TWINMP: ADAPTING INVERSE-FOLDING MODELS FOR MULTI-CONFORMATION DESIGN AND MUTATION-EFFECT PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein inverse folding models learn strong priors from individual protein backbones, but typically overlook the conformational variability that is often essential to protein function. Existing multi-conformation adaptations either pool structural representations, potentially obscuring conformation-specific constraints, or process each conformation independently and combine their logits only during decoding, without information exchange between structural states. We introduce **Twin**-Stream with **M**essage **P**assing (**TwinMP**), a model-agnostic post-training framework that adapts pretrained single-conformation inverse folding models to dual-conformation learning. TwinMP maintains two streams that separately preserve the structure-to-sequence mapping of each conformation while introducing bidirectional message passing between their hidden representations before sequence prediction. This design retains the single-state knowledge acquired during pretraining while enabling explicit exchange and reconciliation of shared and state-specific structural constraints. We instantiate TwinMP on ProteinReasoner, ESM-IF, and ProteinMPNN. Across all three model families, TwinMP improves mean amino-acid recovery over pooling and Logit mean, with backbone-dependent gains from message passing. Moreover, TwinMP with AF2 and MD-derived structural inputs achieves an official ProteinGym score of 0.486, compared with 0.473 for the single-conformation baseline. These results support TwinMP as a practical framework for adapting pretrained inverse-folding models to multi-conformation design and mutation-effect prediction.

1 INTRODUCTION

Protein inverse folding aims to design amino-acid sequences compatible with a target three-dimensional backbone. Representative methods, including ProteinMPNN (Dauparas et al., 2022), ESM-IF (Hsu et al., 2022), and PiFold (Gao et al., 2023), have substantially improved sequence recovery and structure consistency by learning conditional structure-to-sequence mappings. Complementing these task-specific architectures, protein language models such as DPLM (Wang et al., 2024) and ProteinReasoner (Liu et al., 2025) have also demonstrated inverse-folding capabilities by conditioning sequence generation on protein structures. Despite their architectural differences, these models are typically trained and applied by conditioning on a single static backbone at a time.

A single structural snapshot may provide an incomplete description of a functional protein. Enzymatic catalysis, allosteric regulation, molecular recognition, and membrane transport frequently depend on transitions among multiple conformational states (Wei et al., 2016; Guo & Zhou, 2016; del Alamo et al., 2022; Janson et al., 2023; Chen et al., 2024). This consideration is also relevant to enzyme engineering, where local conformational flexibility and active-site preorganization can affect functional properties (Cao et al., 2023; Wei & Bornscheuer, 2026).

A sequence optimized for only one conformation may not preserve the flexibility or contact rearrangements required for another. Multi-conformation inverse folding, therefore, seeks a single sequence that is compatible with multiple structural states, requiring the model to reconcile constraints shared across states with requirements specific to each conformation.

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.

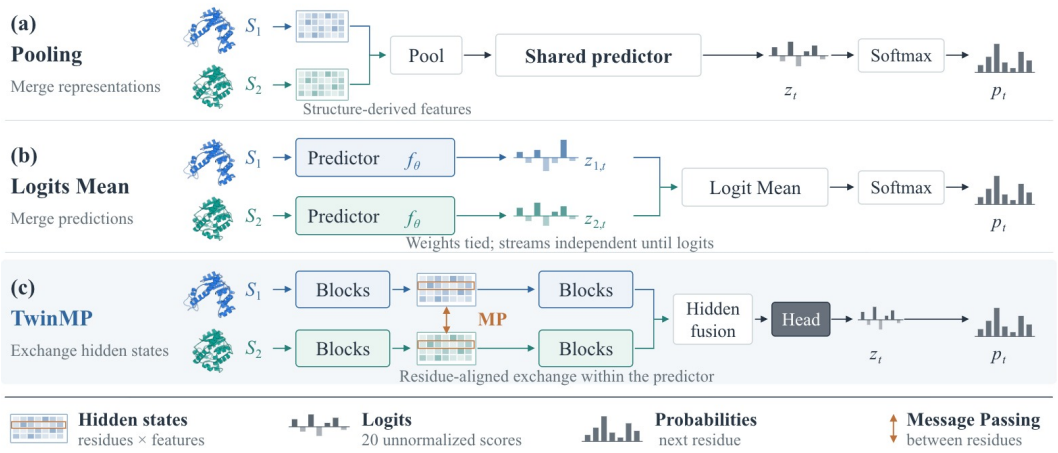


Figure 1: Conformational integration strategies. (a) Early pooling combines structure-derived representations. (b) Logit mean combines predictions from parameter-tied streams. (c) TwinMP exchanges residue-aligned hidden states before fusion and sequence prediction.

Two common strategies for handling multi-conformation information in inverse-folding models are illustrated in Figure 1. As shown in Figure 1(a), representation-pooling methods independently encode multiple conformations and aggregate their structural representations before sequence decoding. DynamicMPNN (Abrudan et al., 2025), for example, applies order-invariant pooling to obtain a shared representation across structural states. While this enables joint conditioning, collapsing the states into a single representation may obscure the distinct constraints that each conformation imposes on the target sequence. In contrast, the output-level strategy illustrated in Figure 1(b) processes each conformation with an independent single-state branch and combines the resulting amino-acid logits during autoregressive decoding. ProteinMPNN-based multi-state design, for example, has been applied to two-state hinge protein design by averaging the logits predicted independently from the two target conformations (Praetorius et al., 2023). This strategy preserves the original single-conformation computation of each branch, but the branches remain isolated until the final residue decision. Although both branches receive the same generated sequence prefix, structural evidence from one conformation cannot influence the intermediate representation of the other. The resulting prediction therefore combines state-specific outputs without cross-conformation interaction in hidden space.

