Two Stages of Folding: Convergent Mechanisms in AI Protein Folding Trunks

Kevin Lu[1*‡] **Jannik Brinkmann**[12†] **Stefan Huber**[3†] **Aaron Mueller**[4]
Yonatan Belinkov[35] **David Bau**[1†] **Chris Wendler**[1*‡]

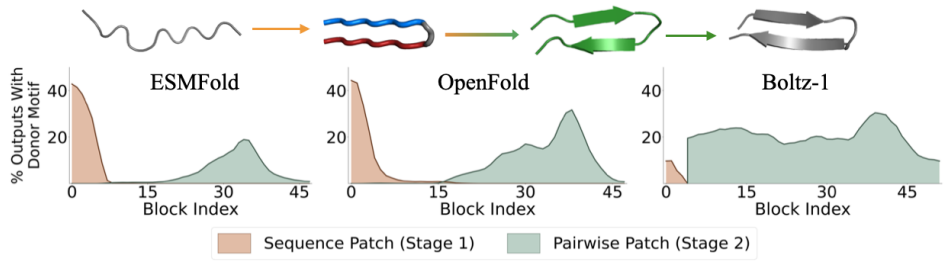


Figure 1: **Two computational stages in AI protein folding trunks.** We identify which latent representations in ESMFold, OpenFold, and Boltz-1 influence secondary structure formation by patching activations from a hairpin protein into a helical protein at each block of the trunk, then measuring whether the output folds as a hairpin (or vice versa). Sequence patches (orange) induce hairpin formation in early blocks; pairwise patches (green) are effective in later blocks.

Abstract

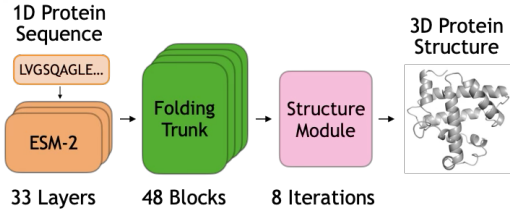
How do protein structure prediction models fold proteins? We investigate this question through causal interventions on the folding trunks of ESMFold, OpenFold, and Boltz-1. Across all three models, we find a shared two-stage computational structure for certain structural decisions. In the first stage, early blocks initialize *pairwise biochemical signals*: features like charge propagate from sequence into pairwise representations through architecture-specific pathways. In the second stage, late blocks develop *pairwise spatial features*: distance and contact information accumulate in the pairwise representation. We support this mechanistic account with causal evidence, showing that steering charge and distance features induces predictable structural changes. Furthermore, these representations support partial functional transfer: pairwise states can be linearly aligned and substituted across models. Together, these results suggest that folding trunks with different architectures, inputs, and training procedures converge on a shared representational organization for mapping sequence chemistry into spatial geometry.

1 Introduction

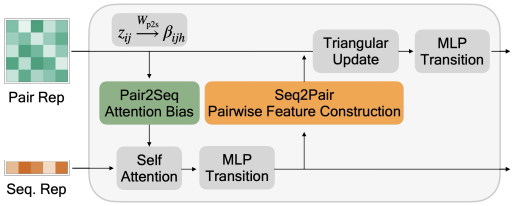
Have neural protein folding models like AlphaFold2 [Jumper et al., 2021] truly learned the physical principles of how proteins fold? Our community has studied these models extensively as predictors, with benchmarks, scaling curves, and downstream applications, but comparatively little on what they have figured out about folding itself. We ask whether physically interpretable decisions can be identified within the internal computations of these models, conducting the first cross-architectural mechanistic account of three modern folding models: ESMFold [Lin et al., 2023], OpenFold [Ahdritz et al., 2024], and Boltz-1 [Wohlwend et al., 2024].

[1]Northeastern University [2]TU Clausthal [3]Harvard University [4]Boston University [5]Technion – Israel Institute of Technology [*]Lead author. [†]Core contributor. [‡]Corresp. authors: lu.kev@northeastern.edu, ch.wendler@northeastern.edu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) **ESMFold.** A protein language model (ESM-2) encodes an amino acid sequence into an initial sequence representation; the pairwise representation is initialized with learned positional embeddings. The folding trunk iteratively refines both representations over 48 blocks (consisting of multiple layers each). The structure module converts these into 3D coordinates for each residue. Adapted from [Lin et al., 2023].



(b) **Folding block.** *A sequence update* consisting of `Pair2Seq` and a sequence transformer layer (attention plus MLP) followed by a *pairwise update* consisting of `Seq2Pair` and multiplicative and attention based updates of the pairwise representation (triangular update, see [Jumper et al., 2021]) plus MLP. Adapted from [Lin et al., 2023].

Figure 2: ESMFold architecture and folding block.

We begin with alpha helices and beta hairpins, using motif transfer as a controlled way to change a model's structural decision. Hairpins are among the simplest motifs requiring coordination between separated strands [Muñoz et al., 1998, Blanco et al., 1998], making them a natural starting point for tracing how sequence information becomes geometry. We then extend the analysis to four-strand sheets, contacts between distant sequence regions, and pathway measurements across 1,000 diverse proteins. We ask three questions:

***When* does a folding model commit to a structural decision?** Using activation patching [Vig et al., 2020, Meng et al., 2022] we localize the computation that determines whether a region folds as a hairpin or as a helix, identifying a two-stage structure shared across all three architectures: an early sequence-driven commitment phase and a late pairwise-refinement phase. In Boltz-1, the early phase is compressed into the upstream MSA module, followed by a broad window of pairwise influence.

***What* features drive each stage?** Using linear probing and representation steering, we find that stage 1 writes biochemical signals (like charge and hydrophobicity related information) into pairwise space. Steering these features induces strand attraction, repulsion, and even solvent exposure. Stage 2 develops pairwise representations as approximate distance maps, and steering them induces contacts and hairpin-like structure.

Do *different* architectures share an internal organization? Using linear projections between pairwise representations of pairs of models, we find that representations from one model are partially functionally transplantable: states from one model can drive structural changes in another at the corresponding stage of computation. Folding trunks trained with different recipes, encoders, and decoders converge on a shared internal organization for mapping sequence chemistry into spatial geometry.

2 AI Protein Folding Background

For background on protein structure (amino acids, secondary structure motifs like alpha helices and beta hairpins), see Appendix A and Brändén and Tooze [1999].

We provide background on ESMFold as an example (Fig. 2a), which consists of three modules. (1) A protein language model (ESM-2 [Lin et al., 2023]) maps an amino acid sequence of length L to an initial sequence representation $s^{(0)} \in \mathbb{R}^{L \times d_s}$. (2) A *folding trunk* iteratively refines s alongside a pairwise representation $z \in \mathbb{R}^{L \times L \times d_z}$ (initialized from learned positional embeddings) over $K{=}48$ blocks. (3) A structure module based on invariant point attention [Jumper et al., 2021] decodes the final $(s^{(K)}, z^{(K)})$ into 3D coordinates. We focus on the folding trunk.

Folding trunk. Each of the $K{=}48$ blocks (Fig. 2b) applies a *sequence update* followed by a *pairwise update*, with residual connections throughout: $(s^{(k)}, z^{(k)}) = g^{(k)}(s^{(k-1)}, z^{(k-1)})$.

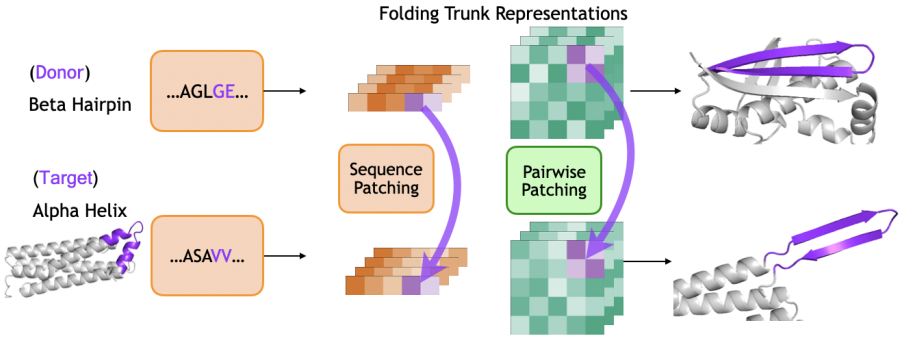


Figure 3: **Activation patching setup.** We extract donor hairpin representations (top) and replace the aligned target motif's sequence states (orange), pairwise states (green), or both during a selected trunk block. Purple marks the motif. We measure whether the output adopts the donor motif; the helix-donor experiment reverses the motif roles.

Sequence update. Self-attention is modulated by a `pair2seq` bias derived from z:

$$A_{ij}^{(h)} = \frac{\langle q_i^{(h)}, k_j^{(h)} \rangle}{\sqrt{d_h}} + \beta_{ijh}(z_{ij}), \tag{1}$$

where $\beta_{ijh}(z_{ij}) = (W_\beta z_{ij})_h$ is a learned linear projection producing one scalar bias per attention head.

Pairwise update. A `seq2pair` operation constructs pairwise features from sequence representations:

$$\phi_{ij} = \left[u_i \odot v_j \; ; \; u_i - v_j \right], \tag{2}$$

where $u_i = W_u s_i$ and $v_j = W_v s_j$ are linear projections of s. A learned projection maps these features into an update to z, which is then refined via triangular multiplicative updates and attention [Jumper et al., 2021].

OpenFold and Boltz-1. OpenFold [Ahdritz et al., 2024] is a faithful reimplementation of AlphaFold2 with a 48-block Evoformer trunk. Boltz-1 [Wohlwend et al., 2024] is an AlphaFold-3-style architecture with an MSA module followed by a 48-block Pairformer. Both use multiple sequence alignments (MSAs) and maintain sequence and pairwise representations analogous to ESMFold's. ESMFold and OpenFold exchange information in both directions throughout the trunk (`seq2pair` and `pair2seq` in every block). Boltz-1 confines `seq2pair` to its upstream MSA module (which we treat as blocks 0–3 of the trunk). Within the remaining Pairformer blocks, the only cross-representation pathway is `pair2seq`: sequence representations are updated by pair-biased self-attention and an MLP, but never write back into the pairwise track.

3 When Does the Model Fold a Hairpin or Helix?

Activation patching setup. We use activation patching to find the blocks at which replacing internal states changes the predicted motif. We select a *donor protein* containing a secondary structure motif (a beta hairpin or alpha-helical region) and a *target protein* containing the opposite motif. We run both proteins through the folding trunk, extracting the sequence representation $s^{(k)}$ and pairwise representation $z^{(k)}$ at each block k. During the target's forward pass, we replace representations in the target's motif region with the donor's. We then observe whether the output structure contains the donor motif in the patched region (Fig. 3). We disable recycling to isolate a single pass through the trunk (App. G).

Dataset construction. We selected 200 target proteins and approximately 130,000 donor motif regions from the PDB [Berman et al., 2000]. Using DSSP [Kabsch and Sander, 1983], we identified motif regions in each target, sampled 10 donors per motif, and aligned motif positions to define 15–20 residue intervention regions, yielding ≈10,000 patching experiments (App. F).

Full patching establishes feasibility. We first verify that patching can induce donor motif formation by patching both sequence and pairwise representations across all 48 blocks of the folding trunk.

Approximately 40% of patches (≈4,000 cases) successfully produce the donor motif in the target region, as measured by DSSP secondary structure assignment (App. D). In ESMFold, patching at the encoder or structure module is substantially less effective (App. H.1), identifying the trunk as the most effective site for motif transfer in this comparison.

Single-block patching reveals two regimes across architectures. To localize the computation, we repeat the patching experiment but intervene at a single block, patching either sequence or pairwise representations alone (restricted to the ≈4,000 donor–target pairs where full patching was successful). Sequence patches are most effective early, while pairwise patches become effective later in ESMFold and OpenFold (Fig. 1). Repeating the analysis over all donor–target pairs lowers success rates but preserves the early/late pattern (App. M). We patch sequence positions in the aligned motif and, by default, pairwise entries with both indices in that region. A broader "touch" mask extends the effective pairwise window (App. H.2). We report the effective windows for each mask, since the patch must also integrate with the surrounding structure.

Structural quality and scope. We evaluate intervention success alongside chain integrity, clashes, confidence, design/refold self-consistency TM-score (scTM), and motif recovery (App. N, Table 4). Filtering invalid or non-designable outputs lowers success but preserves the localization. After filtering, Boltz-1 pairwise patches remain effective from blocks 3–47, covering most of the analyzed network rather than a narrow late interval.

The ordering also appears in four-strand-sheet patches spanning 40–60 residues, long-range contacts with median sequence separation 132 residues, and length-stratified tests up to 1,000 residues (App. M). Long-range pairwise patches induce contacts in 46–73% of cases, versus 6–9% for norm-matched random perturbations. We check whether the new contacts match the donor and whether the rest of the protein contracts, to distinguish targeted transfer from global collapse. Across 1,000 diverse proteins, motif-independent sequence-to-pair updates peak at blocks 1–2, before pair-to-sequence updates at blocks 11–34. Together, these experiments extend the two-stage organization from local motif transfer to larger structures, long-range interactions, and whole-protein information flow.

