STRUCTURE-CONTRASTIVE REVEAL: GUIDING GENERATION ORDER IN PROTEIN INVERSE FOLDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein inverse folding aims to design amino-acid sequences compatible with a given protein backbone structure. Many recent models generate sequences iteratively, predicting amino acids at unresolved positions and filling only a subset of these positions at each decoding step. Decoded residues become context for later predictions. Thus, the order in which positions are selected can affect the final generation trajectory. Existing position-selection policies usually prioritize positions where the model is most confident. However, high confidence does not necessarily mean that a prediction is supported by the target structure. Since inverse folding is conditioned on a target structure, identifying predictions that actually rely on structural information is pivotal. Intuitively, if the predicted amino acid at a position is supported by the structure, weakening the structural information should reduce the probability assigned to that prediction. Motivated by this intuition, we explore comparing the probability of the same predicted amino acid under the original and weakened structural conditions, using the probability decrease as a proxy for structural dependence. In this work, we introduce **Structure-Contrastive Reveal (SCR)**, which combines this probability decrease with predictive confidence to decide which residue positions to decode first, while using the predictions under the original structural condition for the selected positions. Extensive experiments show that SCR consistently improves structural quality across different inverse-folding models and further enhances the state-of-the-art baseline under matched search configurations. An extension to trajectory-conditioned motion generation further suggests that condition-dependent reveal selection can generalize beyond protein inverse folding.

1 INTRODUCTION

Designing amino-acid sequences for given protein structures is a central problem in structure-based protein design. Given a target protein backbone, protein inverse folding aims to generate sequences compatible with the specified structure. Many recent methods (Dauparas et al., 2022; Hsu et al., 2022; Hayes et al., 2025; Liu et al., 2025) generate sequences iteratively, filling in the amino-acid sequence over multiple decoding steps. At each step, the model predicts amino-acid distributions at unresolved residue positions, but only a subset of positions is selected and added to the current sequence context. The revealed residues, *i.e., the predicted amino acids added at the selected unresolved positions*, are incorporated into the sequence context to inform subsequent predictions, making the choice of which positions to reveal crucial for later generation.

Previous studies have shown that *reveal order* can substantially affect the final generation quality (Zhong et al., 2026), yet how to determine an effective order remains unclear. Existing methods commonly determine reveal order based on randomness, predictive confidence, or predictive entropy (Dauparas et al., 2022; Hayes et al., 2025; Liu et al., 2025). Confidence prioritizes positions with larger maximum amino-acid probabilities, while entropy prioritizes positions with lower predictive-distribution entropy, both favoring positions where the model is more certain about its current prediction. Notably, ProtInvTree (Liu et al., 2025), a state-of-the-art tree-search method, introduces dedicated Focus strategies for selecting residue positions during tree search, but these heuristics do not consistently outperform simple random selection. One possible reason is that these

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

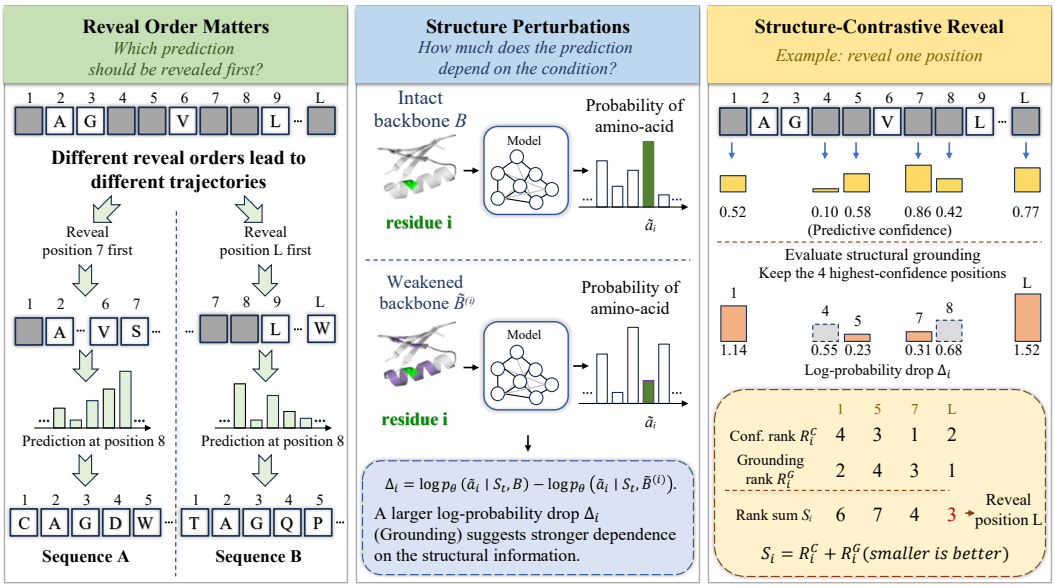

Figure 1: **Overview of Structure-Contrastive Reveal (SCR). Left:** Different reveal orders produce different partial-sequence contexts, changing predictions at the same unresolved position and leading to different final sequences. **Middle:** SCR compares the log-probability of the same sampled amino acid under the original and weakened structural conditions, while keeping the partial sequence and model parameters fixed. The resulting contrast provides a proxy for structural dependence. A local perturbation is illustrated. **Right:** In this example, one position is revealed. SCR retains the four highest-confidence positions, evaluates their structural grounding, and selects the position with the smallest sum of confidence and grounding ranks. The amino acid revealed at the selected position is sampled under the original structural condition.

heuristics do not directly measure how strongly each prediction is supported by the *target structure*. For example, a confident amino-acid prediction may be driven by previously revealed residues or learned sequence biases, in addition to the target structure. This motivates a complementary criterion for reveal selection: estimating how strongly each predicted amino acid depends on the target structural information.

In this work, we seek to determine whether a prediction is specifically supported by the target structure for inverse folding. Our key intuition is simple: if a predicted amino acid depends on structural information, weakening that information should reduce the probability of the same amino acid. Motivated by this intuition, we compare the probability of the same predicted amino acid under the original and structurally weakened conditions, and use the resulting probability decrease as a signal of structural support. Toward this end, we introduce **Structure-Contrastive Reveal (SCR)**, which explores structural support alongside predictive confidence to decide which residue positions should be revealed first.

The appropriate way to weaken structural information depends on the model's structural representation. For models (Jing et al., 2021; Dauparas et al., 2022; Hsu et al., 2022) operating directly on backbone geometry, the structural condition can be weakened by perturbing the three-dimensional backbone, while for models (Hayes et al., 2025) using discrete structural representations, it can instead be weakened in structure-token space. SCR therefore does not prescribe a single perturbation for all models. Instead, it provides a shared reveal principle: compare the same predicted amino acid before and after weakening structural information, and use the resulting response to guide reveal selection. Importantly, the weakened input is used only to determine reveal order, while the amino acid assigned to each selected position is always taken from the prediction under the original structural condition.