This motivates a paired-stream adaptation that preserves state-specific computation while allowing the two streams to be jointly optimized and, when beneficial, to exchange information before prediction. We therefore propose **TwinMP**, a model-agnostic post-training framework for adapting pretrained single-conformation inverse-folding models to dual-conformation learning. Built on two parameter-tied streams, TwinMP processes the two conformations separately while providing both streams with the same autoregressive sequence prefix. The original inverse-folding computation of each backbone is retained, together with conformation-specific hidden states. Bidirectional message passing between residue-aligned representations then allows evidence from one conformation to refine the other before the two streams produce a shared amino-acid prediction. By combining state-specific reasoning with cross-conformation interaction, TwinMP reuses pretrained backbones without requiring a dedicated multi-conformation architecture to be trained from scratch.

We instantiate TwinMP on three distinct inverse-folding backbone families: ProteinReasoner, a multimodal protein language model conditioned on discrete structure tokens; ESM-IF, a geometric encoder–decoder inverse-folding model; and ProteinMPNN, a graph-based protein sequence design model. For each backbone, we compare TwinMP with the original single-conformation model, a post-trained representation-pooling variant, and an Logit mean baseline. Under separately decontaminated evaluation sets for each model family, TwinMP improves mean amino-acid recovery over pooling and Logit mean across all three backbones, with backbone-dependent gains from message passing and mixed structural outcomes. These results demonstrate the applicability of the twin-stream adaptation across architecturally different models.

Protein function is often coupled to conformational changes, suggesting that multiple structural states may provide complementary information about sequence–structure–function relationships. We therefore investigate whether multi-conformation inverse-folding post-training also improves a model's ability to interpret the functional effects of sequence mutations. To this end, we evaluate TwinMP-ProteinReasoner on zero-shot mutation fitness prediction using ProteinGym (Notin et al., 2023). At a fixed TwinMP checkpoint, replacing the duplicated AF2 input with an MD equilibration endpoint improves the official ProteinGym score from 0.479 to 0.486 on 214 assays, compared with 0.473 for the single-structure baseline. This matched comparison supports the utility of an alternative structural input for mutation-effect predictions, while leaving the separate contributions of message passing and structural relaxation to further controls.

Our contributions are summarized as follows:

- We propose **TwinMP**, a model-agnostic post-training framework that preserves conformation-specific inverse-folding streams while enabling bidirectional message passing between residue-aligned hidden representations.
- We instantiate TwinMP on ProteinReasoner, ESM-IF, and ProteinMPNN, and evaluate them on backbone-specific decontaminated sets, reporting repeated sequence-recovery results and complementary structural compatibility measurements.
- We show that a single MD-derived alternative input improves zero-shot ProteinGym prediction at a fixed TwinMP checkpoint, using a matched 214-assay comparison.

2 Related Work

2.1 Single-Conformation Inverse Folding Models

Protein inverse folding designs amino-acid sequences conditioned on a target backbone structure. Early learning-based methods formulated the task as conditional generation over protein graphs (Ingraham et al., 2019). Representative models such as ProteinMPNN (Dauparas et al., 2022), ESM-IF (Hsu et al., 2022), and PiFold (Gao et al., 2023) improve fixed-backbone design through geometric encoding and autoregressive or message-passing decoding. Protein language models, including LM-Design (Zheng et al., 2023), DPLM (Wang et al., 2024), and ProteinReasoner (Liu et al., 2025), further extend pretrained sequence priors to structure-conditioned generation. Despite their architectural differences, these methods generally process one static backbone at a time.

2.2 Multi-Conformation Fusion for Inverse Folding

Protein function often depends on transitions among conformational states involved in catalysis, allostery, recognition, and transport (Wei et al., 2016; Guo & Zhou, 2016). Multi-conformation inverse folding therefore seeks one sequence compatible with multiple backbone states. Hsu et al. (Hsu et al., 2022) also study dual-state conditioning using PDBFlex, and ADFLIP evaluates multi-structure inverse folding on NMR ensembles (Yi et al., 2025). DynamicMPNN jointly conditions on multiple conformations by pooling state-specific structural representations before sequence generation (Abrudan et al., 2025). Alternatively, output-level methods independently process each conformation and combine their amino-acid distributions during decoding, as used in two-state hinge protein design (Praetorius et al., 2023). Representation pooling may obscure the distinct sequence constraints imposed by each conformation, whereas output-level fusion prevents interaction between their intermediate representations. TwinMP instead preserves separate conformation-specific streams and enables cross-conformation message passing in hidden space.

2.3 Structure-Aware Mutation Effect Prediction

Protein language models can predict mutation effects without assay-specific supervision by comparing probabilities assigned to mutant and wild-type residues (Meier et al., 2021). ProteinGym provides a standardized benchmark for evaluating protein fitness prediction across deep mutational scanning assays (Notin et al., 2023). Structure-aware methods such as SaProt (Su et al., 2023) and S3F (Zhang et al., 2024) show that structural context can improve mutation effect prediction. However, these methods generally represent each protein using a single structural state. We examine whether

alternative structural inputs improve zero-shot mutation-effect prediction within a multi-conformation inverse-folding model.

3 METHOD

3.1 PROBLEM FORMULATION

We consider dual-conformation inverse folding, where the objective is to design one amino-acid sequence compatible with two structural states of the same protein. Each example consists of two residue-aligned backbone conformations, $\mathbf{X}_1$ and $\mathbf{X}_2$, and a shared target sequence $\mathbf{y} = (y_1, \ldots, y_L)$ defined over their common residue core.

We model the sequence autoregressively as $p(\mathbf{y} \mid \mathbf{X}_1, \mathbf{X}_2) = \prod_{t=1}^{L} p(y_t \mid \mathbf{X}_1, \mathbf{X}_2, \mathbf{y}_{<t})$, where $\mathbf{y}_{<t}$ is the prefix shared by both streams.

3.2 TWINMP OVERVIEW

TwinMP adapts a pretrained single-conformation inverse-folding model to dual-conformation sequence design using two parameter-tied streams, as illustrated in Figure 2. Each stream processes one aligned conformation through the original structure-to-sequence pathway, while both streams receive the same autoregressive sequence prefix and retain separate conformation-specific hidden states.

After backbone-specific layers, TwinMP exchanges bidirectional messages between residue-aligned hidden representations. The interaction is applied after the original layer computation, preserving the pretrained single-conformation pathway while allowing each stream to incorporate information from the other conformation.

