrete flow matching with all-atom structural context (Yi et al., 2025). HiGenIF places its explicit protein-wide–regional–residue hierarchy before ESM2 refinement, which receives a hierarchy-conditioned draft and continuous structural states. We additionally evaluate DPLM-2 Bit and ADFLIP as external generators guided by HiGenIF's complete structure-conditioned output distribution.

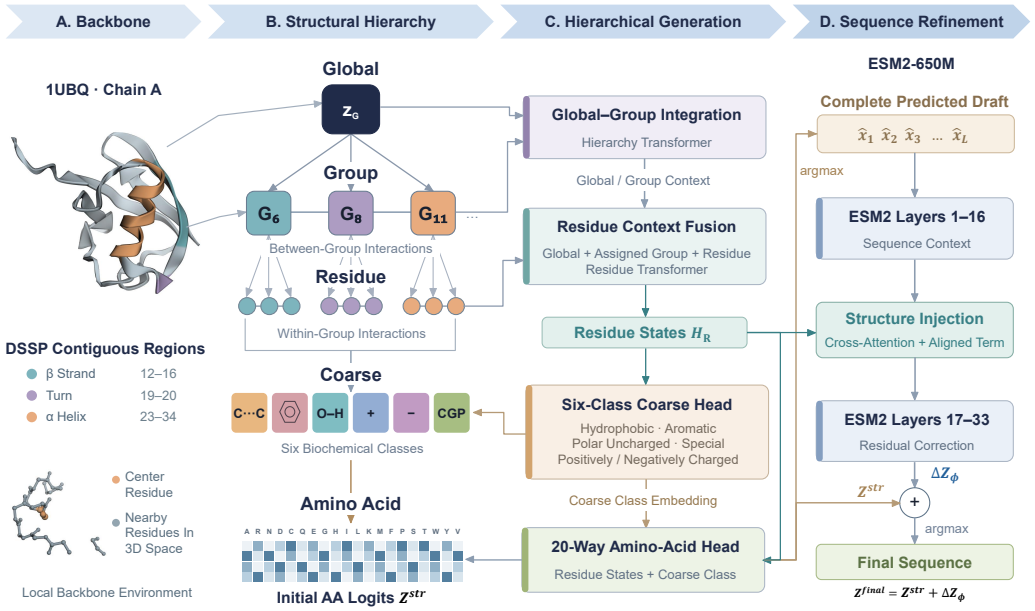

Figure 1: Overview of HiGenIF. (A) The target backbone provides local geometry and contiguous DSSP regions, illustrated with ubiquitin (PDB: 1UBQ, chain A). (B) The structural hierarchy connects global, group, and residue-level context to coarse biochemical classes and amino-acid preferences; selected groups are shown. (C) Global–group integration and residue-context fusion produce H_R. A six-class prediction conditions, but does not restrict, the 20-way head that produces Z^{str}. (D) ESM2-650M processes the complete predicted sequence draft $\widehat{X}$, receives residue-level structural memory after layer 16 through cross-attention and a position-aligned term, and predicts the residual correction ΔZ_ϕ. The final sequence is the position-wise argmax of $Z^{\mathrm{final}} = Z^{\mathrm{str}} + \Delta Z_\phi$. The amino-acid heatmap is schematic.

3 Method

3.1 Overview

Figure 1 separates the backbone and structural hierarchy (A–B), the hierarchical prediction modules (C), and pretrained sequence refinement (D). Let S denote the processed structural input for a protein of length L, including backbone coordinates, dihedral and secondary-structure signals, solvent accessibility, and geometry-derived graph features. Let $x_i \in \mathcal{A}$ be the amino acid at position i, where $\mathcal{A}$ is the 20-type amino-acid alphabet. At inference, the amino-acid head receives an all-mask input, and HiGenIF maps S to continuous residue-level structural states $H_R \in \mathbb{R}^{L\times 256}$ and initial amino-acid logits $Z^{\mathrm{str}} \in \mathbb{R}^{L\times 20}$. Their position-wise argmax defines a complete predicted sequence draft $\widehat{X}$:

$$\begin{aligned}(H_R, Z^{\mathrm{str}}) &= f_\theta(S),\\ Z^{\mathrm{final}} &= Z^{\mathrm{str}} + \Delta Z_\phi(H_R, \widehat{X}),\\ p(x_i = a \mid S) &= \mathrm{softmax}(Z_i^{\mathrm{final}})_a.\end{aligned} \tag{1}$$

Here, θ is the structural predictor, which is frozen in Stage 2, and ϕ comprises ESM2, the structural interface, and the residual head. Since H_R and $\widehat{X}$ are deterministic functions of S at inference, the final distribution remains structure-conditioned. HiGenIF realizes the structural stage as a Global–Group–Coarse–AA cascade before ESM2 supplies the pretrained sequence prior.

Design rationale. Drawing inspiration from FractalGen's composition of generative modules across resolutions (Li et al., 2025), we organize the structural predictor as a protein-specific cascade from

protein-wide and regional context to biochemical preferences and amino-acid predictions. Global, group, and residue states separate protein-wide, regional, and position-specific context before fusion. The coarse module supplies an intermediate biochemical preference without restricting the 20-way output space.

3.2 HIERARCHICAL STRUCTURAL PREDICTION

Structural encoding and DSSP groups. We represent S as a residue graph $G_S = (V, E)$, where V indexes residues and E contains geometry-derived residue edges. Native amino-acid one-hot input channels are fixed to zero for every example before an E(n)-equivariant graph neural network (EGNN) produces $H^{(0)} = [h_1^{(0)}; \dots; h_L^{(0)}]$. We partition each chain into maximal contiguous runs sharing one of the standard eight residue-aligned DSSP labels (Kabsch & Sander, 1983). Unknown labels form singleton groups, noncontiguous runs remain separate, and group sizes are not capped. We denote the resulting groups by $\{\mathcal{G}_k\}_{k=1}^K$ and the assignment of residue i by $g(i)$. These backbone-derived assignments provide annotation-free regional computational units.

Regional and global context. Two message-passing layers first update residues using only graph edges whose endpoints lie in the same group. Attention pooling then forms a summary $v_k^{(0)}$ for each group. We connect every group to up to eight spatially nearest groups and apply two further message-passing layers whose inputs include group-center distance and mean sequence-index distance:

$$\begin{aligned} u_i &= f_{\text{intra}}\Big(h_i^{(0)}, \{h_j^{(0)} : j \in \mathcal{N}_{\text{in}}(i)\}\Big), \\ v_k^{(0)} &= \sum_{i \in \mathcal{G}_k} \beta_{ki} u_i, \qquad \sum_{i \in \mathcal{G}_k} \beta_{ki} = 1, \\ \bar{h}_k &= f_{\text{inter}}\Big(v_k^{(0)}, \{v_j^{(0)}, \|\mu_j - \mu_k\|_2, |s_j - s_k|\}_{j \in \mathcal{N}_{\text{grp}}(k)}\Big), \end{aligned} \tag{2}$$

where μ_k and s_k are the mean coordinate and mean sequence index of group k, and f_{intra} and f_{inter} denote the corresponding two-layer message-passing networks. A separate attention pool over the encoder states produces the global summary $\bar{h}_G$. Broadcasting $\bar{h}_{g(i)}$ back to residue i forms a group-conditioned residue summary. Learned projections map the global, group, and residue streams to width $d = 256$, yielding initial states $z_G^{(0)}$, $z_{\text{grp},k}^{(0)}$, and $h_{R,i}^{(0)}$, respectively. Detailed operators are given in Appendix A.1.

