RETRIEVED TEMPLATES AS REVISABLE PRIORS FOR PROTEIN INVERSE FOLDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Structural retrieval can provide strong target-specific evidence for protein inverse folding, yet retrieved sequences are neither uniformly reliable nor necessarily compatible with the target backbone. We investigate two questions: which retrieved residues should be trusted, and how should this uncertain evidence enter a generative model? We introduce a discrete generative method that treats aligned templates as a revisable sequence prior. The method combines global structural similarity, local geometric consistency, and inter-template agreement to estimate residue-level reliability. These estimates define a partially masked source distribution, which a time-conditioned model iteratively refines while remaining conditioned on the target structure and aligned templates. We further train on model-generated intermediate states to match the states encountered during refinement. Our method achieves sequence recovery of 66.65%, 73.11%, and 73.36% on CATH 4.2, TS50, and TS500, respectively. It also improves over direct template transfer by 13.11 percentage points on CATH 4.2. ESMFold evaluation of the top five among 20 generated candidates yields mean scTM scores of 0.877, 0.907, and 0.897 on the three benchmarks. These results suggest that retrieval is useful when treated as informative but editable evidence.

1 INTRODUCTION

Protein inverse folding asks for amino acid sequences that are compatible with a prescribed three-dimensional backbone. It is a central component of computational protein design, where accurate sequence proposals can reduce the experimental search space for expression, stability, and function (Ingraham et al., 2019; Dauparas et al., 2022). Geometric neural networks, equivariant representations, and generative models have substantially improved sequence prediction from backbone coordinates (Jing et al., 2021; Hsu et al., 2022; Dauparas et al., 2022; Gao et al., 2022; Yi et al., 2023). Yet the problem remains underdetermined: many sequences may fit the same backbone, and a backbone-only model must infer target-specific residue preferences from patterns stored in its parameters. Growing structure databases offer a complementary source of evidence. Foldseek and TM-align can retrieve proteins with similar folds and align their residues to a target backbone (Van Kempen et al., 2024; Zhang & Skolnick, 2005). The aligned sequences often contain useful target-specific cues, and recent work has shown that they can improve inverse folding when supplied as conditioning features (Liao et al., 2026). However, a retrieved sequence is not the target sequence. Even a globally similar template can contain locally mismatched regions, leaving a basic question unresolved: how should a model use evidence that is informative but unevenly reliable?

We consider two aspects of this problem. The first is where retrieval should be trusted. A global TM-score is effective for selecting templates but is constant across all residues in an alignment; it cannot distinguish a conserved local region from a mismatched one. We hypothesize that local position and orientation, together with agreement across templates, provide the residue-level resolution needed to decide what should be retained or revised. The second aspect is how retrieval should enter generation. Template conditioning determines what evidence a predictor can consult, whereas source initialization determines the sequence state from which generation begins. These roles are distinct, but most retrieval-based methods focus on the former, while discrete generative models commonly start from a fully masked state (Campbell et al., 2024; Gat et al., 2024). To study these two aspects jointly, we introduce a reliability-aware, retrieval-guided discrete generative model. For each aligned template residue, we combine global TM-score with local C_α displacement and local-frame

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.

discrepancy. Evidence from multiple templates is then aggregated according to its reliability and agreement. The resulting categorical amino acid prior is mixed with the mask state, producing an informative but non-deterministic source distribution. A time-conditioned model refines this source while remaining conditioned on the target structure and aligned templates. Every retrieved residue remains editable. Because iterative inference exposes the model to states created by its own predictions, we train on intermediate states sampled from the same refinement process.

Our residue-level analysis supports the first design choice. Within a protein, the template-level TM-score is constant and therefore has an AUROC of 0.500 for distinguishing correct from incorrect template-prior residues. Our local confidence reaches a mean AUROC of 0.711 and AUPRC of 0.610 (Sec. 4.4). The analysis shows that global similarity identifies useful templates, whereas local geometry identifies the useful regions within them. Controlled ablations address the second design choice. Retrieval-derived initialization alone improves CATH 4.2 recovery from 37.80% to 51.34%, while template conditioning alone reaches 59.61%. Combining both in the full model yields 66.26% with one-step prediction and 66.65% with five-step refinement. The full model also exceeds direct top-1 template transfer by 13.11 percentage points, indicating that it learns corrections beyond copying aligned residues.Across CATH 4.2, TS50, and TS500, our method achieves sequence recovery of 66.65%, 73.11%, and 73.36%, respectively. On ESMFold refolding, the top five among 20 generated candidates reach mean scTM scores of 0.877, 0.907, and 0.897. Together, these results motivate a simple view of retrieval for inverse folding: retrieved sequences should serve neither as fixed answers nor merely as auxiliary features, but as uncertain evidence that the model can preserve, revise, or override.