At decoding position t, the state-specific representations are combined as $\overline{\mathbf{h}}_t = \mathrm{Fuse}(\mathbf{h}_t^1, \mathbf{h}_t^2)$ and passed through the original output head, $p_t = \mathrm{Softmax}(W_{\mathrm{out}}\overline{\mathbf{h}}_t)$.

A single residue is selected and appended to the prefix shared by both streams. The locations of message passing and representation fusion are adapted to ProteinReasoner, ESM-IF, and ProteinMPNN, while the twin-stream and shared-prefix design remains unchanged.

3.3 CROSS-CONFORMATION MESSAGE PASSING

At interaction location l, each stream applies the shared backbone layer independently, then receives a gated, residue-aligned message from the other stream:

$$\begin{aligned} \widehat{\mathbf{H}}_s^{l+1} &= F_\theta^l(\mathbf{H}_s^l), \\ \mathbf{H}_s^{l+1} &= \widehat{\mathbf{H}}_s^{l+1} + sigma(g_l)\,\mathrm{Dropout}\left[M_l(\widehat{\mathbf{H}}_{\bar{s}}^{l+1})\right], \end{aligned} \tag{1}$$

where $s \in \{1, 2\}$, $\bar{s} = 3 - s$, and $M_l(x) = W_{\mathrm{up}}^l\,\mathrm{GELU}(W_{\mathrm{down}}^l\,\mathrm{LN}^l(x))$ is a bottleneck adapter. The adapter reduces the source representation to a low-dimensional bottleneck and restores the backbone hidden dimension; the scalar gate controls its residual contribution. Messages follow the residue correspondence and validity masks described in Appendix S1.

Both directional messages are computed before either stream is updated, ensuring synchronous and order-independent interaction. We zero-initialize W_{up}^l, so the added message path initially contributes zero and learns cross-conformation communication during post-training.

3.4 BACKBONE-SPECIFIC INSTANTIATIONS

Across all backbones, TwinMP uses two parameter-tied streams with a shared autoregressive prefix, backbone-specific bidirectional message passing, and hidden-state fusion before the original amino-acid output head. Detailed adapter dimensions, layer configurations, masking rules, and initialization settings are provided in Appendix S1.

TwinMP-ProteinReasoner. ProteinReasoner represents structure, MSA, and sequence as an ordered multimodal token sequence. The two streams receive different structure-token blocks while

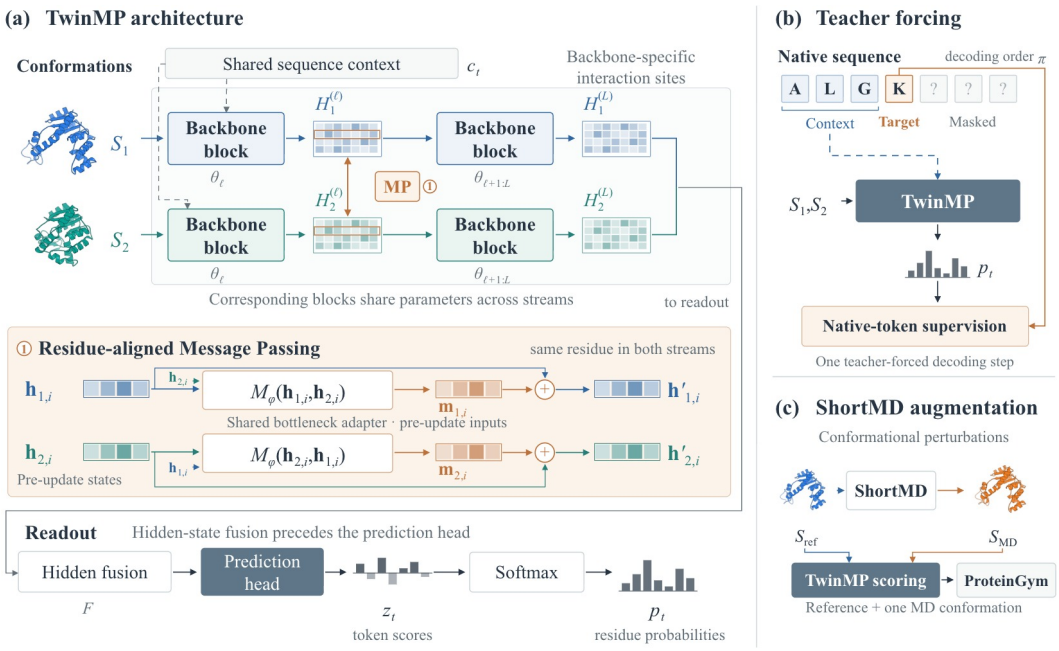


Figure 2: Overview of TwinMP. (a) Parameter-tied streams exchange residue-aligned messages through a shared adapter, followed by hidden-state fusion and readout. (b) Teacher-forced inverse-folding training uses native sequence context to predict the current residue. (c) Schematic of ShortMD augmentation: a reference structure and one NPT equilibration endpoint are paired for ProteinGym scoring.

sharing the same MSA block and autoregressive prefix. Bidirectional message adapters operate on the complete hidden sequence after Transformer layers, allowing cross-conformation information to propagate across modalities. The final sequence-token states are averaged and passed through the original ProteinReasoner sequence head.

TwinMP-ESM-IF. TwinMP-ESM-IF processes the two residue-aligned backbone coordinate sets with a shared geometric encoder while retaining separate conformation-specific encoder memories. Bidirectional messages are exchanged between aligned representations throughout the Transformer encoder. During decoding, both streams use the same sequence prefix but attend to their respective encoder memories. Their final decoder states are fused, refined, and projected through the original ESM-IF output head.

TwinMP-ProteinMPNN. TwinMP-ProteinMPNN constructs a separate residue graph for each conformation while sharing the feature-extraction, encoder, decoder, and output-head parameters. The streams retain distinct graph and decoding states but use the same decoding order and generated sequence prefix. Bidirectional messages are exchanged between aligned node states and corresponding edge states after graph-encoder layers. The final decoder states are averaged and passed once through the shared output head to produce a single amino-acid distribution.