A two-layer hierarchy Transformer maps the initial tokens to refined states z_G and $\{z_{\text{grp},k}\}_{k=1}^K$. For residue i, its global and assigned-group states form $q_i = W_c[z_G; z_{\text{grp},g(i)}]$, which is fused with $h_{R,i}^{(0)}$ through a learned gate:

$$\begin{aligned} \gamma_i &= \sigma\Big(\text{MLP}_{\text{gate}}[W_R h_{R,i}^{(0)}; q_i]\Big), \\ \widetilde{h}_i &= \text{LN}\Big(W_R h_{R,i}^{(0)} + \text{Dropout}(\gamma_i \odot q_i)\Big), \\ H_R &= f_{\text{res}}([\widetilde{h}_1; \dots; \widetilde{h}_L]). \end{aligned} \tag{3}$$

Here, brackets with semicolons denote concatenation, W_c and W_R are learned linear maps, σ is the sigmoid, $\odot$ is element-wise multiplication, and LN is layer normalization. The function f_{res} is a two-layer residue Transformer. Padding is excluded from both hierarchy- and residue-level attention.

Coarse-to-amino-acid prediction. A six-way head predicts one of six mutually exclusive biochemical classes at every position: hydrophobic (A, I, L, M, V), aromatic (F, W, Y), polar uncharged (N, Q, S, T), positively charged (R, H, K), negatively charged (D, E), or special (C, G, P):

$$C = f_{\text{coarse}}(H_R) \in \mathbb{R}^{L \times 6}, \qquad \widehat{c}_i = \arg\max_{r \in \{1,\dots,6\}} C_{i,r}. \tag{4}$$

Here, C contains coarse-class logits and $\widehat{c}_i$ is the predicted class. Its learned embedding conditions, but does not restrict, the 20-way prediction. Given $x^{\text{in}} \in (\mathcal{A} \cup \{\texttt{MASK}\})^L$, let $b_i = \mathbb{1}[x_i^{\text{in}} \neq \texttt{MASK}]$. With learned embedding functions E_c, E_x, and E_b, a two-layer Transformer predicts

$$Z^{\text{AA}} = f_{\text{AA}}\big(H_R, E_c(\widehat{c}), E_x(x^{\text{in}}), E_b(b)\big) \in \mathbb{R}^{L \times 20}. \tag{5}$$

During structural training, x^{in} is a randomly masked native sequence. In the complete model, the head is invoked once with $x^{\text{in}} = [\texttt{MASK}]^L$, giving $Z^{\text{str}} = Z^{\text{AA}}$ in parallel. It is not used as an iterative or autoregressive decoder. Training and inference both use the predicted hard coarse class; its argmax blocks the exact-amino-acid gradient, so the coarse head is trained by its own loss.

3.3 Structure-guided sequence refinement

We use ESM2-650M, a 33-layer protein language model with width 1,280 (Lin et al., 2023). The structural logits define a complete predicted sequence draft,

$$\widehat{x}_i = \arg\max_a Z^{\text{str}}_{i,a}, \qquad \widehat{X} = (\widehat{x}_1, \ldots, \widehat{x}_L). \tag{6}$$

Training and inference both use this draft, which is generated entirely by the structural predictor; native amino acids are never supplied as sequence-model inputs.

The draft supplies complete discrete sequence context, whereas H_R retains information lost by argmax decoding. The residual form and zero initialization make the initial refinement model reproduce the structural predictor before adaptation.

After ESM layer 16, let $U \in \mathbb{R}^{L\times 1280}$ be the residue hidden states. We zero-pad H_R to width 1,280 to obtain structural memory M, then project the sequence queries and structural memory to width 128:

$$\begin{aligned} Q &= W_Q \text{LN}_Q(U), \qquad B = W_M \text{LN}_M(M), \\ A &= \text{MHA}(Q, B, B), \\ U' &= U + W_O(A + B). \end{aligned} \tag{7}$$

Here, $Q, B, A \in \mathbb{R}^{L\times 128}$, MHA denotes four-head multihead attention over valid residue positions, and W_O maps width 128 back to 1,280. Cross-attention A provides whole-chain structural context, while the position-aligned row B_i preserves the state of residue i. ESM layers 17–33 process U'. From their output U^{out}, a 20-way head predicts corrections at every valid residue:

$$\Delta Z_\phi = W_{\text{AA}} \text{LN}(U^{\text{out}}) + b_{\text{AA}}, \qquad Z^{\text{final}} = Z^{\text{str}} + \Delta Z_\phi. \tag{8}$$

The interface output projection and residual head are zero-initialized, so adaptation begins from $Z^{\text{final}} = Z^{\text{str}}$. Corrections apply to every valid residue position.

3.4 Training and inference

Stage 1 trains the structural model. For each batch, we draw $\rho \sim \mathcal{U}(0.15, 1.0)$ and independently mask every valid residue with probability ρ; at least one position is masked. The model minimizes

$$\mathcal{L}_{\text{base}} = 0.5\mathcal{L}_{\text{group}} + 0.5\mathcal{L}_{\text{coarse}} + \mathcal{L}_{\text{AA}}, \tag{9}$$

where $\mathcal{L}_{\text{group}}$ is the mean squared consistency loss between refined and initial valid global/group tokens, $\mathcal{L}_{\text{coarse}}$ is six-way cross-entropy over all residues, and $\mathcal{L}_{\text{AA}}$ is 20-way cross-entropy only over masked residues. Unmasked native tokens provide partial context to the AA head during this stage, while the structural graph encoder receives structure-only inputs throughout training and inference. Exact definitions are in Appendix A.1.

Stage 2 freezes the structural model and adapts ESM2 and the new interface to minimize full-residue cross-entropy,

$$\mathcal{L}_{\text{refine}} = -\frac{1}{N}\sum_{i=1}^{N} \log \text{softmax}(Z_i^{\text{final}})_{x_i^\star}. \tag{10}$$

Here, N is the total number of valid, non-padding residues in the batch and $x_i^\star$ is the native amino-acid target at position i. The input is always the predicted draft in Eq. 6; native amino acids appear only as targets. Optimization and checkpoint-selection details are reported in Appendix A.

Inference derives the final sequence from structure through one structure-only initial prediction and one structure-guided ESM2 refinement pass, with position-wise predictions produced in parallel by taking the argmax of Z^{final}.