Experiments on our NanoIF model and frozen ESM3 (Hayes et al., 2025) show that SCR consistently improves inverse-folding quality and further enhances ProtInvTree on both models. Re-

sults on trajectory-conditioned human motion generation further support the broader applicability of condition-dependent reveal selection.

Our main contributions are as follows:

- We introduce Structure-Contrastive Reveal (SCR), a training-free reveal-selection framework that determines which positions should be revealed first during iterative decoding by exploiting estimated structural dependence.
- SCR estimates how strongly a predicted amino acid depends on the target structure by comparing model predictions before and after structural weakening, and combines this signal with predictive confidence to guide reveal selection.
- Extensive experiments across different inverse-folding models show that SCR consistently improves structural quality by changing reveal order. Its effectiveness on trajectory-conditioned human motion generation further supports the broader applicability of perturbation-based contrast for reveal selection in iterative conditional generation.

2 Related Work

2.1 Protein Inverse Folding

Protein inverse folding aims to design amino-acid sequences that are compatible with a given protein backbone structure. Existing methods differ in both their structural representations and generation procedures. Geometric graph-based approaches such as GVP-GNN (Jing et al., 2021) combine scalar and vector features to encode three-dimensional backbone geometry for sequence design. ESM-IF1 (Hsu et al., 2022) further scales inverse-folding training with millions of predicted protein structures using invariant geometric input processing and a sequence-to-sequence transformer. Autoregressive models such as ProteinMPNN (Dauparas et al., 2022) predict amino acids sequentially conditioned on both the target structure and previously generated residues, allowing earlier predictions to continuously influence subsequent generation. In contrast, parallel approaches such as PiFold (Gao et al., 2023) generate the entire sequence in a single pass for improved inference efficiency. Another line of work progressively improves sequences through iterative refinement. For example, LM-Design (Zheng et al., 2023) performs iterative refinement with a structure-conditioned protein language model, while methods such as MapDiff (Bai et al., 2025) further explore diffusion-based generation. More recently, general-purpose protein generative models such as ESM3 (Hayes et al., 2025) also support iterative sequence generation conditioned on protein structure.

Rather than introducing a new inverse-folding architecture or generative backbone, we focus on an orthogonal inference-time question in multi-step generation: *which residue predictions should be revealed first?*

2.2 Reveal Order and Test-Time Search

Compared with model architecture design, reveal order and test-time search have received relatively limited attention in protein inverse folding. Existing iterative generation methods typically determine which positions to reveal next using random ordering, model confidence, or predictive entropy. Recently, ProtInvTree (Liu et al., 2025) formulates inverse folding as reward-guided tree search, explicitly exploring multiple candidate generation trajectories using rewards provided by a protein structure prediction model. Its primary contribution lies in trajectory-level search and reward-based expansion, rather than explicitly estimating whether individual residue predictions are supported by the target structure.

Meanwhile, recent studies on masked generation have shown that token generation order itself can substantially affect final generation quality (Zhong et al., 2026). These studies establish the importance of generation order but leave open how sensitivity to the target structural condition can inform residue-level reveal selection.

2.3 Contrastive and Perturbation-Based Guidance

Another line of related work exploits changes in model predictions under different conditioning signals to guide generation. Classifier-Free Guidance (CFG) (Ho & Salimans, 2022) compares conditional and unconditional predictions to strengthen the dependence of generated samples on the conditioning signal. Similarly, Context-Aware Decoding (CAD) (Shi et al., 2024) contrasts language-model predictions with and without external context to distinguish generations supported by input evidence from those driven primarily by model priors. More recently, CoRe (Zhai et al., 2026) perturbs the current sequence context and measures the resulting changes in token likelihood to identify context-brittle predictions, which are subsequently used to guide remasking in diffusion language models.

Our method contrasts intact and weakened structural conditions using local or global structural weakening adapted to the model's structural representation. Unlike guidance methods that directly modify token probabilities, SCR uses the contrast only to decide which residue positions to reveal next, with the amino acid at each selected position still taken from the model prediction under the intact structural condition.

3 Method

In this section, we elaborate on the proposed method, Structure-Contrastive Reveal (SCR), as shown in Figure 1. In the following, we introduce three key parts, including iterative decoding, structural weakening and grounding, and structure-contrastive reveal.

3.1 Iterative Decoding

Decoding starts with all eligible residue positions masked and progressively reveals predictions at selected positions. Let B denote the target backbone, represented through the structural inputs used by the underlying model. At decoding step t, let S_t denote the partially revealed sequence and $\mathcal{M}_t$ the set of remaining masked positions. For each $i \in \mathcal{M}_t$, the model produces an amino-acid distribution $p_\theta(a_i \mid S_t, B)$. We obtain a proposal $\tilde{a}_i$ using the model's native token-sampling rule under the intact structural condition. Each unresolved position receives a predictive-confidence score under the intact structural condition:

$$C_i = \max_{a \in \mathcal{A}} p_\theta(a_i = a \mid S_t, B). \tag{1}$$

Larger values of C_i indicate higher confidence. In this work, SCR uses confidence to identify candidate positions and structural grounding to guide which position is revealed.

3.2 Structural Weakening and Grounding

We weaken the structural condition through *local or global perturbations* adapted to how each model represents and propagates structural information. SCR separates the construction of a weakened structural condition from the use of the resulting contrast to guide reveal order. Different weakening strategies may affect different structural information and result in different computational costs.

For each candidate position i, let $\widetilde{B}^{(i)}$ denote a weakened structural condition, with the partial sequence, model parameters, and all non-structural inputs held fixed. Local weakening reduces structural information around the candidate position, whereas global weakening uses a shared weakened condition for all candidate positions. The position index thus allows candidate-specific perturbations without requiring them. We instantiate these structural weakening strategies according to each model's structural representation: NanoIF is perturbed in backbone-coordinate space, while ESM3 is perturbed in its structure-token representation. Global SCR requires one shared weakened-condition forward pass per decoding step, whereas Local SCR evaluates candidate-specific structural weakening. Exact constructions are provided in Appendix A.

We evaluate the same intact-model proposal under the intact and weakened structural conditions and define the raw structure-contrastive signal as

$$\Delta_i = \log p_\theta(\tilde{a}_i \mid S_t, B) - \log p_\theta(\tilde{a}_i \mid S_t, \widetilde{B}^{(i)}). \tag{2}$$

A positive contrast means that weakening the structural condition reduces the probability of the intact proposal. We use this response as a proxy for the proposal's dependence on the structural information affected by the weakening operation. We define the grounding score G_i as:

$$G_i = \max(0, \Delta_i) \quad \text{for NanoIF}, \qquad G_i = \Delta_i \quad \text{for ESM3}.$$

where NanoIF retains the positive part of the contrast, and ESM3 retains its signed value. This difference reflects the implementations evaluated here. We do not isolate the effect of clipping.

Predictive confidence measures certainty, while grounding measures the response to structural weakening. SCR combines their rankings to incorporate both properties into reveal selection. This selection rule is shared across different weakening strategies, and the specific perturbation determines which structural information is probed. Further implementation details are provided in Appendix A.