3.5 Training Objective

TwinMP is post-trained on residue-aligned conformation pairs using teacher-forced sequence prediction. We minimize the conditional sequence loss

$$\mathcal{L} = -\sum_{t=1}^{L} \log p_{\theta,\phi}\left(y_t \mid \mathbf{X}_1, \mathbf{X}_2, \mathbf{y}_{<t}\right), \tag{2}$$

where θ denotes the pretrained backbone parameters and ϕ denotes the newly introduced cross-conformation parameters. Both the backbone parameters and the added modules are optimized. The

loss is computed over the common residue core shared by the two conformations. Backbone-specific loss settings and optimization hyperparameters are provided in Appendix S2.

3.6 Zero-Shot Mutation Effect Scoring

We evaluate TwinMP-ProteinReasoner on zero-shot mutation effect prediction using ProteinGym (Notin et al., 2023). No ProteinGym fitness labels are used for post-training or model selection. We pair the ProteinGym AF2 wild-type structure with a single MD equilibration endpoint (EQ), using these alternative structures only at inference. The AF2+AF2 control duplicates the primary structure in both streams; AF2+EQ replaces the second input. Both conditions use one TwinMP forward pass with two structural slots.

For each assay, the wild-type sequence and its precomputed native MSA profile are provided through teacher forcing.

At sequence position i, we add the sequence-head and MSA-head logits and normalize over the amino-acid vocabulary: $P_i = \mathrm{Softmax}(z_i^{\mathrm{seq}} + z_i^{\mathrm{MSA}})$.

For a variant v, we compute the WT-context sitewise score

$$s(v) = \sum_{i \in \mathcal{M}(v)} \Big[\log P_i(a_i^{\mathrm{mut}}) - \log P_i(a_i^{\mathrm{WT}}) \Big]. \tag{3}$$

Here, $\mathcal{M}(v)$ is the set of mutated positions, while a_i^{WT} and a_i^{mut} denote the wild-type and mutant amino acids at position i.

The context remains the wild-type sequence at every scored position; this additive score is not the full autoregressive likelihood ratio of the complete mutant and wild-type sequences.

Performance is measured by the Spearman correlation between $s(v)$ and the experimental fitness values within each assay. We report the official ProteinGym aggregation across assays and functional categories. Additional scoring and aggregation details are provided in Appendix S3.

4 Experiments

4.1 Experimental Setup

4.1.1 Post-Training Data and Homology-Disjoint Splits

We construct a shared post-training dataset from the CoDNaS conformational pairs released with DynamicMPNN (Abrudan et al., 2025). The same training and validation sets are used across all backbone families, enabling controlled comparison under a common post-training distribution.

We remove cross-split homology using both MMseqs2 sequence search and Foldseek structural search. For MMseqs2, we require at least 30% sequence identity and 80% bidirectional coverage. For Foldseek, we use the default 3Di+AA alignment mode with at least 30% alignment identity, 80% bidirectional coverage, and an E-value threshold of 0.01. Homology groups identified by either method are merged and assigned entirely to one split, yielding a *homology-disjoint split*.

The resulting dataset contains 32,342 training pairs, 4,256 validation pairs, and 4,999 pairs in the common test pool. We additionally include 9,985 single-conformation PDB structures that are filtered against the test pool using the same criteria. During dual-conformation training, each single-conformation example is duplicated across the two input streams. Dataset statistics are summarized in Table 1.

4.1.2 Backbone-Specific Decontamination

Because the three backbones were pretrained on different structure collections, we construct a separate decontaminated test subset for each model family from the common pool of 4,999 conformational pairs.

For ProteinReasoner, PDB-derived pretraining structures are filtered using MMseqs2 and Foldseek, while AlphaFoldDB-derived entries are removed by UniProt ID. This procedure retains 734 test pairs.

Table 1: Statistics of the shared post-training dataset. The common test pool is further filtered for backbone-specific decontamination.

Split	Conformation pairs	Single-conf. structures
Training	32,342	9,985
Validation	4,256	–
Test pool	4,999	–
Total	41,597	9,985

For ESM-IF, we filter against its publicly identifiable CATH-derived training structures using the same sequence and structure criteria. Because the AlphaFoldDB training list is unavailable, decontamination is limited to accessible training data. We use the 381 evaluable pairs within the 532-pair CATH-filtered set (TM-score < 0.4).

For ProteinMPNN, we apply the strictest filtering using its released training set. We remove matches at 30% sequence identity with 80% bidirectional coverage or query TM-score above 0.7. The TM-score threshold balances structural-overlap control with retaining a sufficiently large evaluation set, resulting in 93 test pairs.

All methods within each backbone family are evaluated on the same subset. Because the retained examples differ across families, absolute results are not directly comparable between backbone families.

4.1.3 Compared Methods

We evaluate TwinMP on ProteinReasoner-650M, ESM-IF, and ProteinMPNN. Within each backbone family, all trainable adaptations use the same post-training data and validation split, while all methods are evaluated on the same backbone-specific test set.

Single-state. The original pretrained model provides a single-conformation reference. ProteinReasoner uses state-1 conditioning for AAR; ProteinMPNN averages eight samples conditioned on each state. The structural sample uses the state-1 branch.

Pooling. The two conformation-specific structural representations are aggregated before sequence prediction. This baseline evaluates whether static representation-level fusion is sufficient to combine the two structural states.

Logit mean. The two conformations are processed independently, and their amino-acid logits are combined during autoregressive decoding. This inference-time baseline preserves separate single-state computations but introduces no interaction between their hidden representations.

Retrained DynamicMPNN. We retrain DynamicMPNN from scratch using its official architecture on the paired and single-conformation training data. Single-conformation examples are duplicated across its structural inputs. DynamicMPNN and TwinMP therefore use both data sources. The retrained model achieves performance comparable to the released checkpoint on the official test set (Appendix S2.2).

4.1.4 Sequence Generation and Structural Evaluation

All methods use top-p autoregressive sampling ($p = 0.9$, temperature 1.0). We generate 16 sequences per conformational pair. Sequence recovery is averaged over samples and valid residues, then over test pairs.