4 EXPERIMENTS

Our experiments follow the main claims of HiGenIF from overall performance to the contribution of its design. We first evaluate the complete model on CATH 4.2, then test whether the same locked checkpoint transfers to TS50 and TS500. We next ask whether the learned structure-conditioned distribution remains useful outside the standalone model by combining it with external protein generators. Finally, matched controls isolate the roles of explicit structural regions and pretrained sequence refinement. Reproducibility and implementation details are reported in Appendix A.

4.1 EXPERIMENTAL SETUP

Datasets and model selection. We use the CATH 4.2 topology split curated by Ingraham et al. (Ingraham et al., 2019), whose nominal split contains 18,024 training, 608 validation, and 1,120 test proteins. After preprocessing and holding out 256 training-partition chains for development, ESM2 adaptation uses 17,647 training chains. For ESM2 adaptation, we select checkpoints by residue-weighted development recovery, breaking ties by lower negative log-likelihood. The official validation partition is used for additional evaluation and for selecting the external-generator fusion rules. The locked CATH test evaluation is performed only after selection and contains all 1,120 chains and 180,387 valid residues. TS50 and TS500 are used only for zero-shot evaluation and never for training, adaptation, or model selection, following the standard ProteinInvBench protocol (Gao et al., 2023).

Metrics and inference. Amino-acid recovery (AAR) is the fraction of valid residues whose highest-probability amino acid matches the native residue. Perplexity (PPL) is the exponential of the mean native-residue cross-entropy over all valid residues. At inference, HiGenIF first produces a complete predicted sequence draft from structure with all native amino-acid identities hidden, followed by one ESM2-650M refinement pass using that predicted draft.

Reference baselines. We focus the comparison on established backbone-conditioned models and representative pretrained-sequence approaches with closely related structural inputs. LM-Design uses ESM-1b 650M, and Bridge-IF uses ESM2-650M. We also include recent structural models ScFold, FAMPNN, and ProtSeqGen, with the structure-feedback-trained InstructPLM-DPO included in the transfer evaluation. Reference sources are summarized in Appendix A.9. HiGenIF uses ESM2-650M (Lin et al., 2023) together with DSSP labels and processed solvent-accessibility features. Controlled evidence for regional conditioning and the refinement mechanism is presented in Section 4.5, with full optimization and checkpoint details in Appendix A.

4.2 STANDALONE INVERSE-FOLDING PERFORMANCE

We begin with the central question of whether explicit hierarchical structural modeling and pretrained sequence refinement form an effective complete inverse-folding model. Table 1 reports CATH 4.2 results on the standard Short, Single-chain, and All test sets. HiGenIF obtains 52.05% AAR and 4.41 PPL on Short, 51.67% and 4.35 on Single-chain, and 59.37% and 3.40 on All. On the All test set, HiGenIF achieves the highest recovery within our focused comparison of inverse-folding methods that incorporate pretrained sequence knowledge. Compared with the closest ESM2-650M Bridge-IF configuration, HiGenIF improves AAR from 58.59% to 59.37% and PPL from 3.83 to 3.40.

4.3 GENERALIZATION TO UNSEEN STRUCTURES

Having established performance on the CATH topology split, we next test whether the learned structure-to-sequence mapping transfers to independently curated structures. We use the same locked CATH-trained checkpoint for zero-shot evaluation on TS50 and TS500, with the model and hyperparameters fixed before evaluation. As shown in Table 2, HiGenIF reaches 64.83% AAR and 2.83 PPL on TS50, and 67.96% AAR and 2.60 PPL on TS500. It improves AAR over LM-Design (PiFold) by 6.94 percentage points on TS50 and 0.18 percentage points on TS500. Relative to the recent ScFold and InstructPLM-DPO models, HiGenIF improves recovery by 5.51 and 2.82 points on TS50, and by 6.37 and 1.59 points on TS500, respectively. These results support the transferability of HiGenIF's hierarchical structural representation and pretrained sequence refinement.

Table 1: CATH 4.2 inverse-folding performance on Short ($L \leq 100$), Single-chain, and All test sets. HiGenIF is evaluated on 94, 103, and 1,120 chains, respectively; each entry is PPL / AAR (%), and dashes denote unreported results.

Model	Short	Single-chain	All
GraphTrans (Ingraham et al., 2019)	8.39 / 28.14	8.83 / 28.46	6.63 / 35.82
StructGNN (Ingraham et al., 2019)	8.29 / 29.44	8.74 / 28.26	6.40 / 35.91
GVP (Jing et al., 2021)	7.23 / 30.60	7.84 / 28.95	5.36 / 39.47
ProteinMPNN (Dauparas et al., 2022)	6.21 / 36.35	6.68 / 34.43	4.61 / 45.96
PiFold (Gao et al., 2022)	6.04 / 39.84	6.31 / 38.53	4.55 / 51.66
FAMPNN (single-step) (Widatalla et al., 2025)	– / –	– / –	4.99 / 49.66
ScFold (Zhong et al., 2025)	5.80 / 41.60	5.99 / 40.10	4.61 / 52.22
ProtSeqGen (Gao et al., 2026)	– / –	– / –	4.43 / 53.03
LM-Design (PiFold) (Zheng et al., 2023)	6.77 / 37.88	6.46 / 42.47	4.52 / 55.65
Bridge-IF (Zhu et al., 2024)	5.68 / 43.86	5.06 / 48.96	3.83 / 58.59
HiGenIF	**4.41 / 52.05**	**4.35 / 51.67**	**3.40 / 59.37**

Table 2: Zero-shot generalization from CATH 4.2 to TS50 and TS500 using the locked development-selected HiGenIF checkpoint. Reference-source details appear in Appendix A.9.

Model	TS50		TS500	
	PPL ↓	AAR (%) ↑	PPL ↓	AAR (%) ↑
GVP (Jing et al., 2021)	4.71	44.14	4.20	49.14
ProteinMPNN (Dauparas et al., 2022)	3.93	54.43	3.53	58.08
PiFold (Gao et al., 2022)	3.86	58.72	3.44	60.42
LM-Design (PiFold) (Zheng et al., 2023)	3.50	57.89	3.19	67.78
ScFold (Zhong et al., 2025)	3.71	59.32	–	61.59
ProtSeqGen (Gao et al., 2026)	3.79	58.51	3.37	60.59
InstructPLM-DPO (Xu et al., 2025)	**2.52**	62.01	**2.17**	66.37
HiGenIF	2.83	**64.83**	2.60	**67.96**

4.4 Compatibility with external generators

The first two evaluations treat HiGenIF as a complete model. We next test a different property of the learned structure-conditioned distribution: whether it provides reusable information that complements generators with distinct architectures. We combine it with DPLM-2 Bit (Hsieh et al., 2025) and with ADFLIP at flow time $t = 0$ (Yi et al., 2025). In our integration, DPLM-2 Bit is used for backbone-conditioned sequence prediction, providing amino-acid logits and a generated-sequence vote under a one-step decoding configuration. ADFLIP receives only N/CA/C/O coordinates with all amino-acid identities masked. Temperatures and fusion weights are selected on the 608-protein validation split and fixed before a single evaluation on the test set; native sequences are never used as model inputs.