2 Related Work

2.1 Geometric inverse folding.

Modern inverse-folding methods learn sequence distributions conditioned on protein backbone geometry. Graph-based and equivariant architectures include StructGNN, GVP-GNN, ESM-IF1, ProteinMPNN, and PiFold (Ingraham et al., 2019; Jing et al., 2021; Hsu et al., 2022; Dauparas et al., 2022; Gao et al., 2022; 2024b; Ren et al., 2024). Protein language models have also been adapted to structure-conditioned design, as in LM-Design and KW-Design (Zheng et al., 2023; Gao et al., 2024a). These methods infer a sequence primarily from the target backbone and knowledge in the model parameters. Our work instead asks how residue-aligned evidence from external structure databases can be incorporated into this prediction.

2.2 Structural retrieval for inverse folding.

Foldseek and TM-align provide efficient structure search and residue-level alignment (Van Kempen et al., 2024; Zhang & Skolnick, 2005). DualMPNN processes an aligned template with a dual-stream, alignment-aware network (Liao et al., 2026). PRISM integrates fine-grained structure–sequence representations into a retrieval-aware decoder (Mahbub et al., 2026). These approaches establish retrieval as a useful source of conditioning information. We additionally use retrieval to define the initial generative state: aligned residues are weighted by local reliability, converted into a categorical prior, and left revisable throughout generation. This formulation lets us separate template conditioning from retrieval-based initialization.

2.3 Discrete generative modeling.

Discrete diffusion and flow models generate categorical data through iterative transitions between source and target distributions (Austin et al., 2021; Campbell et al., 2024; Gat et al., 2024). GraDe-IF and MapDiff apply discrete denoising to inverse folding (Yi et al., 2023; Bai et al., 2025), while ADFLIP formulates all-atom inverse folding as discrete flow matching from a generic masked source (Yi et al., 2025). Our source is instead a reliability-calibrated retrieval prior. We also train on intermediate states produced by the deployed refinement process so that the model learns to correct the kinds of states it encounters at inference. This idea is related to scheduled sampling for exposure-bias mitigation (Bengio et al., 2015), but the states here arise from the discrete refinement trajectory rather than an autoregressive prefix.

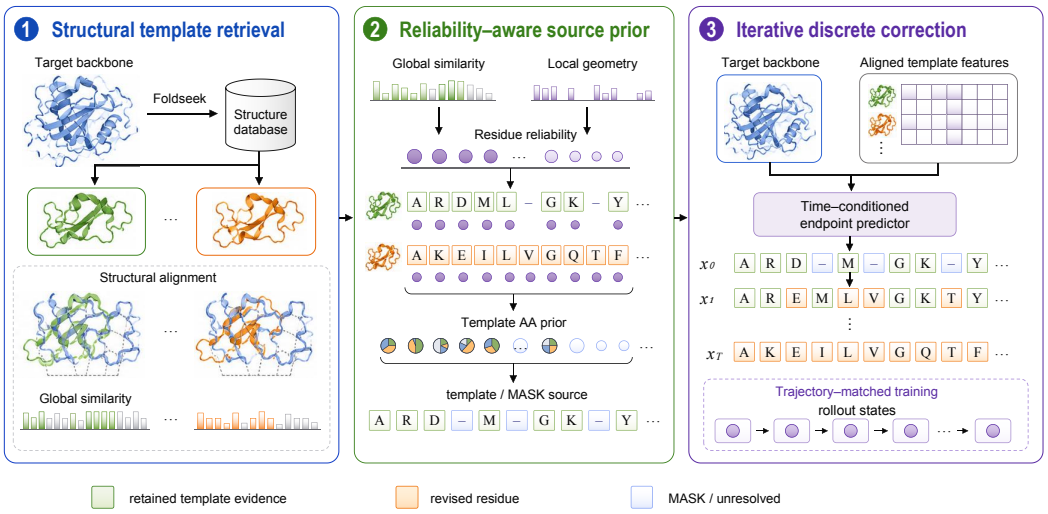


Figure 1: **Overview of the proposed method.** **(1)** Structurally similar templates are retrieved and aligned to the target backbone. **(2)** Global similarity, local geometric consistency, and inter-template agreement are combined to construct residue-wise reliability scores and a partially masked, revisable source distribution. **(3)** A target-conditioned discrete model iteratively refines the source, and trajectory-matched training exposes it to model-generated intermediate states.