We use generated sample 0 and Boltz-2 (Passaro et al., 2025) with seed 42, conditioning on each target conformation as a template.

4.1.5 Evaluation Metrics

We evaluate generated sequences using amino-acid recovery and structural compatibility with both target conformations.

Amino-acid recovery. AAR is the fraction of generated residues matching the native sequence and is averaged over the 16 sampled sequences for each conformational pair.

Structural compatibility. We report mean pLDDT, TM-score (Zhang & Skolnick, 2004), and backbone RMSD over the two target-template-conditioned predictions; $\mathrm{TM}_{\min}$ is the lower of the two TM-scores. Pair-level measurements are then averaged over each evaluation cohort. These metrics measure computational compatibility under target-template conditioning.

4.2 TwinMP Adapts Multiple Inverse-Folding Backbones

Table 2 reports five-run amino-acid recovery (seeds 42–46). Trainable methods use independent training and evaluation seeds; fixed pretrained methods repeat sampling only. We report the mean and sample standard deviation across run-level results. For trainable methods, the standard deviation includes both training and evaluation randomness.

Table 2: Five-run AAR, reported as mean ± sample standard deviation. Fine-tuned methods use independent training and evaluation repeats; other methods use fixed-model sampling repeats. Each cell has complete pair coverage in all five runs.

Method	Fine-tuned	PR (734)	ESM-IF (381)	ProteinMPNN (93)
Single-state	No	0.2610 ± 0.0010	0.2748 ± 0.0011	0.2815 ± 0.0012
Pooling	Yes	0.2972 ± 0.0005	0.3815 ± 0.0006	0.3327 ± 0.0019
Logit mean	No	0.2860 ± 0.0003	0.3269 ± 0.0007	0.2837 ± 0.0019
TwinMP w/o MP	Yes	0.3102 ± 0.0010	0.3722 ± 0.0016	0.3434 ± 0.0020
TwinMP	Yes	$\mathbf{0.3208} \pm 0.0005$	$\mathbf{0.3836} \pm 0.0005$	$\mathbf{0.3436} \pm 0.0009$

For ProteinReasoner, TwinMP achieves mean AAR 0.3208, improving over pooling at 0.2972 and TwinMP without message passing at 0.3102. For ESM-IF, the corresponding values are 0.3836, 0.3815, and 0.3722. For ProteinMPNN, full TwinMP and no-MP have similar mean recovery (0.3436 versus 0.3434). Thus, TwinMP improves recovery over pooling and Logit mean across all three backbones, while the additional benefit of MP is backbone dependent.

4.3 Structural Consistency

We report structural evaluation on the seed-42 checkpoint used as the primary run.

Table 3 shows backbone-dependent improvements in structural consistency. For ProteinReasoner, TwinMP achieves the best pLDDT, $\mathrm{TM}_{\mathrm{avg}}$, and $\mathrm{RMSD}_{\mathrm{avg}}$, while its $\mathrm{TM}_{\min}$ remains comparable to the best result. For ESM-IF, TwinMP obtains the lowest $\mathrm{RMSD}_{\mathrm{avg}}$; Pooling performs better on the two TM-score metrics, and Logit mean has the highest pLDDT. For ProteinMPNN, TwinMP achieves the best values on all four metrics. These results complement sequence recovery by assessing compatibility with both target conformations.

4.4 ProteinGym Zero-Shot Mutation Effect Prediction

We evaluate TwinMP-ProteinReasoner-150M on 214 ProteinGym substitution assays covering 186 proteins, excluding three overlength P53 assays. All variants are scored using wild-type teacher forcing and the combined sequence-head and MSA-head logits. The Single baseline conditions on the primary structure. For TwinMP, AF2+AF2 duplicates the ProteinGym AF2 wild-type structure, whereas AF2+EQ pairs it with one NPT equilibration endpoint. The two TwinMP conditions use the same final-epoch checkpoint (epoch 9; training seed 42), effective records, and scoring procedure, with two structural slots and one forward pass. MD structures are used only at inference. No ProteinGym fitness labels are used for training, checkpoint selection, or conformation selection.

Table 4 reports an official Average of 0.473 for the Single baseline, 0.479 for TwinMP with duplicated AF2 inputs, and 0.486 for TwinMP with AF2+EQ. The Single baseline retains the highest Expression score, while AF2+EQ achieves the highest scores in the other four categories.

At the same TwinMP checkpoint, replacing the duplicated second input with EQ improves all five functional-category scores. The paired gain computed from unrounded assay correlations is

Table 3: Structural consistency on the backbone-specific test sets. Bold and underline indicate the best and second-best results within each block, respectively. pLDDT is on the 0–1 scale; RMSD is in Å. For ProteinReasoner, evaluated coverage is 733/734 pairs except Logit mean (732/734) and DynamicMPNN (728/734); ESM-IF and ProteinMPNN have complete coverage.

Method	pLDDT ↑	TM_{avg} ↑	$RMSD_{avg}$ ↓	TM_{min} ↑
ProteinReasoner: 734 test pairs				
Single-state	0.7303	0.3963	14.7717	0.3616
Pooling	0.7511	0.5065	7.9209	0.4663
Logit mean	<u>0.7609</u>	0.3780	16.6642	0.3438
TwinMP w/o MP	0.7569	<u>0.5111</u>	<u>7.8381</u>	**0.4675**
TwinMP	**0.7644**	**0.5146**	**7.7215**	<u>0.4672</u>
DynamicMPNN	0.6799	0.4744	8.6393	0.4354
ESM-IF: 381 test pairs				
Single-state	0.8211	0.4566	6.5844	0.4347
Pooling	0.8405	**0.5008**	<u>5.7893</u>	**0.4800**
Logit mean	**0.8524**	0.4926	5.8262	<u>0.4741</u>
TwinMP w/o MP	0.8353	0.4915	5.8296	0.4656
TwinMP	<u>0.8428</u>	<u>0.4947</u>	**5.7501**	0.4688
DynamicMPNN	0.7549	0.3871	7.8665	0.3619
ProteinMPNN: 93 test pairs				
Single-state	0.8425	0.6903	<u>4.2200</u>	0.6717
Pooling	0.8384	0.6744	4.5210	0.6566
Logit mean	0.8417	0.6840	4.5616	0.6690
TwinMP w/o MP	<u>0.8480</u>	<u>0.6918</u>	4.3859	<u>0.6773</u>
TwinMP	**0.8531**	**0.6998**	**4.1469**	**0.6867**
DynamicMPNN	0.7628	0.5683	5.9602	0.5513

Table 4: Zero-shot ProteinGym mutation-effect prediction with ProteinReasoner-150M. Values are Spearman correlations aggregated using the official ProteinGym procedure; Average equally weights the five functional categories. EQ denotes one MD equilibration endpoint. Bold indicates the highest reported value in each column.