As shown in Table 3, combining HiGenIF with DPLM-2 Bit raises AAR from 55.32% to 68.15% and lowers PPL from 4.40 to 2.96. Combining it with ADFLIP raises AAR from 45.98% to 61.58% and lowers PPL from 5.50 to 3.17. Both combinations also improve over standalone HiGenIF, demonstrating that its structure-conditioned distribution provides complementary information across distinct generative architectures.

To make this complementarity visible at the residue level, Table 4 shows representative 48-residue windows from three chains in the locked test evaluation.

On 3ggm.A, HiGenIF obtains 70.67% AAR compared with 44.00% for DPLM-2 Bit, and fusion further raises recovery to 73.33%. On 2r9i.A, the corresponding values are 62.32%, 47.83%, and 71.01%. The third example, 4qiw.K, shows a complementary regime: fusion reaches 58.93%, compared with 41.07% for HiGenIF and 42.86% for DPLM-2 Bit. The chains are selected by the

Table 3: HiGenIF combined with external generators on the locked CATH 4.2 test set. Fusion rules are selected exclusively on the validation set, and all configurations use structure-derived inputs.

Configuration	**PPL** ↓	**AAR (%)** ↑
HiGenIF	3.40	59.37
DPLM-2 Bit	4.40	55.32
HiGenIF + DPLM-2 Bit	**2.96**	**68.15**
ADFLIP $t = 0$	5.50	45.98
HiGenIF + ADFLIP $t = 0$	**3.17**	**61.58**

Table 4: Representative 48-residue windows from three chains in the locked test set. Panels A–B show strong standalone HiGenIF predictions; panel C shows a complementary case with the largest fusion gain over the stronger standalone component. Within each chain, the displayed window maximizes positions where HiGenIF is correct and DPLM-2 Bit is incorrect. Native residues are black, correct predictions green, and errors red.

Prediction	**Displayed sequence window**
A **3ggm.A** · residues 28–75 of 75	
Native	ITDGLITAVG GDELLNSATE KTKKIDLKRK RAIPGLNDSH IHVIRGLE
HiGenIF	ITDGKILAVG GEELLDACDE KTEMIDAKQK QVIPGLNDAH IHVIRGLE
DPLM-2 Bit	IKDGKVKAIG GDALLDLADE GTLKIDARNF TVIPALDNKK LKTVLGKK
HiGenIF + DPLM-2 Bit	IKDGKILAVG GEELLDLADE KTEKIDAKQK RVIPGLNDSH IHVIRGLE
B **2r9i.A** · residues 2–49 of 69	
Native	LKDLLAHREN LDSAKRARSA ITDDDPADAA QAVENVKSII SEIESTDE
HiGenIF	LKDLVAQRQE LDSARRARDS ITDTDPAEAA QADEDVKSIE AQIDATDQ
DPLM-2 Bit	LKKLKKKLER LEEARDANAD LNLDTPEEAA AAIENVASIV AEISATRE
HiGenIF + DPLM-2 Bit	LKDLIAQRKD LDSARRARSS ITLDDPAEAA QAVENVKSIV AEIDATQE
C **4qiw.K** · residues 4–51 of 56	
Native	TRFEKARIIG ARALQIAMGA PVLIDVPEGI TPLQAALLEF EKGVIPIT
HiGenIF	SYKEFDYALG ALAEILAKGA PVVKLTPLGD TPLELALLLA DRGVCPIT
DPLM-2 Bit	PKYEIVGILG KRARQLALGA PLLPLPPSGD KPVQRALKEE RRGVVPIR
HiGenIF + DPLM-2 Bit	TRYEIVRILG ARALQLALGA PLLKLVPLGD TPLELALEEE ERGVVPIT

deterministic procedures in Appendix A.5 to visualize strong standalone structural predictions and complementary residue-level corrections; aggregate results are computed over all 1,120 test chains.

4.5 Evidence for regional conditioning and refinement

The preceding results establish overall performance, transfer, and reuse. We now test whether the gains align with the two defining design choices of HiGenIF. We first examine the regional level with a single-group control that assigns all residues in each chain to one group while retaining the global and residue pathways and coarse biochemical conditioning. Both variants use 30,000 structural training updates and the same ESM2-650M adaptation with a complete predicted sequence draft. On the locked CATH test set, the regional model raises AAR from 54.95% to 59.08% and lowers PPL from 3.92 to 3.40 (Table 5). With the remaining pathways and training budgets matched, this 4.13-percentage-point gain demonstrates the contribution of explicit regional organization within HiGenIF.

We next isolate the contribution of pretrained refinement by evaluating HiGenIF's structural stage before ESM2. This structure-only model obtains 51.89% AAR and 4.26 PPL, whereas complete HiGenIF reaches 59.37% AAR and 3.40 PPL (Table 5). Thus, combining the predicted structural draft

with pretrained sequence refinement improves recovery by 7.48 percentage points while substantially reducing perplexity. Detailed controls for the refinement interface are provided in Appendix A.7.

Table 5: Controlled evidence for HiGenIF's regional organization and pretrained refinement on CATH 4.2. Each block reports its corresponding matched comparison.

Configuration	**PPL** $\downarrow$	**AAR (%)** $\uparrow$
Regional organization (matched ESM2 refinement)		
Single group per chain	3.92	54.95
DSSP-defined regions	**3.40**	**59.08**
Pretrained sequence refinement		
HiGenIF structural stage (without ESM2)	4.26	51.89
Complete HiGenIF (with ESM2-650M)	**3.40**	**59.37**

The two blocks address distinct stages of the framework. In the regional comparison, both structural variants receive the same ESM2 refinement procedure, so the comparison tests the value of regional organization in the complete pipeline. In the refinement comparison, the structural checkpoint is fixed, allowing us to assess the benefit of adding pretrained sequence knowledge to its predictions. The 59.08% regional result belongs to the matched 30,000-update structural experiment, whereas 59.37% is obtained with the main structural checkpoint. Reading each block against its own control connects the observed gains to the corresponding design choice and training budget. In particular, the single-group control retains the global and residue pathways and biochemical conditioning. The regional comparison therefore evaluates the organization of structural context while preserving these complementary sources of information.

Together, these experiments establish the highest CATH 4.2 recovery within our focused comparison of inverse-folding methods that incorporate pretrained sequence knowledge. Controlled experiments further demonstrate the effectiveness of explicit structural-region modeling and pretrained sequence refinement. Evaluation on TS50 and TS500 supports transfer with the same locked checkpoint, while combinations with external generators demonstrate complementary predictive information.