3.3 Structure-Contrastive Reveal

SCR first restricts reveal selection to positions with relatively high predictive confidence. For local structural weakening, this restriction also reduces the number of candidate-specific evaluations. For a global structural weakening, a shared weakened-condition forward pass provides grounding scores for all candidate positions. Let q_t be the number of positions to reveal at step t. The candidate pool is

$$\mathcal{C}_t = \mathrm{Top}^{C}_{\min(\rho q_t, |\mathcal{M}_t|)}(\mathcal{M}_t), \qquad \rho = 4. \tag{3}$$

The Top operator selects the requested number of positions with the largest predictive-confidence scores. Grounding is evaluated only for positions in this candidate pool.

Within the candidate pool, we rank each position by predictive confidence and grounding:

$$\begin{aligned} R^C_i &= \mathrm{rank}_{\downarrow}\left(C_i; \{C_j : j \in \mathcal{C}_t\}\right), \\ R^G_i &= \mathrm{rank}_{\downarrow}\left(G_i; \{G_j : j \in \mathcal{C}_t\}\right). \end{aligned} \tag{4}$$

Here, rank 1 denotes the largest score.

We combine the rankings through equal-weight rank fusion and select the positions with the smallest fused scores:

$$S_i = R^C_i + R^G_i, \qquad \mathcal{R}_t = \mathrm{Bottom}^{S}_{q_t}(\mathcal{C}_t). \tag{5}$$

The operator $\mathrm{Bottom}^{S}_{q_t}$ returns the q_t positions with the smallest values of S_i. The candidate pool restricts selection to high-confidence positions, while rank fusion combines confidence and grounding without requiring the two scores to share a numerical scale.

At each selected position $i \in \mathcal{R}_t$, SCR reveals the intact-model proposal $\tilde{a}_i$. At a fixed decoding state, it preserves the intact model's token predictions and changes only reveal-position selection. This reveal operation updates S_{t+1}, thereby changing the sequence context available to subsequent predictions. Model-specific reveal schedules and sampling settings are provided in Appendix A.

4 Experiments

4.1 Experimental Setup

We evaluate SCR on the standard CATH 4.2 inverse-folding split introduced by Ingraham et al. (2019), derived from the CATH 4.2 structural classification database (Sillitoe et al., 2019). The split contains 18,024 training, 608 validation, and 1,120 test backbones. We consider two models: **NanoIF**, a 10.34M-parameter masked-denoising geometric GNN trained on the CATH 4.2 training split, and the frozen 1.4B-parameter **ESM3** `esm3-sm-open-v1` checkpoint.

Standard NanoIF decoding uses 16 decoding steps and generates four independent candidates per backbone. ESM3 uses 8 decoding steps and generates one candidate per backbone. The full-test-set main table uses Confidence as the standard-decoding baseline for both models. The separate ESM3 compute–quality study includes both Confidence and Entropy as its baselines. We also integrate SCR into ProtInvTree with a maximum of 50 tree iterations, expansion width 2, and planning depth 4. ProtInvTree results use the terminal candidate with the highest internal search reward.

All main structural evaluations compare ESMFold-predicted (Lin et al., 2023) structures with the target CATH backbones. ProtInvTree's internal reward instead compares the folded design with the folded native sequence. We report Mean TM, Median TM, Recovery, and RMSD, together with Best TM@4 and Success@4 for standard four-candidate NanoIF decoding. Full protocols and metric definitions are provided in Appendix A.

4.2 Main Results

Table 1: Main results on 1,120 CATH 4.2 test backbones. Standard NanoIF decoding uses four candidates per backbone: Mean TM, Recovery, and RMSD are averaged within each backbone and then across backbones. Median TM is the median of backbone-level mean TM-scores. Standard ESM3 decoding uses one candidate, and ProtInvTree reports one reward-selected terminal candidate per backbone. Bold values indicate the best standard decoding result within each model and the better result within each ProtInvTree pair.

Model	Method	Mean TM ↑	Median TM ↑	Recovery ↑	RMSD ↓	Best TM @4 ↑	Success @4 ↑
NanoIF	Confidence	0.7570	0.8389	0.3450	2.3733	0.8190	90.62%
NanoIF	Global SCR	0.7900	0.8688	0.3779	2.1680	0.8419	92.95%
NanoIF	Local SCR	**0.8104**	**0.8908**	**0.3886**	**2.0322**	**0.8518**	**93.57%**
NanoIF	ProtInvTree	0.8204	0.8982	0.3450	2.0228	–	–
NanoIF	ProtInvTree + Local SCR	**0.8399**	**0.9133**	**0.3773**	**1.8636**	–	–
ESM3	Confidence	0.8253	0.9020	0.3916	1.9304	–	–
ESM3	Local SCR	0.8438	0.9207	0.4014	1.7825	–	–
ESM3	Global SCR	**0.8539**	**0.9231**	**0.4092**	**1.7344**	–	–
ESM3	ProtInvTree	0.8538	0.9180	0.3843	1.7617	–	–
ESM3	ProtInvTree + Local SCR	**0.8634**	**0.9303**	**0.3988**	**1.6700**	–	–

Table 1 shows that both Local and Global SCR improve structural quality on both models, while their relative performance differs. Local SCR achieves the best standard decoding results on NanoIF, increasing Mean TM from 0.7570 to 0.8104. On frozen ESM3, Global SCR performs best, increasing Mean TM from 0.8253 for Confidence to 0.8539, compared with 0.8438 for Local SCR. The paired improvement over Local SCR is 0.0101, with a 95% bootstrap confidence interval of $[0.0051, 0.0152]$. Both variants improve Median TM, Recovery, and RMSD over the corresponding baseline.

SCR also complements reward-guided search. With the same ProtInvTree search configuration, adding Local SCR increases Mean TM from 0.8204 to 0.8399 on NanoIF and from 0.8538 to 0.8634 on ESM3. Oracle Best-of-16 TM-score also improves on both models, indicating that SCR improves the best structural candidate available in the terminal search set before reward-based selection. Global-SCR ProtInvTree results appear in the compute–quality study below, and Oracle results in Appendix B.3.

4.3 Compute–Quality Trade-offs

We evaluate ESM3 compute–quality trade-offs on a fixed subset of 200 CATH 4.2 test backbones shared by all configurations (Figure 2). For Entropy, Confidence, Local SCR, and Global SCR, we vary the number of decoding steps over 4, 8, 16, and 32, with the Local SCR candidate-pool multiplier fixed at four. For ProtInvTree and ProtInvTree+Local SCR, we vary expansion width over 2, 3, and 4. ProtInvTree+Global SCR is evaluated at width 2. Standard decoding reports mean quality and generation time per sampled sequence, whereas search methods report one reward-selected output and its complete search cost. Full protocols appear in Appendix A.6.