3 METHOD

3.1 PROBLEM FORMULATION AND OVERVIEW

Given a target protein backbone of length L, $G = \{\mathbf{X}_i\}_{i=1}^{L}$, where $\mathbf{X}_i \in \mathbb{R}^{4\times 3}$ contains the coordinates of the N, C_α, C, and O atoms of residue i, the goal is to generate $\mathbf{x}^* \in \mathcal{A}^L$, where $\mathcal{A}$ is the alphabet of 20 standard amino acids. The generated sequence should be compatible with G.

Our framework is shown in Fig. 1. Retrieved templates have two distinct roles. Their structures and sequences provide persistent conditioning features, and their aligned amino acids define the initial distribution for generation. We calibrate this initial distribution at the residue level and refine it with a shared time-conditioned predictor. No template-derived residue is fixed: every position can be revised during generation.

3.2 STRUCTURAL RETRIEVAL AND RESIDUE-LEVEL RELIABILITY

For a target backbone G, we use Foldseek to retrieve structurally similar candidates and TM-align to establish residue correspondences between each template and the target (Van Kempen et al., 2024; Zhang & Skolnick, 2005). After applying the leakage-control filters described in Section 4.1, we retain the top K templates and denote them by $\mathcal{T} = \{T_k\}_{k=1}^{K}$. Let $q_k \in [0,1]$ be the target-normalized TM-score between the target and template T_k. The score q_k measures the overall agreement of two folds, but assigns the same value to every position in their alignment. We therefore complement it with two local measurements. For target residue i and the corresponding residue in template k, the aligned C_α displacement is

$$\Delta_{ki}^{\mathrm{CA}} = \left\| \widetilde{\mathbf{x}}_{ki}^{\mathrm{CA}} - \mathbf{x}_i^{\mathrm{CA}} \right\|_2, \tag{1}$$

where $\widetilde{\mathbf{x}}_{ki}^{\mathrm{CA}}$ is the template coordinate after structural superposition. Position alone does not capture local orientation, so we also construct unit vectors from C_α to the N and C atoms:

$$\widehat{\mathbf{v}}_i^{N} = \frac{\mathbf{x}_i^{N} - \mathbf{x}_i^{\mathrm{CA}}}{\|\mathbf{x}_i^{N} - \mathbf{x}_i^{\mathrm{CA}}\|_2}, \qquad \widehat{\mathbf{v}}_i^{C} = \frac{\mathbf{x}_i^{C} - \mathbf{x}_i^{\mathrm{CA}}}{\|\mathbf{x}_i^{C} - \mathbf{x}_i^{\mathrm{CA}}\|_2}. \tag{2}$$

The local backbone-frame discrepancy is then defined as

$$\Delta_{ki}^{\text{frame}} = \arccos\left[\frac{\widehat{\mathbf{v}}_i^N \cdot \widehat{\mathbf{v}}_{ki}^N + \widehat{\mathbf{v}}_i^C \cdot \widehat{\mathbf{v}}_{ki}^C}{2}\right]. \tag{3}$$

The residue-level reliability is the global score modulated by kernels over the two local discrepancies:

$$r_{ki} = q_k \exp\left[-\frac{1}{2}\left(\frac{\Delta_{ki}^{\text{CA}}}{\sigma_{\text{CA}}}\right)^2 - \frac{1}{2}\left(\frac{\Delta_{ki}^{\text{frame}}}{\sigma_{\text{frame}}}\right)^2\right]. \tag{4}$$

The bandwidths σ_{CA} and σ_{frame} control sensitivity to position and orientation, respectively; their values are given in Section 4.1. We set $r_{ki} = 0$ when no valid correspondence exists. Thus, a locally inconsistent residue can receive low reliability even when its template has high global similarity.

3.3 Revisable Retrieval-Derived Source Distribution

Let $a_{ki} \in \mathcal{A}$ denote the amino acid in template k aligned to target position i. We aggregate amino acid evidence from different templates according to their residue-level reliability:

$$p_i^T(a) = \frac{\sum_{k=1}^K r_{ki}\mathbb{I}[a_{ki} = a]}{\sum_{k=1}^K r_{ki}}, \qquad a \in \mathcal{A}. \tag{5}$$

For an uncovered position, p_i^T is unused and its retrieval confidence is zero. For a covered position, we summarize two complementary properties: reliable coverage u_i and inter-template agreement g_i,

$$u_i = 1 - \prod_{k=1}^K (1 - r_{ki}), \qquad g_i = 1 - \frac{H(p_i^T)}{\log|\mathcal{A}|}, \qquad c_i = u_i g_i. \tag{6}$$

The coverage term is high when at least one structurally reliable template supports the position. The agreement term is high when the aligned templates concentrate their mass on the same amino acid. Their product c_i is therefore high only when both structural support and sequence agreement are strong.