4 Early Blocks: Building Pairwise Chemistry

4.1 Sequence Information Flows Into Pairwise Space

Why is sequence patching only effective in early blocks? We hypothesize that early blocks serve a specific role: transferring biochemical information from sequence representations into pairwise representations, where it remains accessible for downstream geometric computation. We present the analysis in detail for ESMFold using 400 successful cases of block-0 sequence patching, and report parallel results for OpenFold and Boltz-1 in App. I.

Within the ESMFold folding trunk, sequence representations s and pairwise representations z interact through two pathways (Figure 2b). The `seq2pair` pathway converts the sequence representation into a pairwise update via Eq. (2). The `pair2seq` pathway projects z into a scalar bias for each attention head and residue pair, which modulates sequence self-attention, via Eq. (1).

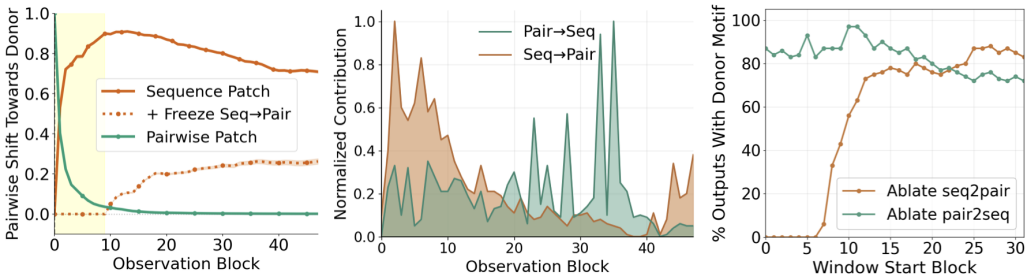
Representation similarity across layers. To quantify how much the patched representation resembles the donor versus the target, we compute an interpolation coefficient:

$$\alpha = \frac{\langle z_{\text{patched}} - z_{\text{target}}, z_{\text{donor}} - z_{\text{target}} \rangle}{\| z_{\text{donor}} - z_{\text{target}} \|^2} \tag{3}$$

Here α measures how far the patched state moves from the target toward the donor along the donor–target direction. Values of 0 and 1 match the target and donor on this axis; the states can still differ in other directions.

After sequence patching at block 0, z rapidly becomes donor-like within the first ≈10 blocks, then changes only gradually (Fig. 4a). Blocking `seq2pair` during blocks 0–10 prevents this shift entirely, while directly patching z at block 0 does not persist: the pairwise state returns toward the target trajectory. This confirms a short early "write-in" window during which sequence information is consolidated into z via `seq2pair`.

Pathway Contributions. The asymmetric importance of `seq2pair` and `pair2seq` across blocks is also visible without intervention. We measure the relative contribution of each pathway at every block



(a) (b) (c)

Figure 4: **Information flows from sequence into pairwise representations during a short early window.** (ESMFold; replicated for OpenFold and Boltz-1 in App. I) **(a)** Shift in z during the target forward pass under three interventions: sequence patching at block 0, sequence patching with `seq2pair` frozen during blocks 0–10, and pairwise patching at block 0. Sequence patching shifts z toward the donor (Eq. (3)), but freezing `seq2pair` during the early window prevents this shift. **(b)** Normalized contribution magnitudes of `seq2pair` (orange) and `pair2seq` (green) at each block. `seq2pair` peaks early; `pair2seq` peaks later, mirroring the staged patching pattern. **(c)** Sliding-window ablation (window size 15) during block-0 sequence patching. Ablating `seq2pair` early sharply reduces motif formation.

by computing the norm of the corresponding update at that block, normalized within each pathway across the trunk. The `seq2pair` contribution to z peaks in the first ten blocks and declines through the middle of the trunk, while `pair2seq` is small in early blocks and rises sharply in late blocks (Fig. 4b). The two pathways are not just *used* at different times: their update magnitudes themselves follow the staged pattern, with sequence-to-pair flow concentrated early and pair-to-sequence flow concentrated late.

Pathway ablations. Sliding-window ablations confirm the directionality of the early-window effect (Fig. 4c). When the ablation window starts in early blocks (0–7), ablating `seq2pair` during block-0 sequence patching reduces hairpin formation to near zero; the effect recovers fully once the window starts past block 12, indicating that `seq2pair` is specifically required during the early write-in phase. As a control, ablating `pair2seq` has little effect at early window positions, confirming that the loss of structure is specific to the sequence-to-pairwise pathway rather than a generic disruption from ablating either inter-representation pathway.

Generalization across architectures. We repeat all three analyses on OpenFold and Boltz-1 (App. I, Fig. 14). OpenFold and Boltz-1 show the same sequence of events: sequence patches shift z toward the donor within an early write-in window, freezing `seq2pair` prevents the shift, `seq2pair` contributions concentrate in early blocks, and early-block `seq2pair` ablations reduce motif formation.

4.2 Which Sequence Features are Propagated?

What biochemical information passes through the early sequence-to-pair pathway? We first focus on charge, which is relevant to hairpin stability: beta-hairpins can be stabilized by cross-strand salt bridges between oppositely charged residues (e.g., lysine (K) and glutamate (E)) on facing positions [Ciani et al., 2003]. We illustrate the construction and controls on ESMFold and report cross-architectural results in Fig. 5b.

Charge information is linearly decodable. To test whether charge-related information can be read out linearly, we use a difference-in-means approach to identify a "charge direction" in the sequence representation space. We collect sequence representations from a new set of 200 alpha-helical proteins sampled from the PDB (using helical proteins avoids confounding charge with the specific distance patterns present in beta hairpins). Let $\mathcal{P} = \{\mathrm{K}, \mathrm{R}, \mathrm{H}\}$ denote positively charged residues and $\mathcal{N} = \{\mathrm{D}, \mathrm{E}\}$ denote negatively charged residues. We compute the charge direction as:

$$v_{\text{charge}} = \frac{\bar{s}_{\mathcal{P}} - \bar{s}_{\mathcal{N}}}{\|\bar{s}_{\mathcal{P}} - \bar{s}_{\mathcal{N}}\|}, \quad \text{where} \quad \bar{s}_{\mathcal{P}} = \frac{1}{|\mathcal{P}|} \sum_{a \in \mathcal{P}} \mathbb{E}[s_i \mid x_i = a], \tag{4}$$

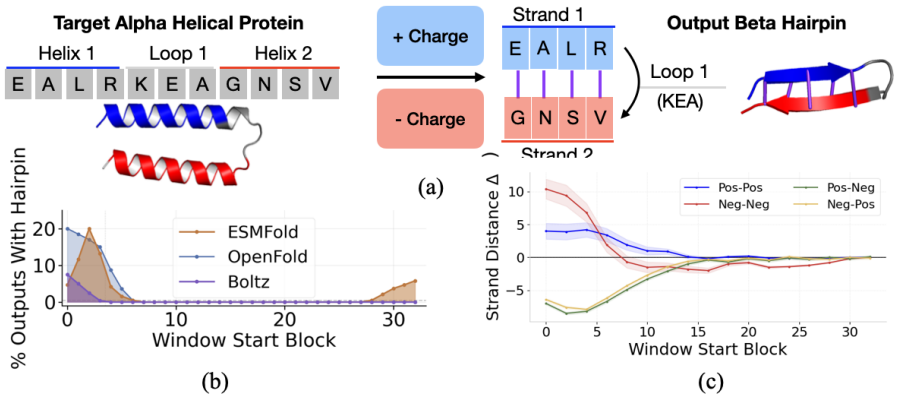


Figure 5: **Charge steering induces hairpin formation.** (a) Intervention setup: we steer the sequence representation toward opposite charges on the two helical regions flanking the loop, motivated by electrostatic complementarity. (b) Early-block steering (window size 15) induces hairpins in all three models. (c) ESMFold control: same-charge steering on β-hairpin strands and opposite-charge steering on α-helical targets produce sign-sensitive changes (strands separate; helices approach), concentrated in early blocks.

and analogously for $\bar{s}_{\mathcal{N}}$. Projections onto v_{charge} separate charge classes with high ROC-AUC across all three models throughout the trunk (App. J.1), showing that a linear readout can distinguish the charge labels. Linear probing on z shows that charge information enters the pairwise representation through each model's sequence-to-pairwise pathway in early trunk blocks (App. J.2). Is this information used to drive folding decisions?

Electrostatic complementarity steering. We add or subtract v_{charge} to steer the internal representation toward positive or negative charge while keeping the amino acid sequence fixed. For a target helix-turn-helix region with residue indices $\mathcal{S}_1$ (helix 1) and $\mathcal{S}_2$ (helix 2), we steer $s'_i = s_i + \alpha v_{\text{charge}}$ for $i \in \mathcal{S}_1$ and $s'_i = s_i - \alpha v_{\text{charge}}$ for $i \in \mathcal{S}_2$, leaving other positions unchanged. The strength α controls the magnitude, and the construction mimics the cross-strand electrostatic complementarity found in natural hairpins (Fig. 5a).

Steering induces hairpin formation across architectures. We apply this intervention to motif regions in our 100 target helical proteins, yielding approximately 500 cases per model, with steering strength $\alpha = 3\sigma$. Complementarity steering induces hairpin formation in all three models, with the effect concentrated in early blocks coinciding with stage 1 (Fig. 5b).

Sign and magnitude controls. Two controls test whether the effect is sign-sensitive and graded. *Same-charge steering* on 400 sampled beta-hairpins (steering both strands to the same charge) increases mean cross-strand distance by $\approx$10Å in early blocks (Fig. 5c, Pos-Pos/Neg-Neg), and opposite-charge steering on α-helical targets decreases it (Pos-Neg/Neg-Pos), as expected from the attraction/repulsion analogy that motivated the experiment. *Scaling charge strength* on small single-motif targets ($\leq$40 residues; App. J.3) shows that cross-strand distance changes smoothly and monotonically as steering strength increases in all three models.

Matched controls and a second biochemical feature. Norm- and variance-matched random directions have weaker effects than charge steering (App. O). To test whether the early pathway also carries information about hydrophobicity, we derive a hydrophobicity direction from uncharged sets $\{I, V, L, F, M, A\}$ and $\{N, Q, S, T, G\}$ using Eq. (4). Steering opposing faces of amphipathic helices increases exposure of the originally hydrophobic face by 43, 39, and 24 percentage points in ESMFold, OpenFold, and Boltz-1, respectively ($n = 500$ targets). Effects localize to early blocks, including Boltz-1's MSA module. The distinct geometric effects of charge and hydrophobicity steering show that the early pathway carries multiple biochemically meaningful features that causally influence structure.

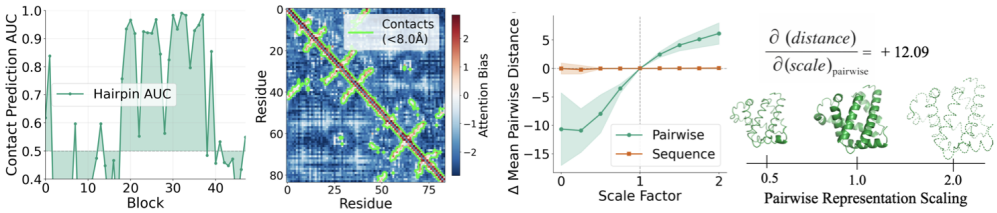


Figure 6: **Late blocks read z as geometric information (ESMFold). (a)** ROC-AUC for classifying contacting (Cα $<$ 8Å) versus non-contacting residue pairs from `pair2seq` bias values alone, computed per block. The bias cleanly separates contacts in middle and late blocks. **(b)** Head-averaged `pair2seq` bias at block 32 (red = positive, blue = negative); green contours mark residue pairs within 8Å. Positive bias concentrates at spatial contacts. **(c)** Effect of scaling the pairwise (z) versus sequence (s) representation by a factor in $[0, 2]$ before the structure module, shown as deviation in mean pairwise Cα distance from baseline (scale = 1.0). **(d)** Example structures at different pairwise scaling factors, with the gradient $\partial(\text{distance})/\partial(\text{scale})$ at the normal operating point.

5 Late Blocks: Pairwise Geometry Emerges

By the late blocks, the pairwise representation z has taken over as the primary carrier of folding-relevant information: pairwise patching induces structural change while sequence patching does not. We characterize how downstream computation reads z and verify across architectures (App. K).

5.1 Downstream Pathways Read z as Geometric Information

Two pathways consume z in the ESMFold trunk: the `pair2seq` bias modulates sequence self-attention (Eq. (1)), and z feeds directly into the structure module that produces 3D coordinates.

`pair2seq` broadcasts contacts to sequence attention. For each block, we extract the `pair2seq` bias values across all residue pairs and measure how well bias magnitudes separate contacts (Cα distance $<$ 8Å) from non-contacts, quantified by ROC-AUC. In middle and late blocks, the bias cleanly separates contacts from non-contacts (Fig. 6a), and visualizing bias values alongside contact maps shows positive bias concentrated at spatial contacts (Fig. 6b). Thus, late pairwise states make spatial contacts available to sequence attention.