Global SCR provides the strongest observed trade-off among the tested standard-decoding configurations. Confidence and Entropy peak at approximately 0.831 and 0.828 Mean TM, respectively, while Global SCR reaches approximately 0.855. All evaluated Local SCR configurations have lower

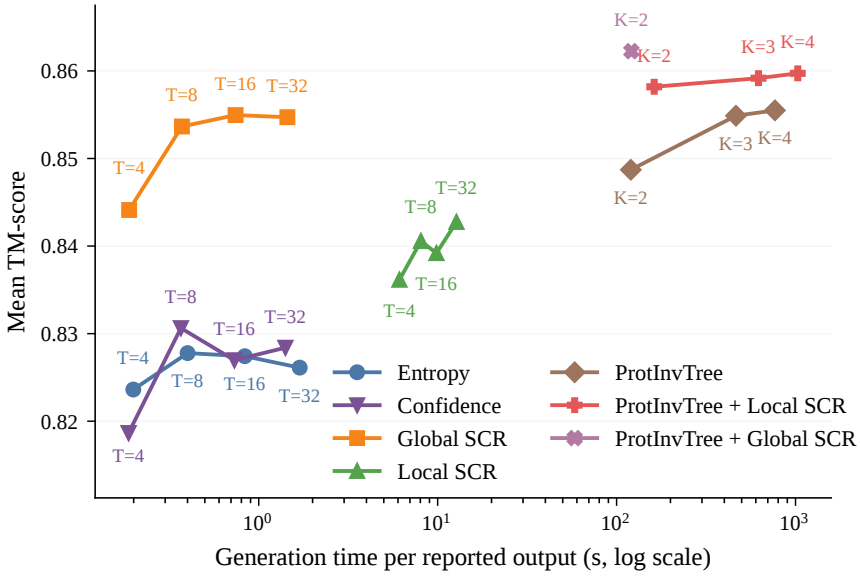

Figure 2: Compute–quality trade-offs on 200 CATH 4.2 test backbones using ESM3. Labels indicate decoding steps for standard decoding and expansion width for search. Local SCR uses a fixed candidate-pool multiplier of four. Standard decoding reports mean quality and time per sampled sequence. Search reports one reward-selected output and complete search time. Global SCR provides the best observed standard-decoding trade-off. At expansion width 2, ProtInvTree+Global SCR improves Mean TM over ProtInvTree with nearly unchanged measured search time.

Mean TM and higher generation cost than the tested Global SCR configurations. This contrasts with the preference for Local SCR on NanoIF and supports model-specific structural weakening selection.

Local SCR improves ProtInvTree at every evaluated expansion width. At width 2, it reaches Mean TM 0.8582, exceeding the approximately 0.856 obtained by unmodified ProtInvTree at width 4 while requiring less time.

At width 2, Global SCR increases Mean TM from 0.8487 to 0.8623, with a paired 95% bootstrap confidence interval of $[0.0064, 0.0216]$ for the improvement. Mean search time is 120.1 s for ProtInvTree and 121.0 s for ProtInvTree+Global SCR. Compared with ProtInvTree+Local SCR, Global SCR reduces runtime by 25.4% and achieves a higher observed Mean TM, although the paired quality difference is not statistically significant (95% CI: $[-0.0041, 0.0131]$). These results show that structural contrast can improve reward-guided search with negligible measured total runtime overhead in this configuration.

4.4 Grounding Responses to Structural Perturbations

Figure 3 examines how the grounding signal responds to different structural perturbations. The stronger responses to targeted and global structural perturbations indicate sensitivity to the structural condition. The contrast between targeted and remote perturbations further supports sensitivity to the location of the perturbed information.

On NanoIF, all tested structural weakening variants improve over Confidence, with Local SCR achieving the highest Mean TM and Median TM and the lowest RMSD (Appendix B.1). This preference differs from ESM3, where Global SCR provides the better observed compute–quality trade-off. Probe definitions, snapshot sampling, and aggregation are provided in Appendix A.7.

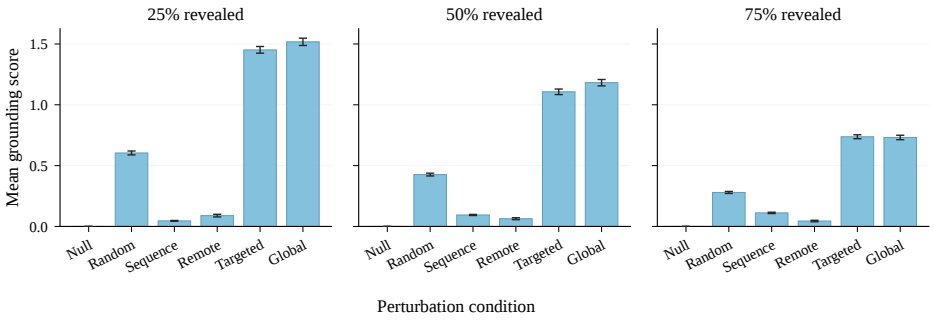

Figure 3: **Grounding responses to different perturbations.** Mean signed log-probability contrasts are shown at 25%, 50%, and 75% revealed positions. Targeted local and global structural perturbations produce the largest responses at each evaluated stage, with response magnitudes decreasing as decoding proceeds.

4.5 Structure-Contrastive Grounding Shapes Generation Trajectories

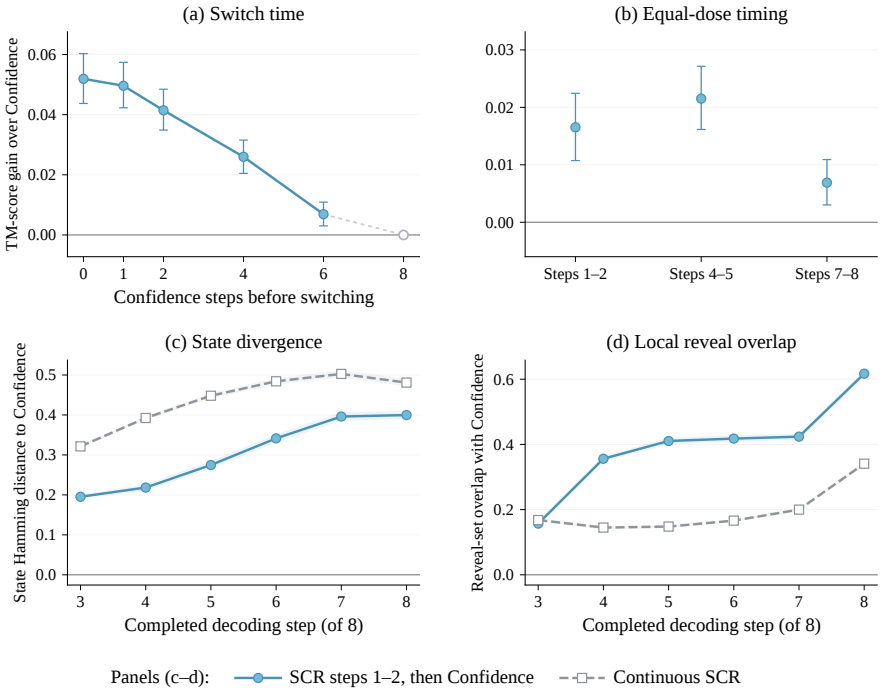

Figure 4: **Early structure grounding has persistent downstream effects. (a)** Final TM-score gain over Confidence decreases as the switch from Confidence to SCR is delayed. The open circle denotes Confidence used throughout decoding. **(b)** Early and middle placements yield larger observed mean gains than late placement. **(c)** Sequence-state Hamming distance from the Confidence baseline persists after SCR is used only at steps 1–2. **(d)** Subsequent reveal-set overlap with Confidence increases despite the persistent sequence-state differences. In **(c)** and **(d)**, blue solid lines denote SCR used only at steps 1–2 followed by Confidence, while gray dashed lines denote SCR used throughout decoding.