Let M denote the mask token. We mix the template prior with the mask state to define a residue-wise source distribution:

$$\rho_i = \min(\rho_{\max}, c_i), \qquad p_{0,i}(a) = \rho_i p_i^T(a) + (1 - \rho_i)\delta_M(a), \tag{7}$$

The cap $\rho_{\max} < 1$ prevents retrieval from fully determining any position; its value is given in Section 4.1. We then sample the initial sequence independently by position:

$$x_{0,i} \sim p_{0,i}. \tag{8}$$

All positions remain uncommitted after initialization. A sampled template residue is therefore evidence for generation, not a fixed prediction.

3.4 Conditional Endpoint Predictor

A shared time-conditioned model predicts the endpoint amino acid distribution from the current sequence state:

$$p_\theta\left(\mathbf{x}^* \mid \mathbf{x}_t, G, \mathcal{T}, t\right). \tag{9}$$

The model encodes the target backbone with atom- and residue-level geometric networks, then adds embeddings of the current sequence state and time t. For each template, a separate encoder jointly

represents backbone geometry and amino acid identity. Structural alignments map these template features to the target positions. We compute their weighted mean and disagreement using $q_k r_{ki}$ as the alignment weight, and inject these summaries and c_i into the target representation through gated fusion. This persistent template conditioning supplies information beyond the retrieval-derived source state.

To provide explicit information about residue exposure and local molecular surface geometry, we construct a static surface with MSMS (Sanner et al., 1996) from the target-backbone N, C_α, C, and O atoms and sequence-independent pseudo-C_β atoms. Surface vertices are represented using curvature and heat-kernel signatures (Sun et al., 2009); molecular-surface representations have also proved useful for geometric learning on proteins (Gainza et al., 2020). Bidirectional message passing connects surface vertices and residue representations. The surface uses neither native side-chain coordinates nor residue identities, does not enter the construction of p_i^T or c_i, and remains fixed across the controlled retrieval ablations.

3.5 Reliability-Aware Multi-Step Discrete Refinement

We use an endpoint-parameterized discrete flow (Campbell et al., 2024; Gat et al., 2024). Generation is divided into T stages with $t_j = j/T$ for $j = 0, \ldots, T$, and all stages share the same endpoint predictor. The update schedule should revise uncertain positions while allowing reliable ones to commit early. We encode this preference in a residue-wise mutability score. Let

$$h_i = \frac{H(p_i^T)}{\log |\mathcal{A}|}, \qquad e_i = \frac{1}{2}\left[1 - \tanh(b_i^S)\right], \tag{10}$$

where h_i is the normalized entropy of the template prior and b_i^S is a centered burial bias derived from the residue-to-surface distance. The term e_i is therefore a normalized measure of surface exposure. We average prior uncertainty, exposure, and lack of template confidence:

$$m_i = \frac{1}{3}\left[h_i + e_i + (1 - c_i)\right]. \tag{11}$$

Template confidence is converted into a path bias according to

$$b_i^T = \mathrm{clip}\left(\mathrm{logit}(c_i), -B_T, B_T\right), \tag{12}$$

with $b_i^T = 0$ for positions without template coverage. Let $\mathcal{C}_B(\cdot)$ denote within-protein centering followed by clipping to $[-B, B]$. The final residue-specific path bias is

$$\bar{b}_i = \mathcal{C}_B(b_i^T + b_i^S), \qquad b_i = \mathcal{C}_B\left[\bar{b}_i + s_m(0.5 - m_i)\right]. \tag{13}$$

High-confidence, low-mutability, and more deeply buried positions receive larger path biases and tend to commit earlier. The cumulative commitment progress of residue i at time t is

$$\kappa_i(t) = \frac{t \exp(b_i)}{1 - t + t \exp(b_i)}. \tag{14}$$

For stochastic generation, each position draws a fixed threshold $z_i \sim \mathrm{Uniform}(0, 1)$. At stage j, the model proposes an amino acid for every position. An uncommitted position accepts its proposal when $\kappa_i(t_{j+1}) \geq z_i$; otherwise, it retains its current state. Because $\kappa_i(1) = 1$, every position is eventually updated. Deterministic decoding uses a fixed threshold and the most probable proposal. After the T update stages, one endpoint readout at $t = 1$ produces the final sequence. Section 4.1 specifies the threshold and number of stages.