5 Conclusion

We presented HiGenIF, a hierarchical structure–sequence framework for protein inverse folding. Its structural predictor integrates protein-wide context, regional interactions, and coarse biochemical preferences to produce initial amino-acid distributions. The complete predicted draft and continuous structural states then condition ESM2-650M, which refines these distributions through residual corrections in a single pass.

On the CATH 4.2 topology split, the complete model achieves 59.37% amino-acid recovery and 3.40 perplexity, establishing the highest recovery within our focused comparison of inverse-folding methods that incorporate pretrained sequence knowledge. A matched regional-partition control raises recovery from 54.95% to 59.08%, and pretrained refinement raises recovery from 51.89% for the structure-only stage to 59.37% for the complete model. The same locked CATH-trained checkpoint further reaches 64.83% recovery and 2.83 perplexity on TS50, and 67.96% recovery and 2.60 perplexity on TS500. HiGenIF's structure-conditioned distribution also complements external generators: validation-selected combinations reach 68.15% recovery with DPLM-2 Bit and 61.58% with ADFLIP at flow time $t = 0$. Together, these results demonstrate that hierarchical regional modeling and pretrained sequence refinement form an effective inverse-folding framework whose structure-conditioned predictions also complement distinct protein generators.

HiGenIF currently models structural hierarchies from a single backbone conformation. Extending this framework to conformational ensembles is a promising direction for capturing structural relationships across different protein states. Our evaluation establishes performance in backbone-conditioned sequence prediction; whether the resulting designs improve protein function or stability remains to be tested experimentally. Integrating multiple conformations and experimental feedback provides a natural next step toward functional protein design.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and language revision, literature searches and organization, discussions of experimental design, writing and debugging some experimental scripts, organizing experimental outputs, and designing and arranging paper illustrations. AI-assisted experimental analyses were grounded in actual run records and model evaluation outputs, and literature citations were checked against original papers and official publication records. The authors have reviewed all AI-generated and AI-assisted content and take full responsibility for the final manuscript and research results, including the methods, code, experimental results, figures, tables, and claims.

ETHICS STATEMENT

This work studies amino-acid sequence design conditioned on a given protein backbone, using existing protein structure datasets and computational evaluation. The reported sequence recovery, perplexity, and cross-dataset results characterize performance on the evaluated computational tasks; they do not establish the biological function, experimentally measured stability, or application safety of designed proteins.

Protein design methods have potential benefits for basic research, enzyme engineering, and biotechnology, but may also be misused. Our study and evaluation focus on general inverse folding and do not aim to enhance harmful biological functions. Any subsequent experimental validation or practical application should undergo appropriate biosafety and ethical review for its intended use and comply with the licenses governing the datasets, pretrained models, and software used.

REPRODUCIBILITY STATEMENT

The Method section describes HiGenIF's architecture, training objectives, and inference procedure. The Experiments section and Appendix A specify the data splits, preprocessing, optimization settings, training budgets, random seeds, checkpoint selection, and metric aggregation. The appendix further details structural-feature caching, external-generator integration protocols, and controlled experimental configurations. Checkpoints are selected on development data, and external-generator fusion parameters are selected exclusively on validation data; the test set is not used for either selection. We also report results across sequence-refinement seeds with a fixed structural checkpoint to characterize training stability in this stage.

REFERENCES

Peizhen Bai, Filip Miljković, Xianyuan Liu, Leonardo De Maria, Rebecca Croasdale-Wood, Owen Rackham, and Haiping Lu. Mask-prior-guided denoising diffusion improves inverse protein folding. *Nature Machine Intelligence*, 7(6):876–888, 2025. doi: 10.1038/s42256-025-01042-6. URL https://www.nature.com/articles/s42256-025-01042-6.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Alexis Courbet, Rob J. de Haas, Neville Bethel, Peter J. Y. Leung, Timothy F. Huddy, Eva-Maria Strauch, Frank DiMaio, Bryan North, and David Baker. Robust deep learning-based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187. URL https://www.science.org/doi/10.1126/science.add2187.

David Ding, Ada Y Shaw, Sam Sinai, Nathan Rollins, Noam Prywes, David F Savage, Michael T Laub, and Debora S Marks. Protein design using structure-based residue preferences. *Nature Communications*, 15(1):1639, 2024.

Qiang Gao, Zhijin Li, Yang Deng, and Zhiwei Ji. ProtSeqGen: a novel deep learning model for protein sequence design. *BMC Bioinformatics*, 27:156, 2026. doi: 10.1186/s12859-026-06482-4.

Zhangyang Gao, Cheng Tan, Pablo Chacón, and Stan Z Li. PiFold: Toward effective and efficient protein inverse folding. *arXiv preprint arXiv:2209.12643*, 2022. URL https://arxiv.org/abs/2209.12643.

Zhangyang Gao, Cheng Tan, Yijie Zhang, Xingran Chen, Lirong Wu, and Stan Z. Li. ProteinInvBench: Benchmarking protein inverse folding on diverse tasks, models, and metrics. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, volume 36, 2023. doi: 10.52202/075280-2983. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d73078d49799693792fb0f3f32c57fc8-Abstract-Datasets_and_Benchmarks.html.

Najeeb Halabi, Olivier Rivoire, Stanislas Leibler, and Rama Ranganathan. Protein sectors: Evolutionary units of three-dimensional structure. *Cell*, 138(4):774–786, 2009. doi: 10.1016/j.cell.2009.07.038. URL https://doi.org/10.1016/j.cell.2009.07.038.

Cheng-Yen Hsieh, Xinyou Wang, Daiheng Zhang, Dongyu Xue, Fei Ye, Shujian Huang, Zaixiang Zheng, and Quanquan Gu. Elucidating the design space of multimodal protein language models. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2504.11454.

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*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8946–8970. PMLR, 2022. URL https://proceedings.mlr.press/v162/hsu22a.html.

John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. *Advances in Neural Information Processing Systems*, 32, 2019. URL https://papers.nips.cc/paper/9711-generative-models-for-graph-based-protein-design.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael J. L. Townshend, and Ron O. Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=1YLJDvSx6J4.

Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983. doi: 10.1002/bip.360221211.

Lucien F Krapp, Fernando A Meireles, Luciano A Abriata, Jean Devillard, Sarah Vacle, Maria J Marcaida, and Matteo Dal Peraro. Context-aware geometric deep learning for protein sequence design. *Nature Communications*, 15(1):6273, 2024.

Tianhong Li, Qinyi Sun, Lijie Fan, and Kaiming He. Fractal generative models. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=Qk9kn6lOlW.

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. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/10.1126/science.ade2574.

Zhenqiao Song, Tinglin Huang, Lei Li, and Wengong Jin. SurfPro: Functional protein design based on continuous surface. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46074–46088. PMLR, 2024. URL https://proceedings.mlr.press/v235/song24c.html.