Method	Input	Average	Activity	Binding	Expression	Org. Fitness	Stability
Single baseline	Primary	0.473	0.466	0.380	**0.494**	0.436	0.590
TwinMP	AF2+AF2	0.479	0.463	0.423	0.475	0.436	0.600
TwinMP	AF2+EQ	**0.486**	**0.468**	**0.433**	0.478	**0.446**	**0.604**

0.006147 (protein-cluster bootstrap 95% CI [0.003886, 0.008350]), with improvements on 159 of 214 assays. These results support the utility of an MD-derived alternative structural input for zero-shot mutation-effect prediction.

5 Limitations and Conclusion

TwinMP requires aligned conformations, and backbone-specific cohorts limit cross-family comparisons. Template-conditioned predictions do not establish physical occupation of both states. The ProteinGym control uses one checkpoint and one MD replica per protein, leaving message-passing and relaxation effects to further matched controls.

TwinMP adapts pretrained inverse-folding models through conformation-specific streams and bidirectional exchange. Experiments on three backbones demonstrate its applicability, with backbone-dependent gains from message passing. A matched ProteinGym comparison further shows that an MD-derived alternative input improves mutation-effect prediction.

AI USE STATEMENT

Generative AI (OpenAI Codex) assisted with formatting, writing and editing, schematic preparation, method discussion, and analysis and presentation of aggregate results. It also assisted with citation and compilation checks. The authors take responsibility for the final manuscript and verification of all results.

REFERENCES

Alex Abrudan, Sebastian Pujalte Ojeda, Chaitanya K. Joshi, Matthew Greenig, Felipe Engelberger, Alena Khmelinskaia, Jens Meiler, Michele Vendruscolo, and Tuomas P. J. Knowles. Multi-state protein design with DynamicMPNN, 2025. URL https://arxiv.org/abs/2507.21938.

Jiaren Cao, Fangfang Fan, Changjiang Lyu, Sheng Hu, Weirui Zhao, Jiaqi Mei, Shuai Qiu, Lehe Mei, and Jun Huang. Pocket modification of ω-amine transaminase AtATA for overcoming the trade-off between activity and stability toward 1-acetonaphthone. *Engineering*, 30:203–214, 2023. doi: 10.1016/j.eng.2023.04.009. URL https://doi.org/10.1016/j.eng.2023.04.009.

Wei Chen, Chi Song, Liang Leng, Sanyin Zhang, and Shilin Chen. The application of artificial intelligence accelerates G protein-coupled receptor ligand discovery. *Engineering*, 32:18–28, 2024. doi: 10.1016/j.eng.2023.09.011. URL https://doi.org/10.1016/j.eng.2023.09.011.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Alexis Courbet, Rob J. de Haas, Neville Bethel, Philip J. Y. Leung, Tim F. Huddy, Sam Pellock, Doug Tischer, Frederick Chan, Brian Koepnick, Hannah Nguyen, Alex Kang, Banumathi Sankaran, Asim K. Bera, Neil P. King, and David Baker. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187.

Diego del Alamo, Davide Sala, Hassane S. Mchaourab, and Jens Meiler. Sampling alternative conformational states of transporters and receptors with AlphaFold2. *eLife*, 11:e75751, 2022. doi: 10.7554/eLife.75751.

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.

Jingjing Guo and Huan-Xiang Zhou. Protein allostery and conformational dynamics. *Chemical Reviews*, 116(11):6503–6515, 2016. doi: 10.1021/acs.chemrev.5b00590.

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 *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8946–8970. PMLR, 2022. URL https://proceedings.mlr.press/v162/hsu22a.html.

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

Giacomo Janson, Gilberto Valdes-Garcia, Lim Heo, and Michael Feig. Direct generation of protein conformational ensembles via machine learning. *Nature Communications*, 14:774, 2023. doi: 10.1038/s41467-023-36443-x.

Chaozhong Liu, Linlin Chao, Shaomin Ji, Hao Wang, Taorui Jiang, Zhangyang Gao, Yucheng Guo, Ming Yang, and Xiaoming Zhang. ProteinReasoner: A multi-modal protein language model with chain-of-thought reasoning for efficient protein design. *bioRxiv*, 2025. doi: 10.1101/2025.07.21.665832. URL https://www.biorxiv.org/content/10.1101/2025.07.21.665832v2.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29287–29303. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f51338d736f95dd42427296047067694-Abstract.html.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems*, volume 36, pp. 64331–64379. Curran Associates, Inc., 2023. doi: 10.52202/075280-2810. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/cac723e5ff29f65e3fcbb0739ae91bee-Abstract-Datasets_and_Benchmarks.html.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, David Kwabi-Addo, Dominique Beaini, Tommi Jaakkola, and Regina Barzilay. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, 2025. doi: 10.1101/2025.06.14.659707. URL https://doi.org/10.1101/2025.06.14.659707.