The structure module reads z as a geometric signal. To isolate the structure module's use of z, we scale z by factors 0–2 before it enters the structure module while holding s fixed, and vice versa, across 600 proteins. Scaling z monotonically scales mean pairwise distance in the output: scaling up expands the predicted structure, scaling down contracts it (Fig. 6c,d).

Generalization across architectures. Both pathway behaviors replicate in OpenFold and Boltz-1 (App. K, Fig. 19). The `pair2seq` bias (and its Pairformer analog) separates contacts from non-contacts in middle and late blocks, and scaling pairwise representations before the structure module monotonically scales output distances.

5.2 Pairwise Representations Encode Distance

An intuitive hypothesis is that z linearly encodes pairwise distance. We test this by training linear probes to predict pairwise Cα distance from z at each block, $\hat{d}_{ij} = \mathbf{w}^\top z_{ij} + b$, where $z_{ij} \in \mathbb{R}^{d_z}$ is the pairwise representation for residues i and j. We use 400 proteins for training and 200 for evaluation (App. F), training a separate probe per block per model.

Distance is linearly encoded in late blocks across architectures. Probe accuracy is low at block 0 in all three models, before substantial trunk refinement, but rises through the trunk and plateaus at $R^2 \approx 0.9$ in late blocks (Fig. 7a). Distance is therefore linearly accessible from z in late blocks across all three architectures.

Steering z induces hairpin formation across architectures. Since the probe is linear, its gradient with respect to z_{ij} is the weight vector: $\nabla_{z_{ij}} \hat{d}_{ij} = \mathbf{w}$. We can therefore steer the representation

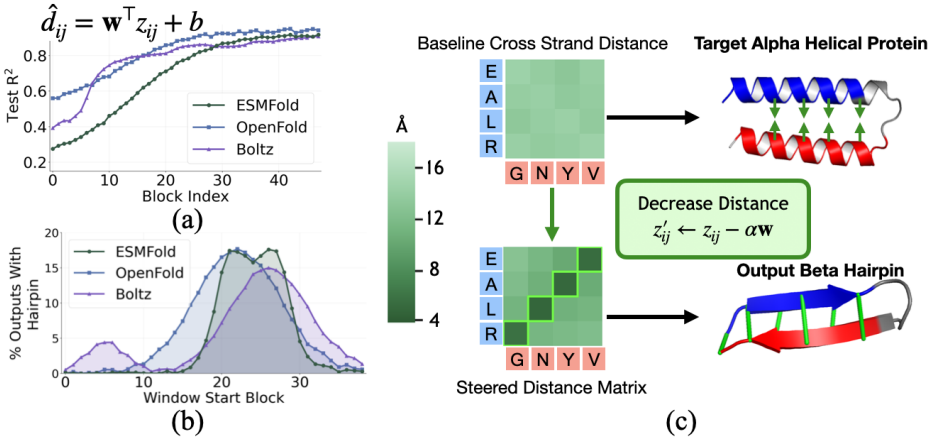


Figure 7: **Distance information is linearly accessible in late blocks and can be manipulated via steering.** (a) R^2 score of linear distance probes per block, reaching ≈ 0.9 at late blocks for all three models. (b) Hairpin induction rate (DSSP) after distance steering with window size 10, for ESMFold (green), OpenFold (blue), and Boltz-1 (purple). Induction peaks when steering middle-to-late blocks. (c) Intervention setup: we move cross-strand pair states along the direction that reduces predicted distance, using 5.5Å as a reference contact distance: $z'_{ij} \leftarrow z_{ij} - \alpha \mathbf{w}$. Successful steering produces a distance matrix with cross-strand residues in contact (dark green), decreasing cross-strand distances and inducing hairpin formation.

toward a target distance by moving along this direction. We choose 5.5Å as the reference cross-strand contact distance in the schematic. We steer by a fixed displacement along the probe direction. For cross-strand pairs $(i, j) \in \mathcal{C}$ that would form contacts in a hairpin, we steer $z'_{ij} = z_{ij} - \alpha \hat{\mathbf{w}}$, where $\hat{\mathbf{w}}$ is the normalized probe weight direction and $\alpha = 20\sigma$, with σ the standard deviation of residue-pair representations projected onto the distance direction. We apply the intervention symmetrically to both z_{ij} and z_{ji} across trunk blocks with window size 10. Steering produces measurable structural changes in all three models: cross-strand distances decrease and hairpins form in the target region (Fig. 7c). Hairpin induction peaks when steering middle-to-late blocks, where high probe accuracy coincides with the stage 2 effective window.

6 Folding Trunks Share a Representational Geometry

The shared stages and causal features suggest that different folding architectures may also develop a common representational geometry. We test this by aligning pairwise states across models, then asking whether those aligned states can drive the receiving model's structural predictions. We illustrate with ESMFold as the receiver; all-pairs results appear in App. L.

We first measure representational similarity using Centered Kernel Alignment [Kornblith et al., 2019], a training-free measure of how much two representations share linear structure. Fig. 8a shows CKA between every pair of blocks across ESMFold and OpenFold, and across ESMFold and Boltz-1. Similarity is high throughout middle and late blocks (CKA > 0.8 for ESMFold vs. OpenFold; CKA > 0.7 for ESMFold vs. Boltz-1) and concentrates along the diagonal: late blocks of one model are most similar to late blocks of the other.

Linear alignment captures this similarity. CKA suggests shared structure but does not show whether it is recoverable by a simple map. We test this with whitened Procrustes: for each model pair (M_A, M_B) and block k, we whiten paired representations $Z_A, Z_B \in \mathbb{R}^{N \times d}$ as $\tilde{Z}_X = Z_X \Sigma_X^{-1/2}$, then solve $\min_{R^\top R = I} \|\tilde{Z}_A R - \tilde{Z}_B\|_F^2$ for the optimal rotation R, giving the full projection $P = \Sigma_A^{-1/2} R \Sigma_B^{1/2}$. Whitening removes scale and covariance differences, leaving rotation to capture geometric correspondence. No folding signal or task supervision is used.

Fig. 8b reports per-block test R^2. Alignment is poor at the earliest measured block, rises through blocks 5–15, and plateaus thereafter. OpenFold aligns to ESMFold more tightly ($R^2 \approx 0.9$) than

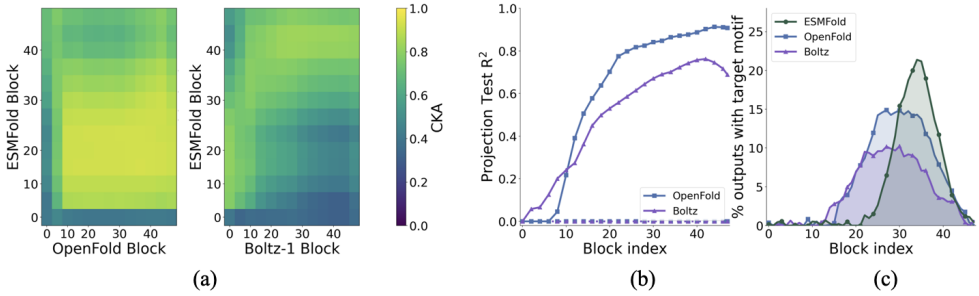


Figure 8: **Pairwise representations share geometry and support cross-model motif transfer.** **(a)** CKA between pairwise representations at every block pair across ESMFold and OpenFold, and ESMFold and Boltz-1. **(b)** Test R^2 of whitened Procrustes projections from OpenFold and Boltz-1 pairwise representations into ESMFold's space, evaluated per block. Solid lines use true correspondences; dotted lines are shuffled-correspondence baselines. **(c)** Single-block pairwise patching success when donor representations come from OpenFold or Boltz-1 (after Procrustes projection into ESMFold's space) versus from ESMFold itself. Cross-model patching reproduces the late-block pairwise window, with strength tracking projection quality from (b).

Boltz-1 does ($R^2 \approx 0.75$), consistent with their closer architectural relationship. Shuffling the residue-pair correspondences removes this strong alignment, indicating that it depends on matching the same pairs across models. We compare against other projection methods in App. L.1.

Aligned representations support partial functional transfer. Geometric alignment does not by itself show the representations carry the same *functional* information. We test this by patching projected donor representations into ESMFold's forward pass (single-block protocol of §3).

Cross-model patching reproduces the late-block pairwise window from within-model patching (Fig. 8c), with strength tracking alignment quality: ESMFold $\rightarrow$ ESMFold peaks near 22%, OpenFold $\rightarrow$ ESMFold near 15%, Boltz-1 $\rightarrow$ ESMFold near 10%. Cross-model patches induce hairpins only in ESMFold's late blocks, matching the pairwise stage from within-model analysis.

Generalization across all model pairs. App. L repeats these analyses across all three model pairings and both directions of each. The pattern is consistent: high CKA and Procrustes R^2 in middle and late blocks, poor held-out alignment under shuffled controls, and cross-model patching reproducing the late-block window in all six directions.

Matched calibration and failure analysis. We compare cross-model and within-model patches on the same cases, using the same receiving model, mask, and late-block windows. Cross-model peak success is 20–75% of the within-model peak. We bootstrap whole cases so that evaluations at different blocks of the same case stay together (App. Q). Final pairwise-state swaps into another model's decoder retain hairpins in 65–78% of $n = 500$ cases per direction, showing that final states can transfer more successfully when the receiving trunk does not process them further. On proteins with divergent predictions, decoded structures resemble the donor more than the host; an ESMFold2 decoder check gives the same qualitative result. These interventions establish functional compatibility across architectures while revealing where transfer depends on the receiving model. The remaining directional asymmetry is discussed in App. Q.

7 Related Work

Probing and causal intervention. Probes identify information that can be recovered from a representation [Belinkov, 2022]. Activation interventions test which states affect model outputs, through causal mediation analysis [Vig et al., 2020] and causal tracing of factual recall [Meng et al., 2022]. Steering connects latent directions to changes in generation [Subramani et al., 2022, Marks and Tegmark, 2024]. We combine these methods to connect biochemical and geometric features in the trunk to predicted structure.

Stages and shared representations. Layer-deletion and layer-swapping experiments motivate stages of inference in language models [Lad et al., 2024]. The Platonic Representation Hypothesis [Huh

et al., 2024] connects growing representational alignment across models and modalities to a shared statistical account of their data. We test staged computation and functional transfer of aligned states in folding architectures.

Protein model interpretability. InterPLM [Simon and Zou, 2025] and Adams et al. [2025] use sparse autoencoders to identify features in protein language models associated with structural motifs, functional annotations, and other biological properties. Gujral et al. [2025] use sparse autoencoders and transcoders to extract features associated with Gene Ontology terms and protein families. Villegas Garcia and Ansuini [2025] use such features to steer sequence generation, while Murakami et al. [2026] study features of an ESM-2 model fine-tuned for nanobody thermostability. Moving beyond sequence encoders, Gut and Lemmin [2025] examine AlphaFold2's structural capabilities through template perturbations and limited sequence information. Our focus is the folding trunk itself: we intervene on its sequence and pairwise states to trace how biochemical information becomes geometry across three architectures.

Beta-hairpin folding. Hairpins are a well-studied folding model. Statistical-mechanical models [Muñoz et al., 1998] and atomistic simulations [Dinner et al., 1999] investigate their kinetics and stabilizing interactions. We use motif transfer to study structural decisions inside prediction models.

8 Limitations

We test local helices and hairpins, four-strand sheets, sequence-distant contacts, and proteins up to 1,000 residues (App. M). Other topologies, larger proteins, multimers, and pairwise-free architectures such as SimpleFold [Wang et al., 2025] remain open. Interventions can produce clashes, chain breaks, and reduced confidence. Design/refold and cross-model checks support many coherent backbones (App. N), but do not establish experimental stability or that the target sequence adopts the patched fold.

9 Discussion

We have conducted a cross-architectural mechanistic analysis of protein folding trunks, revealing a shared two-stage computational structure across the tested structural decisions in ESMFold, OpenFold, and Boltz-1. In the first stage, early computation propagates biochemical information from sequence into pairwise representations; steering a charge-related direction induces hairpin formation. In the second stage, later computation develops geometric features in the pairwise representation; distances are linearly decodable with high accuracy, and steering along the distance direction induces hairpin formation. These features are not merely correlational: interventions show that they influence structural predictions across architectures. Cross-model patching demonstrates that pairwise representations are alignable via linear projection, and representations from one model can partially substitute for another's at the corresponding stage. The two-stage structure holds for both hairpin and helix induction, suggesting it reflects a recurring computational strategy across the folding trunks studied here.

Protein folding models contain learned structural knowledge that may be valuable beyond their ability to output coordinates. Interpretability has supported scientific discovery in other domains [Davies et al., 2021, 2024, Wong et al., 2024, Schut et al., 2025, Wang et al., 2026], raising the possibility of extracting biological insights from folding models as well. Our results provide a starting point by connecting biochemical features to the internal computations that shape a predicted structure. Extending this approach could help reveal which residue interactions a model relies on, explain its structural preferences and errors, and turn these explanations into experimentally testable hypotheses about protein behavior.