We next examine how the timing of structure-grounded reveal affects generation. Delaying the switch from confidence-based reveal to SCR reduces the final TM-score gain (Figure 4(a)). When SCR is applied at two decoding steps in each condition, early and middle placements yield larger

observed mean gains than late placement (Figure 4(b)). Thus, the benefit depends on when SCR is applied, even when the number of SCR-enabled steps is fixed.

Applying SCR only during the first two steps retains a final TM-score advantage after the decoder returns to confidence-based reveal. Subsequent reveal sets increasingly overlap with those of the Confidence baseline, yet sequence-state Hamming distance persists and grows over much of the remaining trajectory (Figure 4(c,d)). Returning to the same reveal criterion therefore does not restore the same generation trajectory. These observations are consistent with a path-dependent effect: early reveal decisions establish different partial-sequence contexts that influence later predictions. Intervention protocols, paired sampling, and trajectory metrics are detailed in Appendix A.8.

5 Beyond Protein Inverse Folding

We next apply condition-grounded reveal to human motion generation, using local trajectory constraints as the conditioning signal.

We evaluate the frozen MaskControl (Pinyoanuntapong et al., 2025) model on 1,200 HumanML3D (Guo et al., 2022) test motions conditioned on text and a target pelvis trajectory. Each motion token represents four frames. For each candidate, we remove the aligned trajectory condition while preserving the text and all other constraints. We then contrast the same intact-model proposal under the original and weakened conditions and combine grounding with confidence through rank fusion. The model and token-sampling rule remain unchanged.

We compare Targeted Grounding with two controls that disrupt token–condition correspondence. Shuffled Grounding randomly permutes grounding scores across candidates, whereas Random-Temporal Grounding removes conditioning over an unrelated temporal interval.

Table 2: Condition-grounded reveal on 1,200 HumanML3D test motions using frozen MaskControl. Lower is better for all metrics. Bold values indicate the best result in each column.

Reveal Policy	Traj. Error ↓	Foot Skating ↓	FID ↓
Confidence	0.3435	0.0840	0.4841
Shuffled Grounding	0.3375	0.0805	**0.4080**
Random-Temporal Grounding	0.3359	0.0798	0.4183
Targeted Grounding	**0.3208**	**0.0795**	0.4220

As shown in Table 2, Targeted Grounding reduces mean trajectory error from 0.3435 to 0.3208, a 6.6% reduction, with a paired 95% confidence interval of $[-0.0392, -0.0065]$ for the mean difference from Confidence.

Targeted Grounding achieves the lowest observed trajectory error, while all grounding variants improve foot skating and FID over Confidence. Shuffled Grounding obtains the lowest FID, indicating that the strongest trajectory adherence does not coincide with the best distributional metric.

The weakening is adapted to the conditioning representation: structural weakening for proteins and temporally localized trajectory removal for motion. The shared operation is to estimate a proposal's dependence on the conditioning signal and use it to prioritize which positions are revealed next.

6 Conclusion

We introduced Structure-Contrastive Reveal (SCR), a training-free policy that uses structure-contrastive grounding to guide reveal order while preserving intact-model proposals. Across NanoIF and ESM3, SCR improves both standard decoding and reward-guided search, with different models favoring different structural weakening strategies. Early interventions retain downstream benefits, and the same principle transfers to motion generation. These results support condition-sensitive reveal selection and model-specific structural weakening design as practical inference-time mechanisms.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, improving the clarity and readability of the manuscript, literature search, code implementation and debugging. All scientific ideas, methodological choices, experimental designs, analyses, and conclusions were developed and verified by the authors. All AI-assisted content and code were reviewed and validated by the authors, who take full responsibility for the final manuscript.

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*, 7:876–888, 2025. doi: 10.1038/s42256-025-01042-6.

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. 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/abs/10.1126/science.add2187.

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

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3D Human Motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5152–5161, June 2022.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. doi: 10.1126/science.ads0018. URL https://www.science.org/doi/abs/10.1126/science.ads0018.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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 Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8946–8970. PMLR, 17–23 Jul 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. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f3a4ff4839c56a5f460c88cce3666a2b-Paper.pdf.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from Protein Structure with Geometric Vector Perceptrons. In *International Conference on Learning Representations*, 2021. URL https://mlanthology.org/iclr/2021/jing2021iclr-learning/.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. 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/abs/10.1126/science.ade2574.

Mengdi Liu, Xiaoxue Cheng, Zhangyang Gao, Hong Chang, Cheng Tan, Shiguang Shan, and Xilin Chen. ProtInvTree: Deliberate protein inverse folding with reward-guided tree search. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0955.

Ekkasit Pinyoanuntapong, Muhammad Saleem, Korrawe Karunratanakul, Pu Wang, Hongfei Xue, Chen Chen, Chuan Guo, Junli Cao, Jian Ren, and Sergey Tulyakov. MaskControl: Spatio-temporal control for masked motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9955–9965, October 2025.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 783–791, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-short.69. URL https://aclanthology.org/2024.naacl-short.69/.

Ian Sillitoe, Natalie Dawson, Tony E Lewis, Sayoni Das, Jonathan G Lees, Paul Ashford, Adeyelu Tolulope, Harry M Scholes, Ilya Senatorov, Andra Bujan, Fatima Ceballos Rodriguez-Conde, Benjamin Dowling, Janet Thornton, and Christine A Orengo. CATH: expanding the horizons of structure-based functional annotations for genome sequences. *Nucleic Acids Research*, 47(D1): D280–D284, 01 2019. ISSN 0305-1048. doi: 10.1093/nar/gky1097. URL https://doi.org/10.1093/nar/gky1097.

Jinrui Xu and Yang Zhang. How significant is a protein structure similarity with TM-score = 0.5? *Bioinformatics*, 26(7):889–895, 04 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq066. URL https://doi.org/10.1093/bioinformatics/btq066.

Kevin Zhai, Sabbir Mollah, Zhenyi Wang, and Mubarak Shah. CORE: Context-robust remasking for diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=bmKHxLWkz9.

Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004. doi: https://doi.org/10.1002/prot.20264. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.20264.