For stochastic multi-sequence generation, a control parameter $\lambda \in [0, 1]$ adjusts the exploration strength through a residue-specific temperature,

$$\tau_i(\lambda) = \tau_{\min} + \lambda m_i(\tau_{\max} - \tau_{\min}). \tag{15}$$

Section 4.1 gives the evaluation-specific value of λ and the corresponding deterministic or stochastic protocol.

3.6 TRAJECTORY-MATCHED TRAINING

Artificially corrupted native sequences do not reproduce the intermediate states created by the model during multi-step inference. We instead sample training states from the deployed refinement process. For each example, we draw $\mathbf{x}_0$ from p_0, sample a stage $s \in \{0, \dots, T\}$, and roll out the current model to that stage without gradient computation. The exploration control is sampled from the range given in Section 4.1. A single gradient-bearing endpoint prediction is then made from the resulting state $\mathbf{x}_{t_s}^{\text{roll}}$; gradients do not pass through the rollout.

The target is centered on the native sequence but retains a small amount of template uncertainty at mutable positions:

$$y_i(a) = (1 - \alpha_i)\mathbb{I}[x_i^* = a] + \alpha_i p_i^T(a), \qquad \alpha_i = \eta m_i, \tag{16}$$

We set $\alpha_i = 0$ without template evidence and apply uniform label smoothing with coefficient ϵ_{ls}:

$$\bar{y}_i(a) = (1 - \epsilon_{\text{ls}})y_i(a) + \frac{\epsilon_{\text{ls}}}{|\mathcal{A}|}. \tag{17}$$

To emphasize positions that require correction along the deployed trajectory, we increase the loss weight for both incorrect rollout states and incorrect template-derived initializations:

$$w_i = \left[1 + \omega_{\text{src}}\mathbb{I}(x_{0,i} \neq M,\ x_{0,i} \neq x_i^*)\right]\left[1 + (\gamma - 1)\mathbb{I}\left(x_{t_s,i}^{\text{roll}} \neq x_i^*\right)\right], \tag{18}$$

Here, ω_{src} upweights incorrect residues already present in the retrieval-derived source, whereas γ upweights incorrect residues in the rollout state. Their values are specified in Section 4.1. The final training objective is

$$\mathcal{L} = -\frac{\sum_{i=1}^{L} w_i \sum_{a \in \mathcal{A}} \bar{y}_i(a) \log p_\theta\left(a \mid \mathbf{x}_{t_s}^{\text{roll}}, G, \mathcal{T}, t_s\right)}{\sum_{i=1}^{L} w_i}. \tag{19}$$

We also drop templates at random and corrupt a small fraction of template tokens to discourage direct copying. At inference, the same reliability calibration constructs the source, followed by five update stages and a final endpoint readout. Training and inference thus use the same transition mechanism, so the model learns to preserve or revise retrieval evidence along the trajectory it will actually follow.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets and splits. We evaluate on CATH 4.2, TS50, and TS500. The CATH 4.2 training, validation, and test splits contain 18,024, 608, and 1,120 protein chains, respectively. TS50 and TS500 contain 50 and 500 targets and test generalization to external datasets. Every model variant receives only the N, C_α, C, and O coordinates of the target backbone.

Structural template retrieval. We follow the DualMPNN retrieval protocol and use its official target–template pairs. Foldseek retrieves candidate structures, and TM-align provides residue correspondences and a TM-score normalized by target length. We remove candidates from the same PDB entry as the target and those with TM-score above 0.99. From the remaining candidates, with no additional lower threshold, we retain the top $K = 2$. All controlled comparisons use the same template pairs and alignments.

Architecture and calibration. The time-conditioned sequence model uses a hidden dimension of 128, three target-backbone blocks, three template-encoding layers, and 32 structural neighbors per residue. The complete model has 5.62M trainable parameters. Further implementation and hyperparameter details are provided in Section F of the supplementary material.

Table 1: Perplexity (PPL) and sequence recovery on CATH 4.2, TS50, and TS500.