Conditional protein behavior offers one concrete opportunity. Proteins can adopt different conformations as their biochemical environment changes; for example, specific histidine residues influence the pH sensitivity of influenza hemagglutinin's conformational changes [Mair et al., 2014]. Our charge-steering results suggest a way to investigate whether folding models have learned information relevant to such transitions: intervene on biochemical features and examine how the predicted structure responds. Comparing these responses with experimentally measured conformations and mutations could establish whether latent interventions capture meaningful aspects of conditional behavior. This would be a step toward using the knowledge inside folding models to study how proteins respond to changing conditions.

References

Etowah Adams, Liam Bai, Minji Lee, Yiyang Yu, and Mohammed AlQuraishi. From mechanistic interpretability to mechanistic biology: Training, evaluating, and interpreting sparse autoencoders on protein language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 460–476. PMLR, 2025. URL https://proceedings.mlr.press/v267/adams25a.html.

Gustaf Ahdritz, Nazim Bouatta, Christina Floristean, Sachin Kadyan, Qinghui Xia, William Gerecke, Timothy J O'Donnell, Daniel Berenberg, Ian Fisk, Niccolò Zanichelli, Bo Zhang, Arkadiusz Nowaczynski, Bei Wang, Marta M Stepniewska-Dziubinska, Shang Zhang, Adegoke Ojewole, Murat Efe Guney, Stella Biderman, Andrew M Watkins, Stephen Ra, Pablo Ribalta Lorenzo, Lucas Nivon, Brian Weitzner, Yih-En Andrew Ban, Shiyang Chen, Minjia Zhang, Conglong Li, Shuaiwen Leon Song, Yuxiong He, Peter K Sorger, Emad Mostaque, Zhao Zhang, Richard Bonneau, and Mohammed AlQuraishi. OpenFold: retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization. *Nat. Methods*, 21(8):1514–1524, August 2024. doi: 10.1038/s41592-024-02272-z.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1): 235–242, 2000. doi: 10.1093/nar/28.1.235.

F Blanco, M Ramírez-Alvarado, and L Serrano. Formation and stability of beta-hairpin structures in polypeptides. *Current Opinion in Structural Biology*, 8(1):107–111, 1998. doi: 10.1016/S0959-440X(98)80017-1.

Carl-Ivar Brändén and John Tooze. *Introduction to Protein Structure*. Garland Science, 2 edition, 1999. doi: 10.1201/9781136969898.

Salvatore Candido, Thomas Hayes, Alexander Derry, Roshan Rao, Zeming Lin, Robert Verkuil, Bryan Wu, Jin Sub Lee, Elise S. Bruguera, Jehan A. Keval, Mykhailo Kopylov, John E. Pak, Wesley Wu, Neil Thomas, Samson Mataraso, Alvin Hsu, Ashton C. Trotman-Grant, Kilian Fatras, Allan dos Santos Costa, Rohil Badkundri, Halil Akın, Deniz Oktay, Jonathan Deaton, Elizabeth Montabana, Hrishita Sitwala, Yue Yu, Marius Wiggert, Dylan Alexander Carlin, Anthony W. Goering, Tomasz Blazejewski, McCullen Sandora, Michael Hla, Tina Z. Jia, Leon H. Kloker, Nicholas J. Sofroniew, Masatoshi Uehara, Jassi Pannu, Sharrol Bachas, Daniel S. Liu, Tom Sercu, and Alexander Rives. Language modeling materializes a world model of protein biology, 2026. URL https://www.biorxiv.org/content/10.64898/2026.06.03.729735. Preprint.

Barbara Ciani, Muriel Jourdan, and Mark S. Searle. Stabilization of beta-hairpin peptides by salt bridges: Role of preorganization in the energetic contribution of weak interactions. *Journal of the American Chemical Society*, 125(30):9038–9047, 2003. doi: 10.1021/ja030074l. PMID: 15369359.

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.

Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, et al. Advancing mathematics by guiding human intuition with ai. *Nature*, 600(7887):70–74, 2021. doi: 10.1038/s41586-021-04086-x.

Alex Davies, András Juhász, Marc Lackenby, and Nenad Tomašev. The signature and cusp geometry of hyperbolic knots. *Geometry & Topology*, 28(5):2313–2343, 2024. doi: 10.2140/gt.2024.28.2313.

Aaron R. Dinner, Themis Lazaridis, and Martin Karplus. Understanding beta-hairpin formation. *Proceedings of the National Academy of Sciences of the United States of America*, 96(16):9068–9073, 1999. doi: 10.1073/pnas.96.16.9068.

Onkar Gujral, Mihir Bafna, Eric Alm, and Bonnie Berger. Sparse autoencoders uncover biologically interpretable features in protein language model representations. *Proceedings of the National Academy of Sciences*, 122(34):e2506316122, 2025. doi: 10.1073/pnas.2506316122. URL https://www.pnas.org/doi/10.1073/pnas.2506316122.

Jannik Adrian Gut and Thomas Lemmin. Dissecting AlphaFold2's capabilities with limited sequence information. *Bioinformatics Advances*, 5(1):vbae187, 01 2025. ISSN 2635-0041. doi: 10.1093/bioadv/vbae187. URL https://doi.org/10.1093/bioadv/vbae187.

Maarten L. Hekkelman, Daniel Álvarez Salmoral, Anastassis Perrakis, and Robbie P. Joosten. DSSP 4: FAIR annotation of protein secondary structure. *Protein Science*, 34(8):e70208, 2025. doi: 10.1002/pro.70208.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis, 2024. URL https://arxiv.org/abs/2405.07987.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Dmitry Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596: 583–589, 2021. doi: 10.1038/s41586-021-03819-2.

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

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529, 2019.

Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. The remarkable robustness of llms: Stages of inference?, 2024. URL https://arxiv.org/abs/2406.19384.

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.

Caroline M. Mair, Tim Meyer, Katjana Schneider, Qiang Huang, Michael Veit, and Andreas Herrmann. A histidine residue of the influenza virus hemagglutinin controls the pH dependence of the conformational change mediating membrane fusion. *Journal of Virology*, 88(22):13189–13200, 2014. doi: 10.1128/JVI.01704-14.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pages 17359–17372, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Victor Muñoz, Eric R Henry, James Hofrichter, and William A Eaton. A statistical mechanical model for β-hairpin kinetics. *Proceedings of the National Academy of Sciences*, 95(11):5872–5879, 1998. doi: 10.1073/pnas.95.11.5872.

Taihei Murakami, Yuki Hashidate, and Yasuhiro Matsunaga. Mechanistic interpretability of fine-tuned protein language models for nanobody thermostability prediction. *Bioinformatics*, 42(10): btag685, 2026. doi: 10.1093/bioinformatics/btag685.

Lisa Schut, Nenad Tomašev, Thomas McGrath, Demis Hassabis, Ulrich Paquet, and Been Kim. Bridging the human–AI knowledge gap through concept discovery and transfer in AlphaZero. *Proceedings of the National Academy of Sciences*, 122(13):e2406675122, 2025. doi: 10.1073/pnas.2406675122.

Elana Simon and James Zou. InterPLM: discovering interpretable features in protein language models via sparse autoencoders. *Nature Methods*, 22:2107–2117, 2025. doi: 10.1038/s41592-025-02836-7.

Nishant Subramani, Nivedita Suresh, and Matthew Peters. Extracting latent steering vectors from pretrained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.48. URL https://aclanthology.org/2022.findings-acl.48/.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

Edith Natalia Villegas Garcia and Alessio Ansuini. Interpreting and steering protein language models through sparse autoencoders. In *ICLR 2025 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2025. URL https://openreview.net/forum?id=zjGwoJaVPS.

Nicholas Wang, Ching Fang, Mark Bissell, Dron Hazra, Michael Pearce, Archa Jain, Daniel Balsam, Christoforos Nalmpantis, Pouya Niki, Pooja Kathail, Andrey Karailiev, Javkhlan-Ochir Ganbat, Luca Giacomoni, Jonathan Wan, and Ravi Solanki. Using interpretability to identify a novel class of Alzheimer's biomarkers. Goodfire and Prima Mente research blog, January 2026. URL https://www.goodfire.com/research/interpretability-for-alzheimers-detection. January 28, 2026.

Yuyang Wang, Jiarui Lu, Navdeep Jaitly, Josh Susskind, and Miguel Angel Bautista. SimpleFold: Folding proteins is simpler than you think, 2025. URL https://arxiv.org/abs/2509.18480.

Jeremy Wohlwend, Gabriele Corso, Saro Passaro, Mateo Reveiz, Ken Leidal, Wojtek Swiderski, Tally Portnoi, Itamar Chinn, Jacob Silterra, Tommi Jaakkola, and Regina Barzilay. Boltz-1 democratizing biomolecular interaction modeling. *bioRxiv*, 2024. doi: 10.1101/2024.11.19.624167. URL https://www.biorxiv.org/content/early/2024/11/20/2024.11.19.624167.

Felix Wong, Erica J Zheng, Jacqueline A Valeri, Nina M Donghia, Melis N Anahtar, Satotaka Omori, Alicia Li, Andres Cubillos-Ruiz, Aarti Krishnan, Wengong Jin, et al. Discovery of a structural class of antibiotics with explainable deep learning. *Nature*, 626(7997):177–185, 2024. doi: 10.1038/s41586-023-06887-8.

A Protein Structure

Chains of amino acids. Proteins are linear chains of amino acids; once incorporated into a protein, each amino acid is called a *residue*. Each residue consists of a backbone (common to all residues) and a side chain (which differs between residue types and drives folding). The backbone contains the amino nitrogen (N), central carbon (Cα), and carbonyl group (C=O); Cα–Cα distances are commonly used to measure inter-residue distances, while backbone N–H and C=O groups form the hydrogen bonds that stabilize secondary structures. The linear order of residues—the *protein sequence*—determines how they interact when the chain folds into a three-dimensional structure.

...AQTVDSWTWENGKWTWKEQLNS...

Strand 1 Loop Strand 2

Figure 9: **Sequence view of a β-hairpin (highlighted) within a protein**: two β-strand segments separated by a loop. Colored braces label the strand–loop–strand decomposition.

From sequence to structure. The amino acid sequence (Fig. 9) defines the linear order of residues in the chain. During folding, residues far apart in sequence can become neighbors in 3D space. Protein structure is described at multiple levels: *secondary structure* refers to local motifs such as alpha helices and beta strands (Fig. 10). We focus on these two motifs as canonical structural decisions a folding model must make.

Beta-hairpins. A beta-hairpin (Fig. 10) is a common secondary-structure motif formed by two beta strands that run in opposite directions along the protein backbone (antiparallel orientation). The two strands are connected by a short loop that allows the chain to reverse direction. This loop brings the strands into close proximity, enabling them to align side by side. Once aligned, the strands are stabilized primarily by backbone hydrogen bonds; side-chain interactions provide additional, sequence-dependent stabilization.

Beta-hairpins constitute a non-trivial minimal example of a protein folding motif, and have a long history as a model system in experimental, theoretical, and molecular dynamics studies of folding [Muñoz et al., 1998, Blanco et al., 1998, Dinner et al., 1999]. Simpler features, such as isolated secondary structure elements, do not involve longer-range residue–residue interactions, while larger motifs and full protein folds introduce many interacting components that obscure causal analysis.

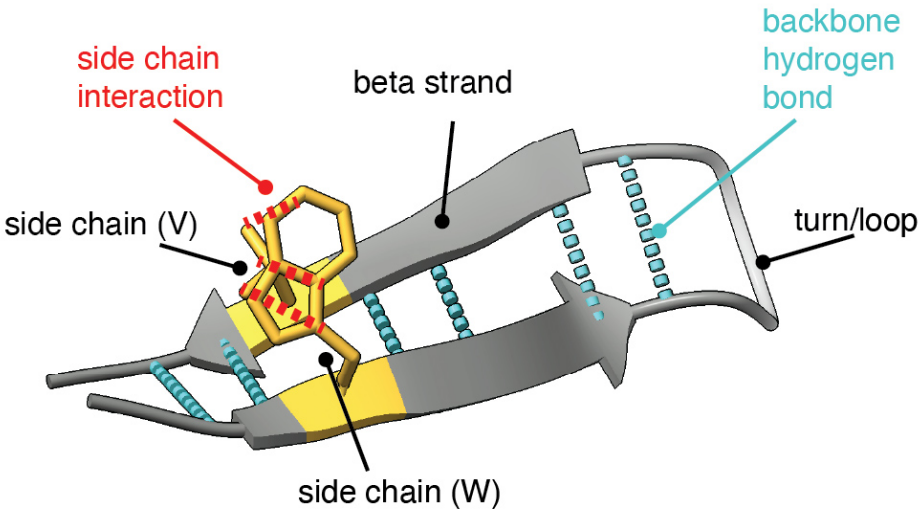


Figure 10: **3D cartoon diagram of a beta-hairpin**, a common secondary-structure motif consisting of two antiparallel β-strands (side chain V and side chain W) connected by a short turn/loop. Interactions between side chains on opposing strands and backbone hydrogen bonding together determine the folded hairpin geometry.