Yang Zhang and Jeffrey Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research*, 33(7):2302–2309, 04 2005. ISSN 0305-1048. doi: 10.1093/nar/gki524. URL https://doi.org/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 Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42317–42338. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/zheng23a.html.

Yangyang Zhong, Yanmei Gu, Zhengqing Zang, Xiaomeng Li, Yuqi Ding, Xibei Jia, Yuting Shen, Zhenzhong Lan, Liwang Zhu, Weiping Liu, Junlin Zhou, Haisheng Liu, Zhong Xin Yu, Pengxin Luo, Donglian Qi, Yunfeng Yan, and Junbo Zhao. Parallelism and generation order in masked diffusion language models: Limits today, potential tomorrow. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 7178–7218, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.357. URL https://aclanthology.org/2026.findings-acl.357/.

A Implementation and Experimental Details

A.1 SCR Instantiations

NanoIF and ESM3 use different structural representations and therefore require different local and global weakening strategies. In both cases, however, SCR preserves the same separation of roles: amino-acid proposals are generated exclusively under the intact structural condition, whereas weakened structural conditions are used only to estimate structural grounding and rank reveal positions. The amino acid revealed at each selected position is always the proposal sampled from the intact model. Predictive confidence and structural grounding are combined using the rank-fusion rule defined in Equation 5.

NanoIF. NanoIF is a 10.34M-parameter ProteinMPNN-style masked-denoising geometric GNN. It operates on backbone coordinates (N, C_α, C, O) and constructs a 32-nearest-neighbor graph using C_α distances. For each residue i, we construct a self-excluding neighborhood $\mathcal{N}_{32}(i)$ from the intact backbone and reuse it throughout decoding.

Local SCR. For a candidate center residue i, the local structural condition is weakened by contracting the N, C, and O coordinates of each $j \in \mathcal{N}_{32}(i)$ toward its own C_α coordinate:

$$\widetilde{\mathbf{x}}_{j,a}^{(i)} = \mathbf{x}_{j,C_\alpha} + \lambda \left(\mathbf{x}_{j,a} - \mathbf{x}_{j,C_\alpha} \right), \qquad a \in \{N, C, O\}, \qquad \lambda = 0.25. \tag{6}$$

Thus, C_α coordinates remain unchanged, while the local C_α-relative backbone displacement is reduced to 25% of its original magnitude. The center residue and residues outside $\mathcal{N}_{32}(i)$ are left unchanged.

Global SCR. The same structural contraction is applied to every residue along the backbone:

$$\widetilde{\mathbf{x}}_{j,a}^{\text{global}} = \mathbf{x}_{j,C_\alpha} + \lambda \left(\mathbf{x}_{j,a} - \mathbf{x}_{j,C_\alpha} \right), \qquad j \in \{1, \ldots, L\}, \quad a \in \{N, C, O\}, \quad \lambda = 0.25. \tag{7}$$

The global construction therefore perturbs the N, C, and O coordinates throughout the full sequence while preserving all C_α coordinates. It produces one shared weakened backbone for all candidate positions, whereas the local construction produces a candidate-specific weakened backbone.

At decoding step t, NanoIF first performs an intact forward pass using the current partial sequence S_t. Amino-acid proposals are sampled from the intact amino-acid distributions. Local SCR performs a separate forward pass under a candidate-specific weakened structural condition for each candidate position. Global SCR performs one forward pass on the shared globally weakened backbone. Both use the same S_t as the intact forward pass.

NanoIF uses the predictive-confidence score defined in Equation 1, computed from the intact amino-acid distribution, while structural grounding evaluates the sampled intact proposal $\tilde{a}_i$:

$$G_i = \max \left(0, \log p_{\text{intact}}(\tilde{a}_i) - \log p_{\text{weak}}^{(i)}(\tilde{a}_i) \right). \tag{8}$$

Confidence first restricts structural evaluation to the Top-$4q_t$ unresolved positions, where q_t is the reveal quota at step t. Confidence and grounding are then independently ranked within this candidate pool and combined using Equation 5. The q_t positions with the smallest fused ranks are revealed.

NanoIF uses 16 decoding steps. The reveal quota is chosen to distribute the remaining eligible positions across the remaining steps:

$$q_t = \left\lceil \frac{\text{number of remaining eligible positions}}{\text{number of remaining decoding steps}} \right\rceil. \tag{9}$$

Four independent candidates are generated per backbone using stochastic sampling with temperature 1.0 and no top-k or top-p truncation.

ESM3. We use the frozen 1.4B-parameter `EvolutionaryScale/esm3-sm-open-v1` checkpoint. Each target backbone is converted to the native ESM3 representation and encoded once into a discrete structure-token track $Z = (z_{\mathrm{BOS}}, z_1, \ldots, z_L, z_{\mathrm{EOS}})$. A self-excluding 32-nearest-neighbor graph is constructed from the intact C_α coordinates and reused throughout decoding.

Structural conditioning during sequence generation is provided through the encoded structure-token track. *Local SCR.* For a candidate center residue i, the local structural weakening is constructed by masking the structure token at i and those of its 32 nearest neighbors. All structure tokens outside this local neighborhood remain unchanged.

Global SCR. All residue-level structure tokens are masked:

$$\widetilde{Z}^{\mathrm{global}} = (z_{\mathrm{BOS}}, z_{\mathrm{MASK}}, \ldots, z_{\mathrm{MASK}}, z_{\mathrm{EOS}}). \tag{10}$$

The BOS and EOS tokens and all non-structural inputs are preserved. Local SCR evaluates each candidate-specific token mask with a separate ESM3 sequence forward pass, whereas Global SCR evaluates the shared fully masked structure track once per decoding step. Both use the same partial sequence S_t as the intact pass.

At each decoding step, amino-acid proposals are sampled from the intact sequence logits. ESM3 uses the same predictive-confidence definition as NanoIF (Equation 1). Positions with larger confidence scores receive higher priority during candidate selection.

Structural grounding is the signed log-probability contrast,

$$G_i = \log p_{\mathrm{intact}}(\tilde{a}_i) - \log p^{(i)}_{\mathrm{weak}}(\tilde{a}_i). \tag{11}$$

Unlike the NanoIF instantiation, negative grounding values are retained. Confidence first restricts evaluation to the $4q_t$ highest-confidence unresolved positions. Confidence and grounding are then ranked in descending order within this candidate pool and combined using Equation 5.

SCR changes only which currently masked positions are selected for reveal. Amino-acid predictions are sampled from the intact structure-conditioned ESM3 distributions, and the number of positions revealed at each step follows the same native cosine schedule as the Confidence baseline. Once revealed, an amino-acid token remains fixed and serves as sequence context for subsequent steps; neither Confidence nor SCR remasks or refines revealed positions. We use eight decoding steps, temperature 0.7, and top-$p = 1.0$. One candidate is generated per backbone.