	CATH 4.2		TS50		TS500	
Method	PPL ↓	Recovery (%) ↑	PPL ↓	Recovery (%) ↑	PPL ↓	Recovery (%) ↑
GraphTrans	6.63	35.82	5.60	42.20	5.16	44.66
GVP	5.36	39.47	4.71	44.14	4.20	49.14
AlphaDesign	6.30	41.31	5.25	48.36	4.93	49.23
ProteinMPNN	4.57	49.87	3.93	54.43	3.53	58.08
PiFold	4.55	51.66	3.86	58.72	3.44	60.42
GraDe-IF	4.35	52.21	3.71	56.32	3.23	61.22
VFN-IF	4.17	54.74	3.58	59.54	3.19	63.65
LM-Design	4.52	55.65	3.50	57.89	3.19	67.78
DualMPNN	3.18	65.51	2.76	70.99	2.71	70.37
PRISM	**2.71**	60.43	**2.43**	67.92	**2.27**	70.53
Full method	3.03	**66.65**	2.48	**73.11**	2.46	**73.36**

Training. Training has two stages. We first train a template-conditioned endpoint model with learning rate 10^{-3} and select the initialization checkpoint by validation loss. We then run $20{,}000$ trajectory-matched updates and select the final model by validation recovery. In the second stage, the generation and template modules use learning rate 2×10^{-5}, and the remaining encoder parameters use 2×10^{-6}. We use AdamW with weight decay 0.01, dropout 0.1, gradient clipping at 1.0, and an exponential moving average with decay 0.999. The schedule consists of 500 linear warm-up steps followed by cosine decay. Training uses four NVIDIA RTX 3090 GPUs, a global batch size of 16, and two-step gradient accumulation, giving an effective batch size of 32.

Inference protocol. Generation starts from the reliability-calibrated source, performs five discrete updates, and returns the terminal prediction. Sequence-recovery experiments use deterministic decoding with $\lambda = 0$. Refolding experiments use $\lambda = 0.5$ and generate 20 sequences per target.

Evaluation metrics. Our primary metrics are sequence recovery and perplexity (PPL). For paired analyses, we also retain recovery for each test protein. To assess structural compatibility, we refold generated sequences with ESMFold (Lin et al., 2023) and report target-normalized scTM, TM-align RMSD, and the fractions above scTM thresholds of 0.5 and 0.8.

4.2 Overall Inverse-Folding Performance

Table 1 compares perplexity and sequence recovery on the three benchmarks. Our method reaches 66.65%, 73.11%, and 73.36% recovery on CATH 4.2, TS50, and TS500, respectively. These results improve over DualMPNN, the strongest listed retrieval-based baseline in recovery, by 1.14, 2.12, and 2.99 percentage points. The metrics do not rank all methods identically: PRISM has the lowest PPL on each dataset, while our method has the highest recovery. We therefore treat the result as an improvement in native-sequence agreement rather than a uniform improvement across metrics.

4.3 Ablation Studies

We next isolate how retrieval enters the model and whether the resulting predictions go beyond direct template transfer. All ablations use CATH 4.2.

How should retrieval be integrated into generation? Table 2 separates retrieval-derived source initialization from feature-level template conditioning. The former determines where generation starts; the latter determines what aligned evidence the predictor can consult throughout generation. With no template conditioning, replacing a masked source with the retrieval-derived source raises recovery from 37.80% to 51.34%. With a masked source, template conditioning reaches 59.61% and is therefore the stronger pathway in isolation. Enabling both yields 66.26% with one-step prediction, consistent with complementary roles for an informed initial state and persistent aligned features. Five-step refinement increases recovery from 66.26% to 66.65% while leaving PPL at 3.03. The modest gain shows that the endpoint predictor already accounts for most deterministic recovery; iterative inference mainly provides a structured path on which the initialized residues remain revisable.

Table 2: Effects of retrieval initialization, template conditioning, and inference steps on CATH 4.2.

Source initialization	Template conditioning	Steps T	PPL ↓	Recovery (%) ↑
Single-path baselines				
MASK	No	1	7.58	37.80
Retrieval	No	1	5.79	51.34
MASK	Yes	1	4.27	59.61
Full method				
Retrieval	Yes	1	3.03	66.26
Retrieval	**Yes**	**5**	**3.03**	**66.65**