B Impact Statement

This work studies interpretability in protein structure prediction models. Improved mechanistic understanding can support safer and more reliable use of these models in basic research and downstream applications, including protein design, by enabling targeted diagnosis of failure modes and more controllable interventions. The methods are analytical and do not introduce new predictive capabilities or training data. Potential negative impacts include dual-use concerns if interpretability insights accelerate protein engineering in harmful contexts; such risks are best mitigated through standard biosecurity governance and careful application review. Overall, we view the primary impact as advancing scientific understanding of how folding models operate.

C Code and Computational Cost

Code. Code, ESMFold demonstration notebooks, and reproduction instructions are available at `https://github.com/kevinlu4588/ProteinFolding`. Data and saved results are available at `https://huggingface.co/datasets/kevinlu4588/ProteinFolding`. The repository contains a single `reproduce.sh` entry point that runs the key experiments in sequence for ESMFold, along with three Jupyter notebooks that demonstrate the patching, charge-steering, and distance-steering interventions on a single example without requiring the full dataset.

Hardware. The original experiments were run on a single workstation with 2× NVIDIA RTX A6000 GPUs (48 GB VRAM each). Each model inference fits on a single A6000; the second GPU was used to parallelize across cases rather than to shard a single forward pass.

Wall-clock budget. The dominant cost is repeated forward passes under intervention. We study three models, ESMFold, OpenFold, and Boltz-1, and run the same intervention sweeps on each. Approximate per-model wall-clock times on the setup above are:

Experiment	Wall-clock (per model)
Single-block patching (all trunk blocks)	∼1.5 days
Sliding-window ablations	∼1 day
Charge / distance steering sweeps	∼1 day
Cross-model patching	∼1.5 days
Module patching, representation tracking, misc.	∼0.5 day

Table 1: Approximate wall-clock cost on 2× A6000 per model. Costs scale roughly linearly with the number of cases, which is configurable in `reproduce.sh`.

Aggregated across the three models and including pilot runs and re-runs after iteration, total compute spent on the experiments reported in this paper is approximately **2–3 weeks of wall-clock time on** 2× **A6000** (roughly 1,000–1,500 GPU-hours). These estimates describe the original workstation experiments. The longer-protein extension used two H100 GPUs; a complete compute budget for the added validation experiments is not reported here.

D Secondary Structure Detection Algorithm

D.1 Hairpin Detection Algorithm

A beta hairpin is defined as two antiparallel β-strands connected by a short loop. We detect hairpins from predicted structures using the following criteria:

1. **Secondary structure assignment**: We run DSSP [Hekkelman et al., 2025] on the predicted structure to identify β-strand regions (DSSP codes E or B).
2. **Strand length**: Each strand must contain at least 2 residues.
3. **Loop length**: The connecting loop must be 0–5 residues.
4. **Adjacency**: The two strands must be sequential in the primary sequence (no intervening strands).

D.2 Helix-Loop-Helix Detection Algorithm

A helix-loop-helix motif is defined as two α-helices connected by a short loop. We detect helix-loop-helix motifs from predicted structures using the following criteria:

1. **Secondary structure assignment**: We run DSSP [Hekkelman et al., 2025] on the predicted structure to identify α-helical regions (DSSP code H).
2. **Helix length**: Each helix must contain at least 4 residues.
3. **Loop length**: The connecting loop must be 2–8 residues.
4. **Adjacency**: The two helices must be sequential in the primary sequence (no intervening helices).

E Licensing of existing assets

We use several existing models and public protein-structure resources. ESMFold/ESM is released under the MIT license, OpenFold is released under the Apache 2.0 license, and Boltz-1 is released under the MIT license. For data resources, the Protein Data Bank (PDB) archive is distributed under CC0 1.0, CATH and UniProt are distributed under CC BY 4.0, and DSSP is used for secondary-structure annotation under its published usage terms. We use these resources for research purposes and will include full license and attribution details in the accompanying code and data release.

F Experiment Datasets

Alpha-helical protein dataset (CATH Class 1). To obtain a diverse set of all-α proteins for representation analysis, we constructed a dataset from CATH Class 1 ("mainly α") domains. For each unique Class 1 PDB chain in the CATH domain list, we retrieved the full structure and extracted the complete chain sequence.

To ensure that these proteins were purely helical, we verified secondary structure assignments using DSSP and excluded any chain containing β-strand residues (DSSP codes E or B). We further filtered sequences to length 90–400 residues and removed duplicate sequences, yielding 5,000 protein chains. From this remaining set, we selected 200 proteins spanning diverse CATH architectures and topologies, including helix bundles, helix–turn–helix motifs, coiled coils, EF-hand proteins, and helical repeat proteins.

This dataset is also used for training the difference-of-means (DoM) vectors that identify biochemical directions (e.g., charge) in sequence representations.

Beta Hairpin Dataset Protein structures were selected from the PISCES database using a non-redundant culling at 25% sequence identity, restricting to X-ray structures with resolution between 0.0 and 2.5 Å and R-factor $\leq$ 0.3. Only protein chains with lengths between 40 and 10,000 residues were retained. We used the PISCES culling set `cullpdb_pc25.0_res0.0-2.5_len40-10000_R0.3_Xray_ d2025_02_19_chains11652`, which yields 11,652 high-quality protein chains. Subsequent identification of beta-hairpins was performed on this filtered set as described in App. D.

Activation patching target dataset. For activation patching experiments, we selected 100 alpha helical proteins and 100 hairpin proteins from the datasets described above. Sequence lengths were restricted to 100–400 residues. These proteins serve as targets into which donor motif representations are patched.

Probing dataset. To evaluate whether biochemical and geometric features are linearly decodable from trunk representations, we built a train/test dataset containing both α-helical and β-hairpin proteins.

The *training set* consists of:

- the 200 α-helical proteins described above (CATH Class 1), and
- 200 full-chain β-hairpin proteins sampled from a curated motif dataset.

The *test set* consists of:

- all α-helical target proteins used in patching experiments, and
- 100 additional full-chain β-hairpin proteins disjoint from the training hairpins.

All sequences were required to contain only the 20 standard amino acids. Hairpin sequences were further filtered to ensure $\leq 25\%$ pairwise sequence identity to any α-helical sequence or previously selected hairpin, and to exclude exact matches to donor sequences used in activation patching. The hairpin filters constrain sequence overlap within and between these cohorts.

G Effect of Recycling on ESMFold Predictions

We evaluated the effect of ESMFold recycling by comparing predictions generated with different numbers of recycles for a randomly sampled subset of UniProt protein sequences. For each protein, structures obtained with fewer recycles were compared to the highest-recycle prediction using Cα RMSD. Short proteins (< 100 aa) show a rapid decrease in RMSD between 0 and 1 recycle, followed by progressively smaller changes with additional recycling, indicating early structural convergence. In contrast, medium and long proteins continue to exhibit substantial refinement across multiple recycles, suggesting that recycling is most critical for larger systems and has limited impact for small proteins.

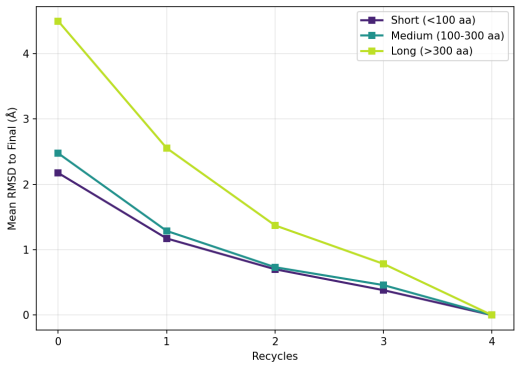


Figure 11: Mean Cα RMSD to the highest-recycle ESMFold prediction as a function of the number of recycles, shown for short (< 100 aa), medium (100–300 aa), and long (> 300 aa) proteins sampled from UniProt. For short sequences, the RMSD drops sharply between 0 and 1 recycle and decreases only modestly thereafter, indicating that most structural convergence occurs within the first recycle. In contrast, longer proteins continue to show substantial refinement with additional recycling.

H Additional Patching Experiments

H.1 Performance in Other Modules

To localize where hairpin-relevant computation occurs within ESMFold, we compared activation patching across three architectural components: the ESM-2 encoder, the folding trunk, and the structure module. For each module, we patched donor hairpin representations into the corresponding target representations and measured whether the output structure contained a hairpin.

ESM encoder patching. We extracted hidden states from all 33 layers of the ESM-2 encoder for the donor hairpin region and patched them into the corresponding positions during the target's forward pass. Despite replacing the encoder's learned sequence representations, encoder patching rarely induced hairpin formation (Fig. 12), suggesting that the encoder representations alone do not determine secondary structure.

Structure module patching. We patched the outputs of the invariant point attention (IPA) module at each of its eight iterations. Structure module patching was similarly ineffective: the IPA operates

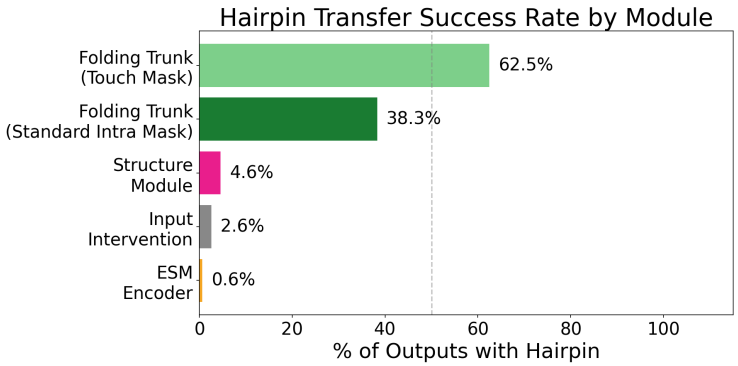


Figure 12: We apply activation patching to the ESM protein language model encoder, folding trunk, and structure module. Patching in the folding trunk is the only place where hairpins consistently transfer. Input intervention: alter the sequence residue text to match hairpin residues before inputting into ESMFold model

on representations that have already been shaped by the trunk, and patching at this late stage cannot override the geometric information established earlier.

Trunk patching. In contrast, patching both sequence and pairwise representations across all 48 blocks of the folding trunk successfully induced hairpin formation in approximately 40% of cases using standard patching. This makes the trunk a useful site for studying how these motif predictions can be changed, motivating the main-text analysis.

Pairwise patching masks. When patching the pairwise representation z, we must specify which entries z_{ij} to replace. We experimented with two masking strategies (illustrated in Fig. 13):

- **Standard intra patching**: We patch only entries where both residues i and j fall within the hairpin region, i.e., z_{ij} for $i, j \in [\text{start}, \text{end})$. This captures interactions strictly within the hairpin motif.
- **Touch patching**: We patch a cross-shaped region that includes (1) all intra-region entries, plus (2) entries where exactly one residue is in the hairpin region and the other is in the flanking sequence. Formally, we patch z_{ij} whenever $i \in [\text{start}, \text{end})$ or $j \in [\text{start}, \text{end})$. This captures how the hairpin region interacts with its sequential neighbors, which may be important for properly situating the motif within the larger structure.

Input intervention baseline. As an additional baseline, we performed a literal sequence substitution: replacing the target's amino acid sequence in the patch region with the donor's hairpin sequence before running ESMFold. This provides a direct comparison between replacing the input sequence and replacing internal states. As shown in Fig. 12, input intervention induces hairpins at a lower rate than trunk patching, indicating that the learned representations in the trunk carry information beyond what is present in the raw sequence.

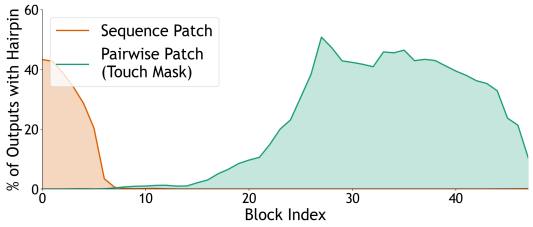


Figure 13: **Single-block patching with touch mask.** Pairwise patching using the touch mask (green) shows a broader effective window extending into late blocks (25–47), compared to sequence patching (orange) which is effective only in early blocks (0–8). The touch mask patches interactions between hairpin residues and flanking regions, bypassing the need for information to propagate from the patched region to neighboring residues.

H.2 Touch Patching Extends the Effective Window for Pairwise Interventions

In the main text (Fig. 1), we report pairwise patching results using standard (intra-region) masking, which patches only entries z_{ij} where both i and j fall within the hairpin region. Here we compare this to touch patching, which additionally patches entries where the hairpin region interacts with flanking residues.

Touch patching yields a broader effective window that extends further into late blocks (Fig. 13). We attribute this to information propagation constraints. With standard patching, donor information is injected only into the intra-hairpin entries of z; this information must then propagate outward through subsequent blocks to update how the hairpin interacts with flanking residues. If we patch too late, insufficient blocks remain for this propagation to occur, and the resulting structure is incoherent—the hairpin region conflicts with its surroundings.