A.2 Dataset and NanoIF Training

We use the standard CATH 4.2 inverse-folding split introduced by Ingraham et al. (2019), containing 18,024 training, 608 validation, and 1,120 test backbones with sequence lengths between 40 and 500 residues. The input backbone representation contains the (N, C_α, C, O) coordinates for each residue. A residue is treated as structurally valid only when all four backbone atoms are available and finite. Invalid residues are excluded from structural processing, decoding, and recovery evaluation. The validation set is used for model and method development. The main-result comparisons are reported on the unchanged 1,120-backbone test set, whereas the compute–quality study uses a fixed 200-backbone subset shared by all configurations in that study.

NanoIF is a ProteinMPNN-style masked-denoising geometric GNN with 10,335,252 parameters. It uses hidden dimension 256, edge dimension 128, six encoder layers, six decoder layers, eight attention heads, and a 32-nearest-neighbor structural graph based on C_α distances.

NanoIF is trained on the CATH 4.2 training split using masked amino-acid denoising. Valid residues are independently masked with probability 0.5, with at least one masked position retained per protein. Cross-entropy loss is computed only over masked residues with valid backbone coordinates. We use AdamW with learning rate 2×10^{-4} and weight decay 0.01, without label smoothing or a learning-rate schedule. All NanoIF experiments use the same fixed trained checkpoint.

A.3 PROTINVTREE CONFIGURATION

For the full 1,120-backbone main-result comparisons, ProtInvTree is evaluated with maximum tree iterations $M = 50$, planning depth $T = 4$, expansion width $K = 2$, one rollout sample per child, and cosine quota scheduling. The SCR-enhanced full-test-set results use Local SCR. Each search produces 16 deduplicated terminal candidates per backbone.

The compute–quality study instead uses the fixed 200-backbone subset described in Appendix A.6. On this subset, ProtInvTree and ProtInvTree+Local SCR are evaluated at widths $K \in \{2, 3, 4\}$, whereas ProtInvTree+Global SCR is evaluated only at width $K = 2$. We do not report a full-test-set ProtInvTree+Global SCR result.

The internal search reward measures structural agreement between the folded designed sequence and the folded native sequence using ESMFold-based TM-score. The native sequence structure is computed once per backbone and reused during search.

Primary ProtInvTree results use the terminal candidate with the highest internal search reward. The 16 terminal candidates therefore constitute an internal test-time search set rather than 16 independent samples from the final model. Oracle Best-of-16 results, when reported, are explicitly labeled and use ground-truth structural quality only for retrospective analysis.

A.4 STRUCTURAL EVALUATION

All main experiments use ESMFold (Lin et al., 2023) as the sequence-to-structure evaluator. We use `esmfold_3B_v1` in inference mode with four recycling iterations and without MSA or templates.

For each generated sequence, the predicted structure is aligned to the corresponding target CATH backbone using `tmtools`, which implements the TM-align structural alignment procedure (Zhang & Skolnick, 2005). Structural similarity is quantified using TM-score (Zhang & Skolnick, 2004), normalized by the target CATH chain length.

A.5 METRIC AGGREGATION

For NanoIF, four candidates are generated for each backbone. Mean TM, Recovery, and RMSD are first averaged across the four candidates belonging to the same backbone and then averaged across the 1,120 test backbones. Median TM is the median of these 1,120 backbone-level candidate-mean TM-scores.

Best TM@4 selects the highest TM-score among the four candidates for each backbone before averaging across backbones. Success@4 is the fraction of backbones for which at least one of the four candidates achieves TM-score ≥ 0.5 (Xu & Zhang, 2010).

For standard ESM3 decoding, each backbone has one generated candidate, so metrics are computed directly across the 1,120 outputs.

For ProtInvTree, metrics are computed from the reward-selected Top-1 terminal candidate for each backbone. We therefore do not report standard Best TM@4 or Success@4 for ProtInvTree, since its terminal candidate set is generated internally by the search procedure rather than as a set of independent final samples.

A.6 COMPUTE–QUALITY EVALUATION DETAILS

Compute configurations. The Local and Global ESM3 weakening strategies are defined in Appendix A.1. For Entropy, Confidence, Global SCR, and Local SCR, we vary the number of decoding steps $T \in \{4, 8, 16, 32\}$. All use stochastic sampling with temperature 0.7, top-$p = 1.0$, and the same normalized cosine reveal schedule:

$$m_{t+1}^{\text{target}} = \text{int}\left[L \cos\left(\frac{t+1}{T}\frac{\pi}{2}\right) + 0.1\right], \qquad q_t = m_t - m_{t+1}^{\text{target}}.$$

Changing T therefore only rediscretizes the same cumulative schedule. Local SCR fixes the candidate pool to Top-$(4q)$ and $K_{\text{NN}} = 32$.

For ProtInvTree and ProtInvTree+Local SCR, we vary expansion width $K \in \{2, 3, 4\}$ while fixing maximum iterations $M = 50$, planning depth 4, and one rollout sample per child. ProtInvTree+Local SCR replaces the standard Focus policy with Local-SCR Focus while leaving the remaining search settings unchanged. ProtInvTree+Global SCR is evaluated only at $K = 2$ and replaces the standard Focus policy with Global-SCR Focus.

Timing protocol. Both structural quality and timing are measured over the same fixed subset of 200 CATH 4.2 test backbones shared by every compute–quality configuration. Timing is measured on a single NVIDIA A40 48 GB GPU. ESM3 is loaded in BF16, while logits and grounding-score calculations are performed in FP32. Wall-clock measurements include iterative generation, all intact-condition and weakened-condition ESM3 evaluations, and stochastic sampling, but exclude model loading, one-time backbone structure encoding and KNN-cache construction, and the final ESMFold evaluation used only for reporting structural metrics.

For standard ESM3 decoding, four independent sequences are generated sequentially per backbone. If $t_{m,b}$ denotes the total generation time for the four candidates of backbone b, the reported per-sequence cost is

$$C_m = \frac{1}{200 \times 4} \sum_{b=1}^{200} t_{m,b}.$$

Mean TM is averaged over all four candidates without oracle or reward-based selection.

ProtInvTree timing. ProtInvTree timing includes the complete tree search, all ESMFold evaluations required by its internal structural reward, and the native-sequence reference fold used during search. The reported output is the terminal candidate selected by the search reward, so its cost is the complete search time per backbone and is not divided by the number of internal candidates. The final ESMFold evaluation used uniformly for paper metrics is excluded from all methods.

A.7 Grounding Probe Protocol

We evaluate grounding diagnostics on the 1,120 CATH 4.2 test backbones using the frozen NanoIF model and Confidence Reveal. Snapshots are taken when approximately 25%, 50%, and 75% of residues have been revealed. At each snapshot, probes are evaluated on the highest-confidence $\min(|\mathcal{M}_t|, 4q_t)$ unresolved positions.

For each candidate position, the same amino-acid proposal is evaluated under the intact and perturbed inputs. Figure 3 reports the signed log-probability contrast between the two evaluations, without clipping negative values. The diagnostic probes do not alter the underlying Confidence trajectory.