Florian Praetorius, Philip J. Y. Leung, Maxx H. Tessmer, Adam Broerman, Cullen Demakis, Acacia F. Dishman, Arvind Pillai, Abbas Idris, David Juergens, Justas Dauparas, Xinting Li, Paul M. Levine, Mila Lamb, Ryanne K. Ballard, Stacey R. Gerben, Hannah Nguyen, Alex Kang, Banumathi Sankaran, Asim K. Bera, Brian F. Volkman, Jeff Nivala, Stefan Stoll, and David Baker. Design of stimulus-responsive two-state hinge proteins. *Science*, 381(6659):754–760, 2023. doi: 10.1126/science.adg7731. URL https://doi.org/10.1126/science.adg7731.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. SaProt: Protein language modeling with structure-aware vocabulary. *bioRxiv*, 2023. doi: 10.1101/2023.10.01.560349. URL https://doi.org/10.1101/2023.10.01.560349.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. *arXiv preprint arXiv:2402.18567*, 2024. URL https://arxiv.org/abs/2402.18567.

Guanghong Wei, Wenhui Xi, Ruth Nussinov, and Buyong Ma. Protein ensembles: How does nature harness thermodynamic fluctuations for life? the diverse functional roles of conformational ensembles in the cell. *Chemical Reviews*, 116(11):6516–6551, 2016. doi: 10.1021/acs.chemrev.5b00562.

Ren Wei and Uwe T. Bornscheuer. New biocatalytic approaches for plastic depolymerization. *Engineering*, 58:27–31, 2026. doi: 10.1016/j.eng.2025.11.017. URL https://doi.org/10.1016/j.eng.2025.11.017.

Kai Yi, Kiarash Jamali, and Sjors H. W. Scheres. All-atom inverse protein folding through discrete flow matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72244–72257. PMLR, 2025. URL https://proceedings.mlr.press/v267/yi25a.html.

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: 10.1002/prot.20264. URL https://doi.org/10.1002/prot.20264.

Zuobai Zhang, Pascal Notin, Yining Huang, Aurélie Lozano, Vijil Chenthamarakshan, Debora Marks, Payel Das, and Jian Tang. Multi-scale representation learning for protein fitness prediction. *arXiv preprint arXiv:2412.01108*, 2024. URL https://arxiv.org/abs/2412.01108.

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 of *Proceedings of Machine Learning Research*, pp. 42317–42338. PMLR, 2023. URL https://proceedings.mlr.press/v202/zheng23a.html.

S1 BACKBONE-SPECIFIC IMPLEMENTATION

Adapter and fusion. ProteinReasoner and ESM-IF use layer-specific bottleneck messages

$$M_\ell(x) = W^\ell_{\mathrm{up}}\,\mathrm{GELU}\big(W^\ell_{\mathrm{down}}\,\mathrm{LN}_\ell(x)\big), \tag{S1.1}$$

with dropout on the gated residual message. The output projection and bias are initialized to zero; the scalar gate is initialized to zero. Adapters are independent across layers and shared between directions within a layer. ProteinMPNN has separate node and edge adapters, with gate initialization -2, dropout 0.1, and zero-initialized output projections. Its edge messages match directed residue pairs across graphs. ESM-IF additionally applies a zero-initialized, ungated residual refinement after decoder-hidden averaging.

Table S1: Backbone-specific implementation settings. Layer indices are zero based; D and B are hidden and bottleneck dimensions.

Backbone	D/B	Exchanged state	Message layers	Final fusion
ProteinReasoner-650M	1280/128	Complete multimodal hidden	Transformer 0–32	Sequence-hidden mean
ProteinReasoner-150M	640/128	Complete multimodal hidden	Transformer 0–29	Sequence-hidden mean
ESM-IF	512/128	Residue encoder hidden	Encoder 0–7	Decoder-hidden mean + refinement
ProteinMPNN	128/64	Node and matched-edge hidden	Encoder 0–2	Decoder-hidden mean

ProteinReasoner packs conformation-specific structure tokens, shared MSA inputs, and the common sequence prefix. Messages operate on the entire packed hidden sequence; only sequence states are fused before the original sequence head. Its inverse-folding experiments set MSA inputs and loss masks to zero. ESM-IF shares the GVP and Transformer encoders but retains separate memories and decoder caches. Its refinement has bottleneck dimension 128. ProteinMPNN uses the pretrained `v_48_020` architecture with separate $k = 48$ neighbor graphs, three encoder layers, and three decoder layers. ESM-IF and ProteinMPNN have no cross-conformation decoder messages. All models retain the original amino-acid output head.

S1.1 ALIGNMENT AND MISSING COORDINATES

ProteinReasoner uses Needleman–Wunsch alignment for each pair and removes columns containing a gap or unknown residue. A token/sequence length mismatch excludes the pair. The ESM-IF and ProteinMPNN loaders use the identity map when lengths match and Needleman–Wunsch alignment otherwise; equal-length inputs therefore rely on an already corresponding residue register. Each stream retains its own coordinates, tokens, and graph topology after restriction to the common core.

ESM-IF zero-fills missing coordinates and assigns confidence -1; the corresponding sequence position remains a prediction target. ProteinMPNN requires finite N, Cα, C, and O atoms in both conformations. Its shared validity mask applies to graph and message updates, decoding, and loss; edge transfer additionally requires both endpoint masks and the same directed residue pair in the other graph. Padding is introduced after alignment and handled by backbone masks. The implementations process one selected chain per conformation; multi-chain correspondence is outside this input contract.

S2 TRAINING DETAILS

TwinMP and its post-trained controls initialize from their respective pretrained backbones. Backbone parameters, output heads, and added adapters are optimized jointly, with tied parameters counted once. ProteinReasoner and ESM-IF use cross-entropy; ProteinMPNN uses label smoothing 0.1. Table S2 summarizes the primary-run recipes.

The settings use seed 42, maximum lengths 1024 for PR and 500 for ESM-IF/ProteinMPNN, and BF16 for PR/ProteinMPNN versus FP32 for ESM-IF. The documented schedules are constant for PR/ProteinMPNN and cosine for ESM-IF. Parameter tying avoids duplicate backbone weights, while retaining two streams requires separate computations and activations.

Table S2: Post-training settings. Batch size is per GPU. PR denotes ProteinReasoner; the ESM-IF row describes Pooling and full TwinMP, not the no-MP variant.