Touch patching bypasses this bottleneck by directly providing the hairpin-to-flanking interactions, eliminating the need for propagation. This allows effective patching even in very late blocks where standard patching fails. The result highlights that successful structural transfer requires not only the correct intra-motif geometry but also sufficient time (i.e., remaining blocks) for the model to integrate the patched region with the rest of the chain.

I Generalization of Information Flow Across Architectures

In §4.1, we showed that early ESMFold blocks transfer biochemical information from sequence into pairwise representations through the `seq2pair` pathway, and that this transfer underlies the early-block sequence patching window. Here we replicate the four analyses underlying that conclusion, single-block patching, representation similarity under interventions, pathway contributions, and sliding-window ablations, across all three folding architectures. Results are summarized in Fig. 14.

Two-stage patching pattern (Fig. 14a). The single-block patching analysis from §3 produces qualitatively the same pattern in all three models: sequence interventions act earlier than sustained pairwise interventions. ESMFold and OpenFold show stronger sequence-patching success; Boltz-1 sequence success is 15% before filtering and 7% after designability filtering. Boltz-1 has a broader filtered pairwise window, blocks 3–47 (App. N). One possible contributor is the absence of `seq2pair` write-back in Boltz-1's downstream Pairformer: once z enters the Pairformer, it is refined within the pairwise track, without further sequence-to-pair updates. This may make donor pairwise states harder to overwrite. In ESMFold and OpenFold, by contrast, early-block pairwise patches are rewritten by subsequent `seq2pair` updates, narrowing the effective window to late blocks.

Representation similarity under interventions (Fig. 14b). We replicate the interpolation-coefficient analysis (Eq. 3) on OpenFold and Boltz-1. In all three models, sequence patching at block 0 shifts z toward the donor within the first $\approx$10 blocks, while freezing `seq2pair` during this window prevents the shift entirely; direct pairwise patches at block 0 do not persist. The early write-in window (highlighted in yellow) is therefore present in all three architectures. In Boltz-1, the relevant `seq2pair` pathway lives in the upstream MSA module (blocks 0–3) rather than throughout the trunk; the freeze condition correspondingly targets these blocks.

Pathway contributions (Fig. 14c). Without intervention, the relative magnitudes of `seq2pair` and `pair2seq` updates follow the staged pattern in all three models: `seq2pair` peaks during early blocks and declines, while `pair2seq` is small early and rises in middle and late blocks. The location of the `seq2pair` peak shifts to track each model's sequence-to-pairwise pathway: early throughout the trunk in ESMFold and OpenFold, confined to blocks 0–3 in Boltz-1 (corresponding to its upstream MSA module).

Sliding-window ablations (Fig. 14d). In all three models, ablating `seq2pair` during a window starting in early blocks reduces motif formation, with recovery once the window starts past the early write-in phase (block 12 in ESMFold and OpenFold, block 4 in Boltz-1, again tracking the location of each model's `seq2pair` pathway). Ablating `pair2seq` has little effect at any window position in any model, ruling out generic disruption from removing either inter-representation pathway. Together with the patching, similarity, and contribution analyses, these results support the interpretation that

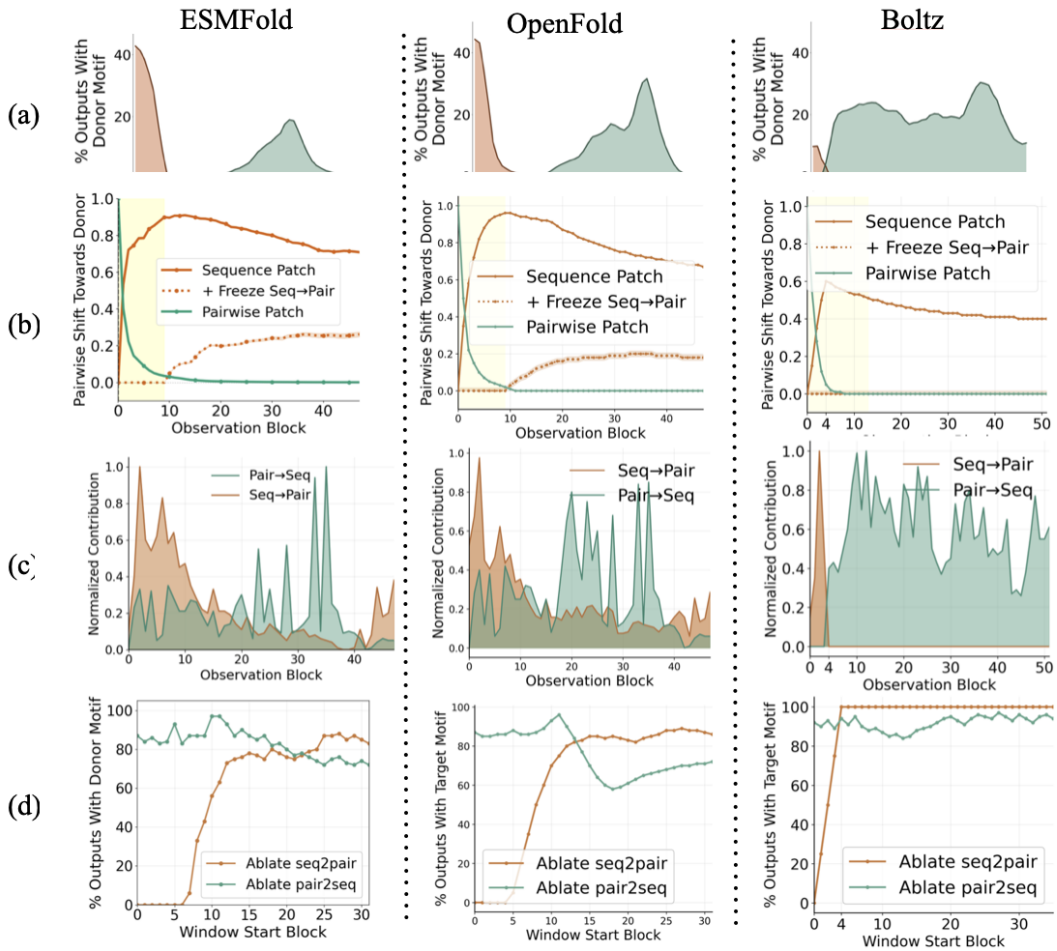


Figure 14: **Information flow analysis across ESMFold, OpenFold, and Boltz-1. (a)** Single-block patching success rate as a function of block index. Sequence patches (orange) are effective in early blocks across all three models; pairwise patches (green) are effective in late blocks. Boltz-1 additionally shows a broad pairwise window across the whole trunk. **(b)** Shift in z during the target forward pass under sequence patching at block 0, sequence patching with `seq2pair` frozen during the early window, and pairwise patching at block 0. All three models show a clear early write-in window (yellow) during which sequence patches shift z toward the donor and freezing `seq2pair` prevents this shift. **(c)** Normalized contribution magnitudes of `seq2pair` (orange) and `pair2seq` (green) at each block. ESMFold and OpenFold show earlier `seq2pair` and later `pair2seq` peaks. In Boltz-1, the `seq2pair` contribution is concentrated in the upstream MSA module (blocks 0–3) and essentially zero across the rest of the Pairformer trunk. **(d)** Sliding-window ablation (window size 15) during block-0 sequence patching. In all three models, ablating `seq2pair` during early blocks reduces motif formation, with full recovery once the window starts past the early write-in phase. Ablating `pair2seq` has little effect at any window position.

stage 1, the biochemical write-in from sequence into pairwise space, is a shared property of all three folding trunks, with its location varying based on where each architecture implements its `seq2pair` pathway.

J Additional Charge Experiments

J.1 Charge Direction Separability

To assess the linear decodability of charge information from each model's sequence representation, we computed a charge direction using the difference-of-means approach (Eq. 4) and measured how well projections onto this direction separate positively charged residues (K, R, H) from negatively charged residues (D, E). For each block in each model, we report ROC-AUC of the projection as a binary classifier.

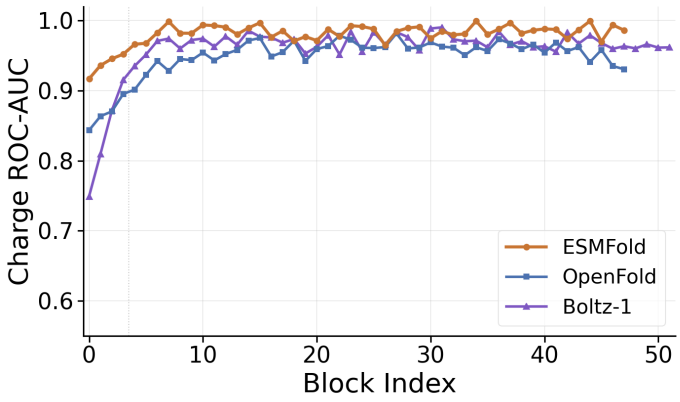


Figure 15: **Charge information is linearly decodable from sequence representations across all three models.** ROC-AUC of the difference-of-means charge direction v_{charge} as a binary classifier separating positively charged residues (K, R, H) from negatively charged residues (D, E), computed per block. All three models reach AUC near 1.0 within the first few blocks and maintain it throughout the trunk.

Fig. 15 shows that projections onto v_{charge} distinguish the charge groups throughout most of the trunk in all three models. ESMFold and OpenFold start at AUC ≈ 0.85–0.9 at block 0 and rise to near 1.0 within the first 5–10 blocks. Boltz-1's curve begins lower at the earliest MSA block (AUC ≈ 0.75) and rises sharply through the MSA module to AUC ≈ 0.95 by block 0 of the Pairformer, after which it remains high.

These results show that charge-related information remains linearly decodable throughout most of the trunk in each model. This property enables our charge steering intervention (§4.2): the difference-of-means construction yields a meaningful charge direction in every architecture, supporting the cross-architectural steering results in Fig. 5b.

J.2 Charge Probing

To verify that charge information propagates from sequence representations into the pairwise representation via `seq2pair`, we trained linear probes to predict residue charge directly from pairwise features. We extend this analysis across all three models to test whether the architectural variation identified in App. I (specifically, that Boltz-1's seq2pair pathway lives in the upstream MSA module rather than the trunk) shows up at the level of a specific biochemical feature.

Probe setup. For each block k in each model, we train two binary linear classifiers on the pairwise representation $z_{ij}^{(k)}$: one to detect positively charged residues (K, R, H) and one to detect negatively charged residues (D, E). Given a pair of residues (i, j) with sufficient sequence separation ($|i-j| \geq 4$), the probe predicts the charge class of residue i from the pairwise feature vector z_{ij}. We train analogous probes on each model's `seq2pair` output (or its architectural analog) to track when charge information first enters the pairwise pathway. We train probes on the probing dataset described in App. F, reporting balanced accuracy to account for class imbalance (charged residues comprise approximately 15% of natural proteins).

Results. Figure 16 shows probe accuracy across blocks for all three models.

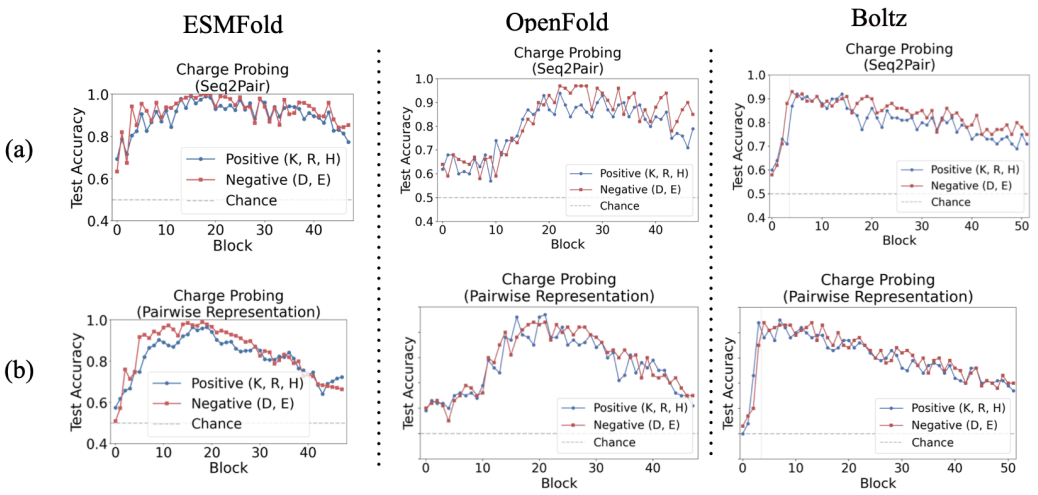


Figure 16: **Charge information is linearly accessible in pairwise representations across all three architectures.** Balanced accuracy of linear probes predicting positive charge (K, R, H; blue) and negative charge (D, E; red) from **(a)** the `seq2pair` output and **(b)** the pairwise representation z. Probes predict the charge of residue i from pairwise features at position (i, j) for sequence-distant pairs ($|i - j| \geq 4$). `seq2pair` output maintains high accuracy from early blocks throughout the trunk, while z probes start near chance at block 0 and rise through blocks 0–15, consistent with charge information transferring into z via `seq2pair` during early trunk blocks. The location of the rise tracks each model's sequence-to-pairwise pathway.