Jue Wang, Sidney Lisanza, David Juergens, Doug Tischer, Joseph L. Watson, Karla M. Castro, Robert Ragotte, Amijai Saragovi, Lukas F. Milles, Minkyung Baek, Ivan Anishchenko, Wei Yang, Derrick R. Hicks, Marc Expòsit, Thomas Schlichthaerle, Jung-Ho Chun, Justas Dauparas, Nathaniel Bennett, Basile I. M. Wicky, Andrew Muenks, Frank DiMaio, Bruno Correia, Sergey Ovchinnikov, and David Baker. Scaffolding protein functional sites using deep learning. *Science*, 377(6604):387–394, 2022. doi: 10.1126/science.abn2100. URL https://www.science.org/doi/10.1126/science.abn2100.

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 L. Hie, and Po-Ssu Huang. Sidechain conditioning and modeling for full-atom protein sequence design with FAMPNN. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EE8Z8Se5Te.

Junde Xu, Zijun Gao, Xinyi Zhou, Jie Hu, Xingyi Cheng, Le Song, Guangyong Chen, Pheng-Ann Heng, and Jiezhong Qiu. Protein inverse folding from structure feedback. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/22ed94fe31a4c9561a5cf7fb4d868523-Paper-Conference.pdf.

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. URL https://openreview.net/forum?id=u4YXKKG5dX.

Kai Yi, Kiarash Jamali, and Sjors H. W. Scheres. All-atom inverse protein folding through discrete flow matching. *arXiv preprint arXiv:2507.14156*, 2025. URL https://arxiv.org/abs/2507.14156.

Zaixiang Zheng, Yifan Deng, Dongyu Xue, Yi Zhou, Fei Ye, and Quanquan Gu. Structure-informed language models are protein designers. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42317–42338. PMLR, 2023. URL https://proceedings.mlr.press/v202/zheng23a.html.

Jiancheng Zhong, Zhiwei Zou, Jie Qiu, and Shaokai Wang. ScFold: a GNN-based model for efficient inverse folding of short-chain proteins via spatial reduction. *Briefings in Bioinformatics*, 26(2): bbaf156, 2025. doi: 10.1093/bib/bbaf156.

Xinyi Zhou, Guangyong Chen, Junjie Ye, Ercheng Wang, Jun Zhang, Cong Mao, Zhanwei Li, Jianye Hao, Xingxu Huang, Jin Tang, et al. ProRefiner: An entropy-based refining strategy for inverse protein folding with global graph attention. *Nature Communications*, 14(1):7434, 2023. doi: 10.1038/s41467-023-43166-6. URL https://www.nature.com/articles/s41467-023-43166-6.

Yiheng Zhu, Jialu Wu, Qiuyi Li, Jiahuan Yan, Mingze Yin, Wei Wu, Mingyang Li, Jieping Ye, Zheng Wang, and Jian Wu. Bridge-IF: Learning inverse protein folding with markov bridges. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Q8yfhrBBD8.

A IMPLEMENTATION AND EVALUATION DETAILS

This appendix connects the method definitions to the reported experiments. We first specify the structural operators and the training procedures for the two stages, then define the common evaluation protocol. We next detail external-generator integration, adaptation-seed stability, and the controls for continuous structural memory and regional organization. The final section documents reference sources.

A.1 DETAILED STRUCTURAL OPERATORS AND OBJECTIVES

We expand the structural predictor in Section 3 by following the computation from residue features to regional and global representations, and then specifying the Stage-1 objectives. The structural graph contains normalized Cα coordinates $r_i \in \mathbb{R}^3$, backbone dihedral and secondary-structure signals, multiscale neighborhood-direction statistics, and scalar attributes including solvent-accessible surface area. Edges encode sequence separation, radial distance, contacts, and local-frame orientations. The encoder updates features without changing coordinates.

For intra-group processing, let $\mathcal{N}_{\text{in}}(i) = \{j : (j,i) \in E,\ g(j) = g(i)\}$ and $u_i^{(0)} = h_i^{(0)}$. Two layers use

$$\begin{aligned} m_{ji}^{(\ell)} &= \text{MLP}_{\text{in}}^{(\ell)}([u_j^{(\ell)}; u_i^{(\ell)}; \|r_j - r_i\|_2]), \\ u_i^{(\ell+1)} &= \text{LN}_{\text{in}}^{(\ell)}\left(u_i^{(\ell)} + \frac{\sum_{j\in\mathcal{N}_{\text{in}}(i)} m_{ji}^{(\ell)}}{\max(1, |\mathcal{N}_{\text{in}}(i)|)}\right). \end{aligned} \tag{11}$$

Each message MLP has two linear layers with SiLU activation. If a batch has no retained within-group edge, this encoder is bypassed. Learned scalar attention gives

$$\beta_{ki} = \frac{\exp(w_{\text{in}}^\top u_i^{(2)})}{\sum_{j\in\mathcal{G}_k} \exp(w_{\text{in}}^\top u_j^{(2)})}, \qquad v_k^{(0)} = \sum_{i\in\mathcal{G}_k} \beta_{ki} u_i^{(2)}. \tag{12}$$

The pooled group summaries provide the input to inter-group processing. For each group, $\mu_k = |\mathcal{G}_k|^{-1} \sum_{i\in\mathcal{G}_k} r_i$ and $s_k = |\mathcal{G}_k|^{-1} \sum_{i\in\mathcal{G}_k} (i-1)$. Spatial nearest-neighbor selection forms $\mathcal{N}_{\text{grp}}(k)$, and two inter-group layers compute

$$\begin{aligned} a_{jk}^{(\ell)} &= \text{MLP}_{\text{grp}}^{(\ell)}([v_j^{(\ell)}; \|\mu_j - \mu_k\|_2; |s_j - s_k|]), \\ v_k^{(\ell+1)} &= \text{LN}_{\text{grp}}^{(\ell)}\left(v_k^{(\ell)} + \frac{\sum_{j\in\mathcal{N}_{\text{grp}}(k)} a_{jk}^{(\ell)}}{\max(1, |\mathcal{N}_{\text{grp}}(k)|)}\right). \end{aligned} \tag{13}$$

The final group state is $\bar{h}_k = v_k^{(2)}$. A separate attention pool gives $\bar{h}_G = \sum_i \alpha_i h_i^{(0)}$, and group context is broadcast as $\bar{h}_i^R = h_i^{(0)} + W_f \bar{h}_{g(i)}$. Independent learned projections produce $z_G^{(0)} \in \mathbb{R}^{1\times 256}$, $Z_{\text{grp}}^{(0)} \in \mathbb{R}^{K\times 256}$, and $H_R^{(0)} \in \mathbb{R}^{L\times 256}$. Invalid padded group and residue tokens are excluded by attention masks.