Model	Optimizer	Learning rate	Weight decay	Batch	Accumulation	GPUs	Epochs
PR-650M / PR-150M	AdamW	3×10^{-5}	0.01	1	8	2	10
ESM-IF	AdamW	10^{-4}	0.01	2	8	1	10
ProteinMPNN full	Adam	10^{-4}	0	8	1	2	10
ProteinMPNN Pooling / no-MP	Adam	10^{-4}	0	2	4	2	10

DynamicMPNN. DynamicMPNN is trained from scratch rather than initialized from the released checkpoint. Like TwinMP, it uses both paired and single-conformation training data; each single-conformation example is duplicated across its structural inputs. It uses aligned multi-conformation clusters, training $k = 3$ with a duplicated conformation for two-state clusters, the `ca_bb` representation, and KNN-32 graphs. The configuration has 8 encoder, 4 autoregressive decoder, and 4 pooled-encoder layers. Optimization uses Adam at 10^{-3}, zero weight decay, no scheduler, label smoothing 0.05, and 100 epochs. The evaluated checkpoint is epoch 99. Both methods use the paired and single-conformation data sources, while initialization and optimization differ.

S2.1 DynamicMPNN baseline

We train the single-chain multi-conformation DynamicMPNN variant from scratch using both paired and single-conformation training data. No released DynamicMPNN checkpoint is used for initialization. Conformational sequences are aligned using Needleman–Wunsch alignment, retaining columns present in all conformations and containing no unknown residues. We require at least 10 residues in the aligned core and exclude chains longer than 500 residues.

The model uses `single_chain_k` pooling with three structural inputs during training. For two-conformation examples, one conformation is duplicated; single-conformation examples are repeated across all three inputs. Table S3 summarizes the architecture and optimization settings.

Table S3: DynamicMPNN architecture and training configuration. Batch size is specified per GPU.

Setting	Value
Structural representation	`ca_bb`
Graph neighborhood	32 nearest neighbors
Encoder / AR decoder / pooled encoder layers	8 / 4 / 4
Node hidden channels	128 scalar, 16 vector
Edge hidden channels	32 scalar, 4 vector
Dropout / coordinate noise	0.5 / 0.1
Optimizer	Adam
Learning rate / weight decay	10^{-3} / 0
Learning-rate schedule	Constant
Objective	Cross-entropy, label smoothing 0.05
Precision	BF16 mixed precision
GPUs	2, DDP with synchronized BatchNorm
Batch size / gradient accumulation	2 / 4
Effective batch size	16
Training epochs	100
Validation frequency	Every epoch
Early stopping	None
Evaluated checkpoint	Final epoch (index 99)

DynamicMPNN and TwinMP use both paired and single-conformation data sources, but differ in initialization, architecture, and optimization. Their comparison therefore evaluates the complete training approaches rather than isolating the effect of cross-conformation message passing.

S2.2 Comparison with the Released DynamicMPNN Checkpoint

We compare the released and retrained DynamicMPNN checkpoints, which share the same architecture, on the official 100-pair test set. Both use two input conformations, sampling temperature 0.1, and three generated sequences per pair. Structural evaluation uses the first generated sequence and the template-conditioned Boltz2 protocol described in the main text.

Table S4: Released and retrained DynamicMPNN on the official 100-pair test set. Differences are retrained minus released.

Metric	Released	Retrained	Difference
Mean AAR	0.3427	0.3576	$+0.0149$
Token-weighted AAR	0.3508	0.3600	$+0.0092$
Two-state mean TM-score	0.6996	0.6825	-0.0172
Cα RMSD (Å)	5.236	5.582	$+0.346$

S3 ProteinGym Evaluation Details

Structures and checkpoint. The starting structures are ProteinGym AF2 wild-type predictions cropped to the assay sequence. The ShortMD preparation uses GROMACS, Amber14sb, TIP3P water, and 300 K, with one replica per protein. Energy minimization is followed by 200 ps NVT and 2 ns NPT equilibration. EQ is the final NPT structure, rather than a frame selected from a production trajectory. No endpoint selection uses DMS scores. All 214 included assays use an actual EQ structure, with no duplicated-AF2 fallback.

The PR-150M model has 30 layers, hidden dimension 640, all-layer message passing, bottleneck 128, dropout 0.1, and mean hidden-state fusion. We use the final epoch-9 checkpoint from the ten-epoch run with seed 42. Training uses the conformational-pair dataset, with input probabilities 0.5 for distinct pairs and 0.25 for each duplicated state. The ShortMD structures are used only for inference.

MD-derived conformations. We prepare one alternative conformation per protein from the ProteinGym AF2 wild-type structure cropped to the assay sequence. Simulations use GROMACS 2026.1 with the Amber14SB force field and TIP3P water. Each protein is centered in a dodecahedral box with a minimum solute-to-box distance of 1.2 nm. Sodium and chloride ions are added to neutralize the system at a salt concentration of 0.15 M.

Steepest-descent energy minimization is followed by 200 ps of NVT and 2 ns of NPT equilibration. We use a 2 fs integration time step, constrain bonds involving hydrogen atoms, and maintain the temperature at 300 K using the V-rescale thermostat. NPT equilibration uses isotropic Berendsen pressure coupling. Each protein is simulated with one replica.

For AF2+EQ evaluation, the second structural input is the final NPT equilibration structure. Subsequent production-trajectory frames are not used, and no structure selection uses DMS scores.

Scoring and aggregation. Both conditions use WT teacher forcing and sequence-head plus MSA-head logits, without autocast. Effective-record counts match within every assay, totaling 2,443,366 per condition. The official aggregation averages assays within protein–function groups, then proteins within each function, and finally the five functions equally. It differs from the unweighted assay mean. Official table values retain the export's three-decimal precision; the precise paired difference is recomputed from unrounded assay correlations using the same hierarchy.

Paired uncertainty. We resample 186 UniProt protein clusters with replacement, preserving all assays, category memberships, and both conditions for each sampled protein. We recompute the hierarchical difference for 100,000 valid draws (statistical seed 42), replacing two draws that lack a category, and report the percentile 95% interval. The assay-macro difference is 0.006261, with cluster interval [0.004073, 0.008306]. There are 159 positive and 55 negative assay differences. These intervals condition on the fixed model and endpoint set and do not estimate training-seed or MD-replica variation.