In all three models, probes on z start near chance at block 0 (before substantial pairwise refinement) and rise through the early blocks, plateauing at accuracies comparable to the corresponding `seq2pair` probes. The location of the rise tracks each model's `seq2pair` pathway: blocks 0–15 in ESMFold and OpenFold, and the narrower window of blocks 0–3 in Boltz-1, where the `seq2pair` pathway lives in the MSA module before the Pairformer trunk begins. This timing aligns with our patching results in §4.1: early trunk blocks constitute a "write-in" window during which biochemical features including charge are transferred from sequence representations into z via `seq2pair`.

In all three models, z probe accuracy declines somewhat in late blocks (35–47), suggesting charge information is progressively transformed into more geometric features as the trunk shifts to stage 2 computation. This is consistent with our finding that pairwise representations encode distance with high accuracy in late blocks (§5.2).

Summary. Charge probing supports the pathway analysis in App. I: charge information enters z during the early-block write-in window in all three models, with the location of the window tracking where each architecture implements its `seq2pair` pathway.

J.3 Charge Steering Dose-Response

We measure how cross-strand distance changes as a function of charge steering magnitude across all three models. To isolate the dose-response relationship from confounding structural variation, we use small single-motif proteins for this experiment: short ($\leq$ 40 residue) PDB proteins consisting primarily of a beta-hairpin (for same-charge steering) or a helix-turn-helix (for opposite-charge steering), rather than the larger, more diverse proteins used elsewhere in the paper. This lets us measure the geometric effect of charge steering without variability from surrounding structural context.

For each model, we apply four steering conditions: same-charge steering on beta-hairpin targets (Both +, Both $-$), expecting strands to repel; and opposite-charge steering on α-helical targets (+/$-$, $-$/+), expecting strands to attract. Steering magnitude is reported in units of the standard deviation of residue projections onto the model's charge direction.

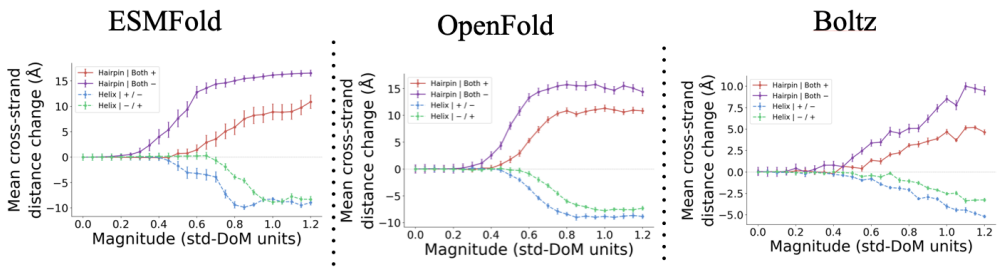


Figure 17: **Charge steering produces a smooth, monotonic dose-response across all three models.** Mean cross-strand distance change as a function of steering magnitude (in units of std-DoM) on small single-motif target proteins ($\leq$ 40 residues). Same-charge steering on hairpin targets (red, purple) increases cross-strand distance; opposite-charge steering on helical targets (blue, green) decreases it. All three models show effects emerging above a threshold around 0.4 std-DoM units and saturating thereafter, showing graded sensitivity to this charge-related latent direction in these structural contexts. Effect magnitudes are largest in OpenFold, intermediate in ESMFold, and smallest in Boltz-1, consistent with the location of the sequence-to-pair pathway in App. I: Boltz-1's seq2pair pathway lives upstream of the trunk, so trunk-level steering has a smaller direct effect.

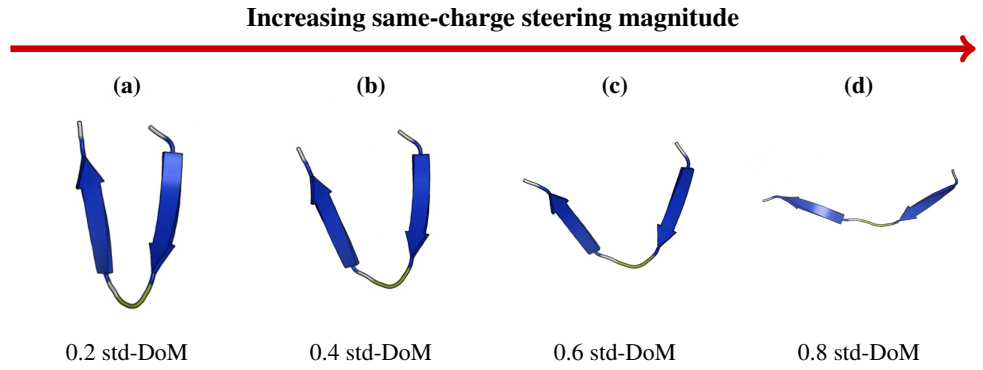


Figure 18: **Progressive separation of beta-hairpin strands under increasing same-charge steering (Pos-Pos).** Predicted structures of a small beta-hairpin target protein under same-charge steering at four magnitudes (0.2, 0.4, 0.6, 0.8 std-DoM). At low magnitude (a), the hairpin retains its native folded geometry with strands in close cross-strand contact. As steering magnitude increases (b–d), the strands progressively separate under the latent intervention, producing the dose-response curve quantified in Fig. 17.

The same dose-response shape appears in all three models: a near-zero region below threshold, a monotonic rise (or fall) through intermediate magnitudes, and saturation at high magnitudes. The threshold and saturation magnitudes are similar across models. The smaller effect size in Boltz-1 is consistent with the location of the sequence-to-pair pathway in App. I: charge written into the sequence track at trunk blocks 0–3 has limited downstream influence on the Pairformer's pairwise representation, since Boltz-1 lacks a seq2pair write-back within the trunk. Fig. 18 visualizes this gradual separation of beta hairpin strands via the charge steering.

K Generalization of Late-Block Geometric Computation Across Architectures

In §5.1, we showed that two pathways read z as geometric information in ESMFold's late trunk: contact information can be read from the `pair2seq` attention bias, and scaling z changes distances in the decoded structure. Here we replicate both analyses on OpenFold and Boltz-1. Results are summarized in Fig. 19.

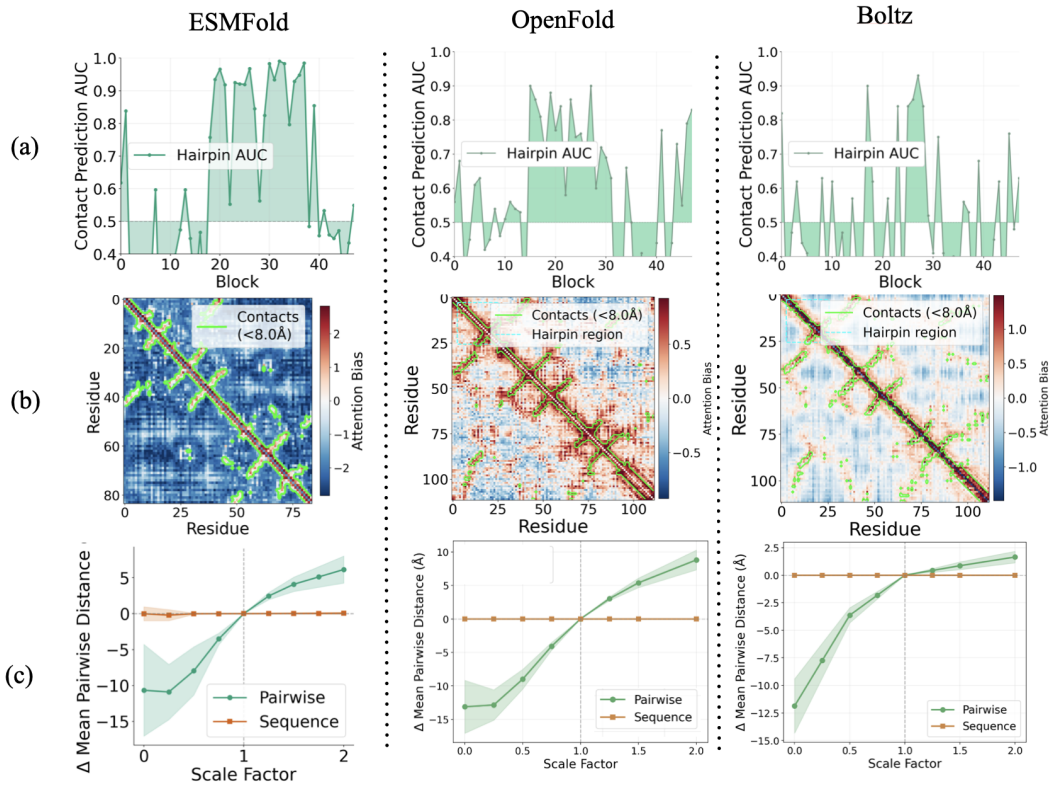


Figure 19: **Late-block pathway analysis across ESMFold, OpenFold, and Boltz-1. (a)** ROC-AUC for classifying contacting (Cα < 8Å) versus non-contacting residue pairs from `pair2seq` bias values alone, computed per block. All three models exhibit a middle-to-late block window in which the bias cleanly separates contacts from non-contacts. **(b)** Head-averaged `pair2seq` bias for an example protein at a representative late block. Heatmap entries are bias values (red = positive, blue = negative); green contours mark residue pairs within 8Å. Positive bias concentrates at spatial contacts in all three models. **(c)** Effect of scaling the pairwise (z, green) or sequence (s, orange) representation by a factor in $[0, 2]$ before it enters the structure module. Mean pairwise Cα distance is shown as deviation from the unscaled baseline (scale = 1.0); shaded regions indicate one standard deviation across the test set. Scaling z monotonically scales output distances in all three models; scaling s has virtually no effect.

`pair2seq` **makes contacts accessible to sequence attention (Fig. 19a,b).** For each block in each model, we extract the bias term that modulates sequence attention from the pairwise representation and measure how well its magnitude separates contacting from non-contacting residue pairs (Cα distance < 8Å), reporting ROC-AUC. All three models show a clear middle-to-late block window in which the bias cleanly separates contacts. ESMFold's separation is the sharpest, with AUC approaching 1.0 in late blocks; OpenFold and Boltz-1 reach AUC $\approx$ 0.9 over a similar block range. Visualizing head-averaged bias values alongside contact maps for a representative protein (panel b) confirms that positive bias concentrates at spatial contacts in all three models, with the off-diagonal contact structure clearly visible. Attention has access to this contact information in all three models, although these readouts do not measure its contribution to the final structure.

The structure module reads z as a geometric signal (Fig. 19c). We replicate the scaling analysis from §5.1 on OpenFold and Boltz-1: we multiply either the pairwise representation z or the sequence representation s by a factor in $[0, 2]$ immediately before the structure module, holding the other fixed, and measure the change in mean pairwise Cα distance. All three models exhibit the same qualitative behavior: scaling z produces a monotonic, approximately linear change in output distance, while scaling s produces no detectable change. The magnitude of the effect varies between models. ESMFold shows an intermediate effect, with mean pairwise distance changing by roughly 15Å across the scaling range. OpenFold shows the largest swing (about 20Å swing) and Boltz-1 the smallest

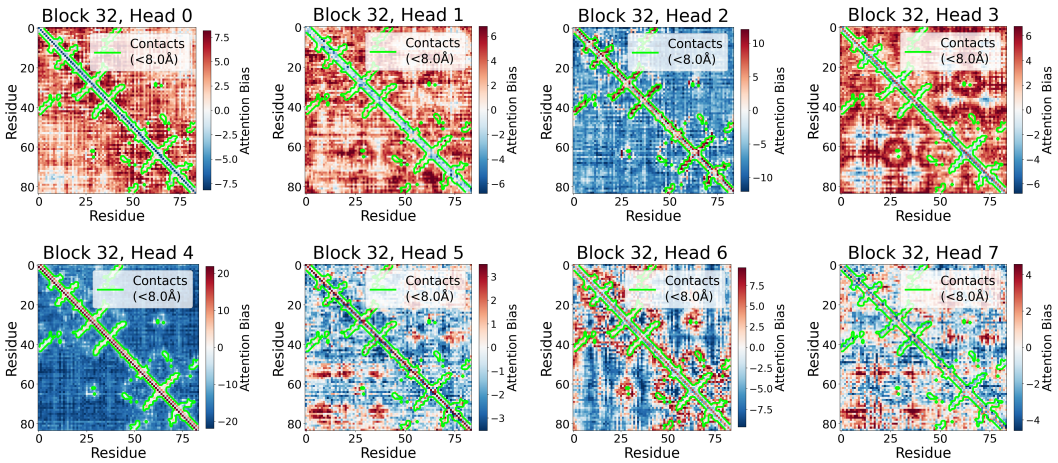


Figure 21: **Per-head `pair2seq` bias at block 32.** Individual attention head bias values for beta sheet protein 6rwc at block 32. Green contours mark structural contacts (Cα $<$ 8Å). Different heads exhibit distinct patterns: some heads show strong contact-aligned bias (e.g., heads that highlight the off-diagonal contact structure), while others capture different spatial relationships or show more diffuse patterns. The different bias patterns suggest possible head specialization; testing its effect on attention and structure requires further intervention.