The resulting representations support the hierarchy-conditioned prediction heads. We now define their training objectives. For a mini-batch $\mathcal{B}$, let $\mathcal{T}_b$ index the valid global and group tokens of protein b, Ω_b its randomly masked residues, and $c_{b,i}^\star$ and $x_{b,i}^\star$ its native coarse-class and amino-acid targets. Let $z_{b,u}^{(0)}$ and $z_{b,u}$ denote the same global or group token immediately before and after the hierarchy Transformer, respectively; the former comes from the same hierarchy constructor rather than a separate target network. Finally, $P_b^{\text{mask}} = \text{softmax}(Z_b^{\text{AA}})$ denotes the amino-acid probabilities for the masked-input training call. The Stage-1 losses are

$$\begin{aligned} \mathcal{L}_{\text{group}} &= \frac{\sum_b \sum_{u\in\mathcal{T}_b} \|z_{b,u} - z_{b,u}^{(0)}\|_2^2}{\sum_b |\mathcal{T}_b|}, \\ \mathcal{L}_{\text{coarse}} &= -\frac{\sum_b \sum_{i=1}^{L_b} \log \text{softmax}(C_{b,i})_{c_{b,i}^\star}}{\sum_b L_b}, \\ \mathcal{L}_{\text{AA}} &= -\frac{\sum_b \sum_{i\in\Omega_b} \log [P_b^{\text{mask}}]_{i,x_{b,i}^\star}}{\sum_b |\Omega_b|}. \end{aligned} \tag{14}$$

The pre-Transformer states $z_{b,u}^{(0)}$ serve as the references in $\mathcal{L}_{\text{group}}$; no separate target encoder is used. Predicted hard coarse classes condition the amino-acid head.

A.2 Structural implementation and checkpoint selection

We next describe how the structural predictor is trained and how its outputs are prepared for sequence refinement. The graph encoder has hidden dimension 128 and six layers, and the hierarchy-conditioned residue states have dimension 256. The processed encoder includes normalization whose output can depend on the graphs in a batch. We therefore preserve the fixed preprocessing cohorts when caching structural states and logits: training, validation, and test cohorts contain up to 16 chains, while development cohorts contain up to eight. Sequence-model microbatching reuses these cached outputs and does not recompute the structural encoder with different cohorts. The mapping $f_\theta(S)$ in Eq. 1 assumes this fixed structural preprocessing and cohort assignment. Reproducing the cached predictions requires preserving chain order and cohort boundaries, as well as preprocessing settings and checkpoint weights; regrouping chains can change the structural outputs.

Structural optimization uses AdamW with learning rate 2×10^{-4}, weight decay 0.01, batch size 16, and gradient clipping at 1.0. The run has a maximum step index of 200,000; the selected checkpoint is at step 63,000. Selection uses all-mask recovery on the configured 20 validation batches, evaluated every 1,000 updates. This checkpoint is frozen throughout sequence-model adaptation.

A.3 Sequence-model implementation

With the selected structural checkpoint frozen, Stage 2 learns to refine its complete predicted draft using the cached continuous structural states. Sequence-adaptation checkpoints are selected on the 256-chain development set held out from the training partition, using residue-pooled AAR and lower NLL to break ties. Table 6 records the sequence-adaptation settings used for the reported model. Full-residue cross-entropy is normalized by the total number of valid residues in each 16-chain training batch. Microbatches contain at most four chains, with a padded-length-squared budget of 500,000. ESM2 dropout and TF32 are disabled, and computation uses FP32. The complete instantiated model contains 661,767,634 parameters, including the frozen structural module and pretrained language-model parameters.

Table 6: Sequence-model adaptation settings.

Setting	**Value**
ESM2 depth / width	33 / 1,280
Injection after layer	16
Interface width / heads	128 / 4
ESM2 frozen warmup	125 updates
ESM2 learning rate	10^{-5}
Interface/head learning rate	10^{-4}
Optimizer / weight decay	AdamW / 0.01
Gradient clipping	1.0
Training batch size	16 chains
Total updates / selected step (seed 42)	4,500 / 2,000
Development evaluation interval	250 updates
Adaptation seeds	42, 43, 44

A.4 Metric aggregation and uncertainty

After checkpoint selection, standalone models and controlled variants share the following evaluation rules. External-generator combinations additionally select their fusion parameters on Validation-608, as detailed below; the test set is reserved for evaluation after these selections. Full-test sequence metrics use the same 1,120 chains and 180,387 valid residues. AAR pools correct predictions over residues, and PPL exponentiates summed native-residue negative log-probability divided by

the number of residues. Paired confidence intervals below resample complete protein chains with replacement and quantify evaluation-cohort variation for the fixed trained checkpoints.

A.5 External-generator integration protocol

To support the compatibility experiments in Section 4.4, we specify the external predictions, their combination with complete HiGenIF, and the selection of the displayed sequence examples.

External predictions. For DPLM-2 Bit, we use the official 650M checkpoint for backbone-conditioned sequence prediction. The Bit variant uses bit-level modeling of quantized structural features while retaining amino-acid sequence prediction (Hsieh et al., 2025). Our integration extracts first-step amino-acid logits and a generated-sequence vote using a one-step decoding configuration. For ADFLIP, all amino-acid identities are masked and only N/CA/C/O coordinates are supplied at flow time $t = 0$. Fusion rules are selected on validation data and locked before evaluating the canonical test set. Both pipelines reproduce the standalone external control on the same 1,120 chains and combine it with logits exported from the current HiGenIF checkpoint, which uses the complete predicted sequence draft. Native amino-acid labels are used only as evaluation targets.

Fusion and calibration. Let $p_i^H = \mathrm{softmax}(Z_i^{\mathrm{final}})$ denote the complete HiGenIF distribution at residue i. For external model $E \in \{D, F\}$, denoting DPLM-2 Bit and ADFLIP respectively, let z_i^E be its amino-acid logits and $p_i^E(T) = \mathrm{softmax}(z_i^E/T)$. The fused distributions are

$$\begin{aligned} p_i^{H+D} &= \mathrm{softmax}\big(\log p_i^H + \alpha \log p_i^D(T_D) + \beta v_i\big)\,, \\ p_i^{H+F} &= \mathrm{softmax}\big((1-w)\log p_i^H + w \log p_i^F(T_F)\big)\,, \end{aligned} \tag{15}$$

where logarithms act element-wise and v_i is the 20-dimensional one-hot vote for the amino acid in DPLM-2 Bit's deterministic generated sequence. The selected parameters are $(T_D, \alpha, \beta) = (1.25, 1.0, 0.1)$ and $(T_F, w) = (1.0, 0.4)$. Here, T_F is a calibration temperature; ADFLIP's flow time remains $t = 0$. Final softmax normalization is applied before computing PPL, and position-wise argmax gives the fused sequence.

Standalone external controls use a separately selected temperature minimizing validation PPL. Fusion parameters instead maximize validation residue-pooled AAR, with ties broken by lower NLL. For DPLM-2 Bit, the standalone temperature grid is $\{0.75, 1, 1.25, 1.4, 1.5, 2\}$; the fusion grid uses $T_D \in \{0.75, 1, 1.25, 1.5\}$ and $\alpha, \beta \in \{0, 0.1, 0.2, 0.3, 0.5, 0.75, 1\}$. For ADFLIP, both temperature searches use $\{0.5, 0.75, 0.9, 1, 1.25, 1.5, 2\}$, and fusion weights use $w \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.75, 0.9\}$. All searches use Validation-608; selected temperatures and weights are fixed for test evaluation.