We consider six perturbations: **Null**, which repeats the intact forward pass; **Targeted**, which weakens the local structure around the target position; **Random**, which applies the same perturbation around randomly selected control positions; **Remote**, which perturbs spatially distant control regions; **Sequence**, which remasks previously revealed nearby sequence tokens while preserving the backbone; and **Global**, which weakens the entire backbone. Local structural neighborhoods are defined using the 32 nearest valid residues by C_α distance.

Scores are averaged within each backbone and then across backbones. Error bars denote 95% bootstrap confidence intervals.

The Random Structure Probe in Table 3 uses the same general random-control perturbation, but as a decoding policy rather than a fixed-state diagnostic: its scores determine which positions are revealed and therefore affect subsequent decoding states.

A.8 Timing and Trajectory Diagnostics

We conduct the timing diagnostics in Figure 4 using the frozen NanoIF model and Local SCR on a fixed 400-backbone subset of the CATH 4.2 test set, with four paired candidates per backbone. All conditions use the same eight-step decoding budget. Paired trajectories share the same stateless Gumbel proposal noise at corresponding steps and residues. Each residue state is either MASK or a revealed amino-acid token, so state distance includes differences in reveal status.

Figure 4(a) varies the number of initial Confidence steps before switching to SCR over an eight-step trajectory. Figure 4(b) applies SCR only during steps 1–2, 4–5, or 7–8, using Confidence at all remaining steps. Figure 4(c) and (d) compare two matched eight-step protocols: SCR applied only at steps 1–2 followed by Confidence for steps 3–8, and SCR applied continuously for all eight steps. Both are compared against an eight-step Confidence trajectory under paired proposal noise.

Trajectory divergence is measured by the fraction of coordinate-valid positions whose current states differ:

$$D_{\text{state}} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{1}\left[S_i^C \neq S_i^{\text{SCR}}\right]. \tag{12}$$

Reveal overlap for the positions newly revealed in the current step is defined as

$$O_t = \frac{|R_t^C \cap R_t^{\text{SCR}}|}{\min\left(|R_t^C|, |R_t^{\text{SCR}}|\right)}. \tag{13}$$

Candidate-level measurements are first averaged within each backbone and then across the panel. Panels (a) and (b) of Figure 4 report 95% bootstrap confidence intervals, while panels (c) and (d) show backbone-averaged trajectory statistics.

A.9 Human Motion Generation Experimental Setup

We evaluate the frozen MaskControl model on 1,200 motions from the official HumanML3D test split. Motions use the standard 263-dimensional HumanML3D representation with 22 joints, and each discrete motion token corresponds to four frames. The model is conditioned on text and the ground-truth pelvis trajectory (joint 0) over all valid frames. All pretrained model parameters remain frozen.

All policies use the same iterative decoding configuration: 10 decoding steps, a cosine mask schedule, temperature 1.0, classifier-free guidance scale 4.0, and residual-decoder conditioning scale 5.0. Token sampling follows the original MaskControl sampler. Reveal policies modify only which masked positions are revealed.

Confidence Reveal selects the highest-confidence masked positions. For Targeted Grounding, when q positions are to be revealed, we first retain the $4q$ highest-confidence candidates. For each candidate i, we remove the pelvis-trajectory condition over its corresponding four-frame interval and compute

$$G_i = \log p_\theta(\hat{z}_i \mid S_t, C) - \log p_\theta(\hat{z}_i \mid S_t, \widetilde{C}_i), \tag{14}$$

where $\hat{z}_i$ is the intact-condition token proposal, S_t is the current decoding state, C is the original condition, and $\widetilde{C}_i$ removes only the temporally aligned local trajectory constraint. Candidates are ranked independently by confidence and grounding score, and the two ranks are summed with equal weight. The q candidates with the smallest combined ranks are revealed.

Shuffled Grounding randomly permutes grounding scores across candidate positions before rank fusion. Random-Temporal Grounding instead removes the trajectory condition over an unrelated valid token interval. Both controls retain the same candidate selection, condition-weakening evaluation, and rank-fusion procedure.

We report mean pelvis trajectory error, the official MaskControl foot-skating metric, and FID using the standard HumanML3D evaluator (`Comp_v6_KLD005`). For trajectory error, paired comparisons against Confidence Reveal use 10,000 bootstrap resamples and 95% confidence intervals.

B Additional Experimental Results

B.1 Structure-Contrastive Design Ablation

Table 3 compares alternative structure-contrastive decoding designs. Local SCR obtains the highest Mean and Median TM-scores and the lowest RMSD. Edge-Attenuated SCR constructs the structural weakening by scaling edge-mediated messages in the GNN to 25% of their original values, rather than perturbing backbone coordinates.

Table 3: Ablation of structure-contrastive reveal designs on NanoIF.

Method	Mean TM ↑	Median TM ↑	Recovery ↑	RMSD ↓
Confidence	0.7570	0.8389	0.3450	2.3733
Random Structure Probe	0.7868	0.8654	0.3671	2.2119
Global SCR	0.7900	0.8688	0.3779	2.1680
Edge-Attenuated SCR	0.8058	0.8880	**0.4010**	2.0595
Local SCR	**0.8104**	**0.8908**	0.3886	**2.0322**

B.2 Additional Decoding Baselines

Table 4 reports standard reveal baselines omitted from the main table for compactness.

Table 4: Additional standard decoding baselines on CATH 4.2.

Model	Method	Mean TM ↑	Median TM ↑	Recovery ↑	RMSD ↓	Best TM @4 ↑	Success @4 ↑
NanoIF	Random	0.7514	0.8362	0.3384	2.4117	0.8149	89.82%
NanoIF	Entropy	0.7456	0.8301	0.3326	2.4412	0.8077	89.73%
ESM3	Random	0.8181	0.8949	0.3790	1.9729	–	–
ESM3	Entropy	0.8240	0.9023	0.3895	1.9423	–	–

B.3 Oracle Best-of-16 Analysis

ProtInvTree normally selects its final output using the internal search reward. To distinguish improvements in candidate generation from improvements in reward-based candidate selection, we additionally evaluate an oracle setting that retrospectively selects the terminal candidate with the highest ground-truth TM-score.

Table 5: Oracle Best-of-16 performance. The terminal candidate with the highest ground-truth TM-score is selected for each backbone.

Model	Method	Mean TM	Median TM	Recovery	RMSD
NanoIF	ProtInvTree	0.8409	0.9148	0.3447	1.8960
NanoIF	ProtInvTree + Local SCR	**0.8647**	**0.9284**	**0.3775**	**1.7118**
ESM3	ProtInvTree	0.8805	0.9360	0.3842	1.5874
ESM3	ProtInvTree + Local SCR	**0.8933**	**0.9470**	**0.3974**	**1.4801**

Table 5 shows that ProtInvTree+Local SCR improves Oracle Best-of-16 performance on both NanoIF and ESM3, indicating that Local SCR improves the structural quality of the terminal candidate set itself rather than only changing which candidate is preferred by the search reward.