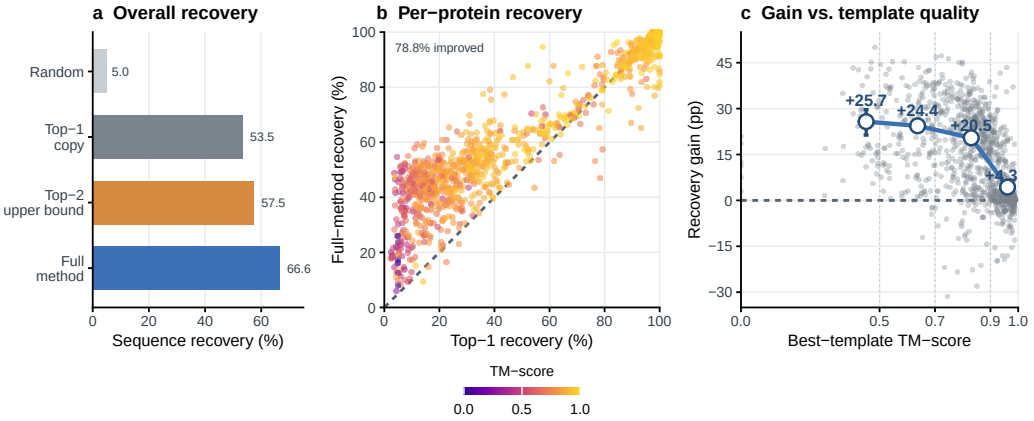


Figure 2: **Direct template transfer versus learned revision on CATH 4.2. (a)** Overall recovery of direct-transfer baselines and the full method. **(b)** Per-protein recovery of the full method versus top-1 transfer. **(c)** Recovery gain over top-1 transfer as a function of best-template quality.

Does the model merely copy retrieved templates? Figure 2 compares learned generation with direct transfer. Random sequences recover 5.00% of native residues. Copying the top-1 template reaches 53.54%, and voting between the top two templates reaches 53.76%. Even a non-deployable upper bound that uses native labels to select a correct residue whenever either template contains one reaches only 57.50%. The full model obtains 66.65%, exceeding top-1 transfer by 13.11 points and the label-assisted upper bound by 9.15 points (Fig. 2a). The improvement is also consistent across targets. The model outperforms top-1 transfer on 882 of $1{,}120$ proteins (78.8%; Fig. 2b). When targets are grouped by best-template TM-score into < 0.5, 0.5–0.7, 0.7–0.9, and ≥ 0.9, the residue-weighted gains are 25.74, 24.38, 20.46, and 4.34 points, respectively (Fig. 2c). The method therefore makes its largest corrections when templates are imperfect, yet retains a positive gain when template quality is high.

4.4 Residue-Level Reliability Beyond Global Similarity

Global TM-score is constant across an alignment and cannot identify which template residues should be trusted. Across all CATH 4.2 test proteins, combining aligned C_α displacement and local-frame consistency raises within-alignment AUROC from 0.500 to 0.653. After aggregating templates, the final residue confidence reaches mean within-protein AUROC 0.711 and AUPRC 0.610, compared with 0.500 and 0.491 for global TM-score. Kernel-width sensitivity and detailed diagnostics are reported in Sections A and B of the supplementary material. As shown in Table 3, unweighted template evidence obtains 60.23% recovery, global TM-score weighting reaches 60.51%, and adding local geometry reaches 61.07%. Global quality and local geometry therefore make distinct contributions to the learned generator.

Table 3: End-to-end reliability ablation on CATH 4.2

Reliability used	PPL ↓	Recovery (%) ↑
Unweighted	3.86	60.23
Global TM-score	3.13	64.98
Global + local geometry	3.09	66.31
Full reliability	**3.03**	**66.65**

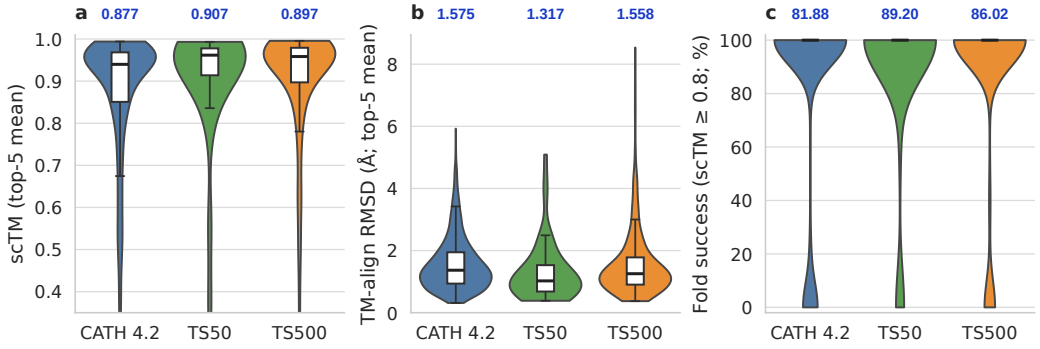


Figure 3: **ESMFold refolding evaluation.** The panels show mean scTM, TM-align RMSD, and the proportion satisfying $\mathrm{scTM} \geq 0.8$. Box plots indicate the median and interquartile range.