Qualitative sequence selection. To illustrate how the combined predictions differ at individual residues, we select sequence windows from the completed evaluation. Sequence-example selection follows the locked aggregate test evaluation and is independent of checkpoint, temperature, and fusion-coefficient selection. Panels A–B use chains of length 45–75 with standalone HiGenIF AAR of at least 60%, an advantage of at least 10 percentage points over standalone DPLM-2 Bit, and a fused prediction that exceeds both standalone components. We rank these candidates by standalone HiGenIF AAR, then fusion AAR and chain identifier. Panel C is selected among 45–65-residue chains whose fused prediction exceeds both standalone components, ranked by the fusion margin over the stronger component, then fusion AAR and chain identifier. Within each selected chain, we display the contiguous 48-residue window containing the most positions where HiGenIF is correct and DPLM-2 Bit is incorrect; ties favor more fusion-correct positions and then the earlier window. Table 4 shows these windows. The visualization illustrates both strong standalone structural predictions and complementary residue-level corrections, while aggregate evaluation uses all 1,120 test chains; exact full sequences remain in the accompanying evidence record.

A.6 Adaptation-seed stability

We next examine the repeatability of the sequence-adaptation stage used by the complete model. With the hierarchical structural checkpoint fixed, we repeat ESM2-650M adaptation using seeds 42, 43, and 44. All three sequence-refinement runs use the predicted draft, 4,500 updates, and

development-based checkpoint selection. Seed 43 selects update 1,250; seeds 42 and 44 select update 2,000. The resulting AAR is 59.3112 ± 0.0713, showing that sequence-refinement adaptation is stable across these seeds. This design isolates seed sensitivity in sequence-refinement adaptation, with all checkpoint selection performed on development data.

Table 7: ESM2 adaptation-seed stability with one frozen structural checkpoint. The final row reports mean and sample standard deviation.

Seed	**PPL** $\downarrow$	**AAR (%)** $\uparrow$
42	3.3997	59.3740
43	3.4062	59.2338
44	3.4065	59.3258
Mean $\pm$ SD	3.4041 ± 0.0038	59.3112 ± 0.0713

A.7 Continuous structural-memory analysis

The refinement stage receives both a discrete sequence draft and continuous structural states. To measure the additional contribution of the continuous input while preserving the draft, we compare the complete model with a no-injection control. The no-injection control is retrained using the complete predicted sequence draft. It keeps the frozen hierarchical predictor, predicted draft, residual structural logits, ESM2-650M initialization, optimizer, 4,500-update budget, and development-selection rule fixed, while removing only the layer-16 structural-state injection. Table 8 reports prespecified strata and 20,000 paired chain-bootstrap replicates. A cross-segment contact denotes a residue with any residue from another DSSP segment within 8 Å and sequence separation greater than two.

Table 8: Complete HiGenIF minus the no-injection control. Positive ΔAAR and negative ΔPPL favor continuous structural-state injection.

Evaluation stratum	**ΔAAR (pp)** **95% CI**	**ΔPPL** **95% CI**
All valid residues	$+0.1530$ $[0.0667, 0.2364]$	-0.0201 $[-0.0240, -0.0162]$
Cross-segment contact	$+0.1669$ $[0.0798, 0.2538]$	-0.0198 $[-0.0237, -0.0159]$
Structural draft incorrect	$+0.3007$ $[0.1568, 0.4451]$	-0.0815 $[-0.0993, -0.0630]$

We then examine whether the residue-level changes also appear when recovery is summarized over structural regions. We define a segment as a maximal contiguous run with one DSSP label and macro-average recovery over segments. The region-level confidence intervals use 5,000 paired chain-bootstrap replicates; the residue-level table above uses 20,000. Continuous-state injection improves macro segment AAR by 0.2245 points on the test set (95% CI $[0.1168, 0.3310]$) and 0.2540 points on the validation set (95% CI $[0.1044, 0.3999]$). Exact segment recovery on the test set increases by 0.2094 points (95% CI $[0.0786, 0.3423]$).

A.8 Regional-partition control

The preceding control concerns the interface to ESM2. Here we specify the complementary control of the structural predictor itself: whether explicit regional organization improves prediction when both variants receive the same sequence-refinement procedure. The regional comparison in Table 5 uses matched 30,000-update structural training, followed by 4,500 updates of ESM2-650M adaptation with the complete predicted sequence draft and development-based checkpoint selection. The regional variant uses DSSP-defined groups with within- and between-group message passing. The control assigns all residues in each chain to one group; it retains the global and residue pathways, the six-way coarse classifier and its conditioning of amino-acid prediction. Both variants use seed 42,

receive no native amino-acid identities at inference, and are evaluated on the locked 1,120-chain, 180,387-residue CATH test set after model selection.

A.9 Reference sources

The main tables combine HiGenIF evaluations with published reference results. We record the configurations and sources here so that each reported value can be traced to its evaluation setting. Literature scores retain their source evaluation protocols.

CATH 4.2 sources. ProteinMPNN values follow the reproduced evaluation in Table 1 of Gao et al. (Gao et al., 2022); Dauparas et al. (Dauparas et al., 2022) is the reference for the method. LM-Design uses the PiFold-based configuration reported in Table 1 of Zheng et al. (Zheng et al., 2023). FAMPNN results follow the single-step, 0.0 Å configuration in Table 1 of Widatalla et al. (Widatalla et al., 2025), which reports median chain recovery of 49.66% and PPL of 4.99; Short and Single-chain results are unreported. ScFold results follow Table 1 of Zhong et al. (Zhong et al., 2025). ProtSeqGen values are the means in Table 5 of Gao et al. (Gao et al., 2026): $53.03 \pm 0.07\%$ AAR and 4.43 ± 0.03 PPL.

TS50 and TS500 sources. ProteinMPNN values follow the reproduced evaluation in Table 5 of Gao et al. (Gao et al., 2022). Consistent with the CATH comparison, LM-Design uses the PiFold-based configuration trained on CATH 4.2 in Table 3 of Zheng et al. (Zheng et al., 2023): 3.50 / 57.89% on TS50 and 3.19 / 67.78% on TS500 (PPL/AAR). ScFold results follow Table 3 and the reported aggregate TS500 recovery in Zhong et al. (Zhong et al., 2025); aggregate TS500 PPL is unreported. InstructPLM-DPO values follow Table 6 of Xu et al. (Xu et al., 2025). ProtSeqGen values follow Tables 6 and 7 of Gao et al. (Gao et al., 2026): $58.51 \pm 0.15\%$/3.79 ± 0.05 on TS50 and $60.59 \pm 0.11\%$/3.37 ± 0.04 on TS500 (AAR/PPL).