4.5 Structural Feasibility under Refolding

As shown in Figure 3, we generate 20 sequences per target at $\lambda = 0.5$, refold them with ESMFold, and retain the five candidates with highest scTM. On CATH 4.2, TS50, and TS500, mean scTM is 0.877, 0.907, and 0.897; respectively 81.88%, 89.20%, and 86.02% exceed scTM 0.8. Without top-5 selection, all 20 candidates still average 0.851, 0.894, and 0.881. Section E of the supplementary material reports the recovery–diversity trade-off.

5 Conclusion

We studied how uncertain retrieval evidence should enter protein inverse folding. Our method assigns aligned residues local reliability, uses the resulting prior to initialize discrete generation, and retains the templates as persistent conditioning during iterative refinement. This separation of initialization and conditioning improves recovery over either pathway alone and over direct transfer from the same templates. Across CATH 4.2, TS50, and TS500, the model reaches 66.65%, 73.11%, and 73.36% recovery, while ESMFold evaluation gives mean scTM between 0.877 and 0.907. The results support treating retrieved sequences as editable evidence rather than fixed answers.

AI Use Statement

In this work, we used generative AI tools to assist with translating, drafting, restructuring, and polishing manuscript text; improving readability and organization; formatting LaTeX and tables. Generative AI was not used to generate synthetic datasets or to formulate or prove mathematical claims. The authors determined the scientific questions, method, experimental design, and final conclusions. All AI-assisted text and artifacts were reviewed and revised by the authors. We verified the reported numerical results against the original experimental outputs and checked the citations against the source literature. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

References

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing*

systems, 34:17981–17993, 2021.

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.

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

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

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.

Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, Davide Boscaini, Michael M Bronstein, and Bruno E Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature methods*, 17(2):184–192, 2020.

Zhangyang Gao, Cheng Tan, Pablo Chacón, and Stan Z Li. Pifold: Toward effective and efficient protein inverse folding. *arXiv preprint arXiv:2209.12643*, 2022.

Zhangyang Gao, Cheng Tan, Xingran Chen, Yijie Zhang, Jun Xia, Siyuan Li, and Stan Z Li. Kw-design: Pushing the limit of protein design via knowledge refinement. In *International Conference on Learning Representations*, volume 2024, pp. 34556–34574, 2024a.

Zhangyang Gao, Jue Wang, Cheng Tan, Lirong Wu, Yufei Huang, Siyuan Li, Zhirui Ye, and Stan Z Li. Uniif: Unified molecule inverse folding. *Advances in Neural Information Processing Systems*, 37:135843–135860, 2024b.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

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

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

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael J. L. Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021.

Xuhui Liao, Zhiqiang Liang, Liwei Xiao, Junjie Chen, et al. Dualmpnn: Harnessing structural alignments for high-recovery inverse protein folding. *Advances in Neural Information Processing Systems*, 38:111623–111643, 2026.

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.

Sazan Mahbub, Souvik Kundu, and Eric P Xing. Prism: Enhancing protein inverse folding through fine-grained retrieval on structure-sequence multimodal representations. In *International Conference on Learning Representations*, volume 2026, pp. 155338–155362, 2026.

Milong Ren, Chungong Yu, Dongbo Bu, and Haicang Zhang. Accurate and robust protein sequence design with carbondesign. *Nature Machine Intelligence*, 6(5):536–547, 2024.

Michel F Sanner, Arthur J Olson, and Jean-Claude Spehner. Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers*, 38(3):305–320, 1996.

Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pp. 1383–1392. Wiley Online Library, 2009.

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.

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.

Kai Yi, Kiarash Jamali, and Sjors H. W. Scheres. All-atom inverse protein folding through discrete flow matching. In *International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=8tQdwSCJmA.

Yang Zhang and Jeffrey Skolnick. TM-align: A protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33(7):2302–2309, 2005. doi: 10.1093/nar/gki524.

Zaixiang Zheng, Yifan Deng, Dongyu Xue, Yi Zhou, Fei Ye, and Quanquan Gu. Structure-informed language models are protein designers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 42317–42338, 2023.