(about 13Å swing, but with most of the change concentrated below scale = 1). The direction and qualitative shape are preserved across architectures: this protocol detects greater sensitivity to scaling z than s. It does not exclude geometric information or causal contributions in s.

Summary. Together, these analyses establish that the stage-2 picture from ESMFold generalizes across all three folding trunks. In all three models, contact information is present in the attention bias, and scaling the pairwise input changes output distances.

K.1 Bias Maps

We visualize the `pair2seq` attention bias (Section 5.1) across ESMFold blocks and across individual attention heads. For a single protein (PDB: 6rwc), we extract the bias term $\beta_{ij}(z_{ij})$ from Eq. (1) and average over heads. Green contours indicate structural contacts (Cα distance $<$ 8Å).

L All-Pairs Cross-Model Alignment and Patching

In §6, we showed that pairwise representations from OpenFold and Boltz-1 can be linearly aligned to ESMFold's space and substituted into ESMFold at the corresponding stage, recovering the late-block patching window. Here we extend this analysis to all three pairings of folding architectures (ESMFold $\leftrightarrow$ OpenFold, ESMFold $\leftrightarrow$ Boltz-1, OpenFold $\leftrightarrow$ Boltz-1) and to both directions of each pairing. Results are summarized in Fig. 24.

CKA across all pairings. The top row of Fig. 24 extends the CKA analysis from §6 to the OpenFold $\leftrightarrow$ Boltz-1 pairing. All three pairings show high CKA throughout middle and late blocks: late blocks of one model are most similar to late blocks of another. ESMFold $\leftrightarrow$ OpenFold shows the highest overall similarity (CKA > 0.8 throughout the middle of the trunk), consistent with their shared lineage from the AlphaFold2 architecture. Pairings involving Boltz-1 show lower CKA (typically 0.5–0.8), showing shared linear structure despite the larger architectural differences involving Boltz-1.

Procrustes alignment in both directions. The middle row reports per-block test R^2 of whitened Procrustes projections in both directions of each pairing. Alignment is poor at the earliest measured block, but rises rapidly through blocks 5–15 and plateaus thereafter. Plateau values are highest for ESMFold $\leftrightarrow$ OpenFold ($R^2 \approx 0.9$ in both directions), intermediate for OpenFold $\leftrightarrow$ Boltz-1 ($R^2 \approx 0.8$–0.9), and lowest for ESMFold $\leftrightarrow$ Boltz-1 ($R^2 \approx 0.7$–0.8). The two directions of

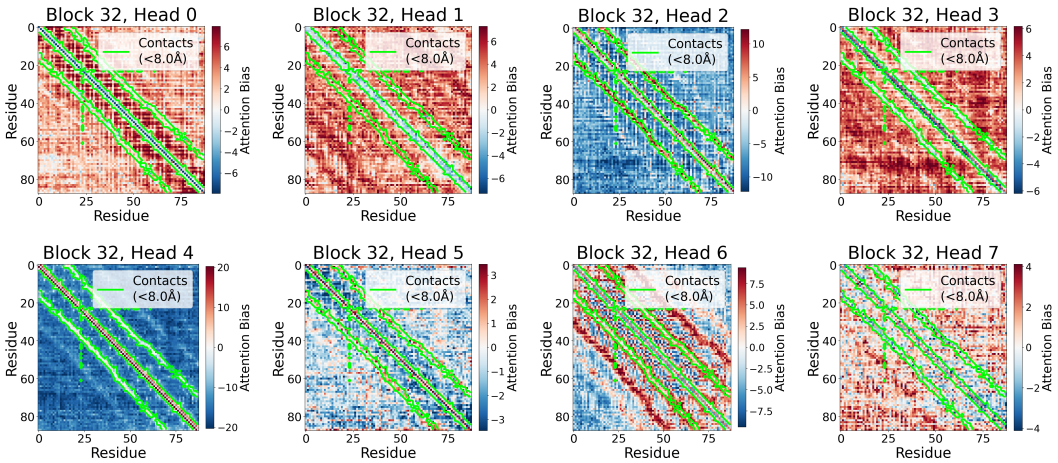


Figure 22: **Per-head `pair2seq` bias at block 32.** Individual attention head bias values for beta protein 1l0s at block 32. Green contours mark structural contacts (Cα $<$ 8Å). Different heads exhibit distinct patterns: some heads show strong contact-aligned bias (e.g., heads that highlight the off-diagonal contact structure), while others capture different spatial relationships or show more diffuse patterns. The different bias patterns suggest possible head specialization; testing its effect on attention and structure requires further intervention.

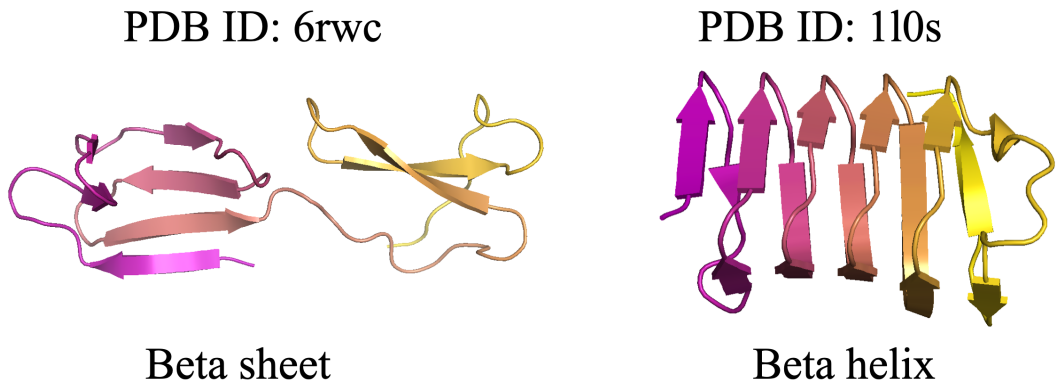


Figure 23: **Example proteins used throughout the analysis.** PyMOL visualization of PDB 6rwc, a beta-sheet protein with antiparallel strands connected by turns, and PDB: 1l0s, a left-handed beta-helix with repeating triangular coils. These two structures illustrate distinct beta-rich folds and serve as representative examples for the pair2seq bias visualizations in Appendix K.1.

each pairing track each other closely. Shuffled-correspondence baselines (dotted lines) perform poorly relative to true correspondences; their absolute values depend on the alignment and centering convention (App. L.1).

A few panels show small positive R^2 at very early blocks (e.g., OpenFold $\rightarrow$ ESMFold and Boltz-1 $\rightarrow$ ESMFold near block 0–5). We attribute these to coincidental alignment of model-specific positional embedding structure that is recoverable by a linear map. These early-block alignments do not translate into successful patching (bottom row), consistent with the interpretation that they reflect surface-level structure rather than the shared computational geometry that develops in middle and late blocks.

Cross-model patching across all directions. The bottom row applies the cross-model patching protocol from §6 to all six donor–receiver directions. In every direction, patching reproduces the late-block pairwise window observed in within-model patching: cross-model patches induce the target motif only when applied to blocks 25–40 of the receiving model, matching the pairwise stage identified in §3 and App. I. Patching strength tracks alignment quality in the middle row. The highest

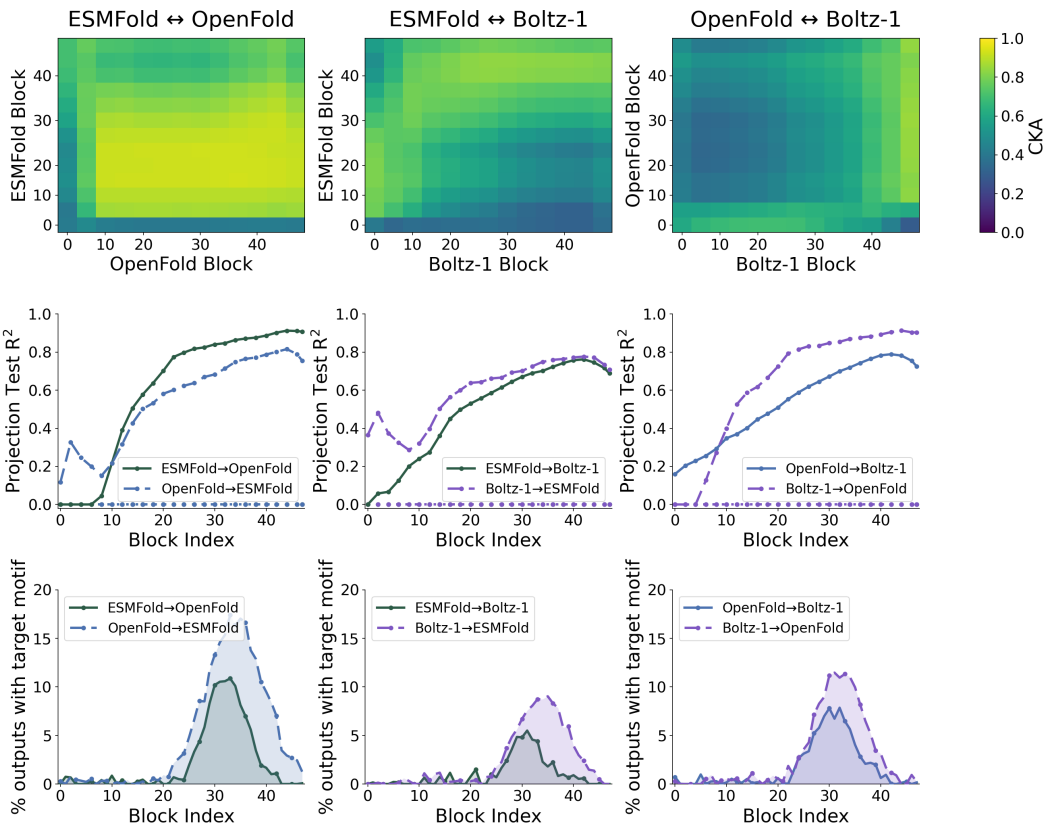


Figure 24: **Pairwise representation alignment and cross-model patching across all three model pairings.** Columns correspond to the three pairings: ESMFold ↔ OpenFold (left), ESMFold ↔ Boltz-1 (middle), OpenFold ↔ Boltz-1 (right). **Top row:** CKA between pairwise representations at every block pair. Similarity is high throughout middle and late blocks for all three pairings, concentrating along the diagonal. ESMFold ↔ OpenFold shows the highest overall similarity; pairings involving Boltz-1 are lower but still substantial. **Middle row:** Test R^2 of whitened Procrustes projections in both directions of each pairing, evaluated per block. Solid and dashed lines correspond to the two directions; dotted lines at the bottom of each panel are shuffled-correspondence baselines, which sit near zero across all blocks. **Bottom row:** Single-block pairwise patching success when donor representations come from one model (after Procrustes projection into the receiving model's space) and are patched into the other. Cross-model patching reproduces the late-block pairwise window in all six directions, with patching strength tracking projection quality from the middle row.

cross-model patching rates occur for ESMFold ↔ OpenFold (peaking near 17% for OpenFold → ESMFold and 11% for ESMFold → OpenFold). Pairings involving Boltz-1 show lower patching rates (typically 5–12%), consistent with their lower projection R^2, but the late-block timing window is preserved in all six cases.

Asymmetry between directions. For each pairing, the two directions of patching produce substantially different success rates even when projection R^2 is similar in both directions. We do not have a definitive explanation for this asymmetry, but plausible factors include differences in the receiving model's tolerance for off-distribution pairwise inputs, differences in how each model's downstream pathways read z, and small numerical differences in how the projection is conditioned. Matched calibration and decoder swaps (App. Q) further characterize this asymmetry, while leaving its full cause unresolved.

Summary. Across all three pairings of folding architectures and all six donor–receiver directions, pairwise representations are linearly alignable, and aligned representations are partially functionally transplantable after linear alignment at the corresponding stage of computation. Combined with

the staged structure (App. I) and stage-specific causal features (charge in §4.2, distance in §5.2) replicating across models, this supports the interpretation that folding trunks converge on a shared representational organization for the late-block pairwise stage, despite differences in training data, architecture, and input modality.

L.1 Comparison of Linear Alignment Methods

In §6, we used whitened Procrustes alignment to learn linear projections between models' pairwise representations. Here we compare against four alternatives under a shuffled-correspondence control.

Methods. Ridge regression (unconstrained linear with L2), CCA, unwhitened Procrustes (orthogonal rotation), whitened Procrustes (the method used in §6), and random orthogonal as a floor baseline.

Shuffled control. For each method, we randomly shuffle residue-pair correspondences between Z_A and Z_B before alignment, then evaluate test R^2. A method exploiting genuine shared structure should perform poorly under shuffling; a method fitting model-specific artifacts (marginal statistics, dimensionality) can still achieve nontrivial R^2.

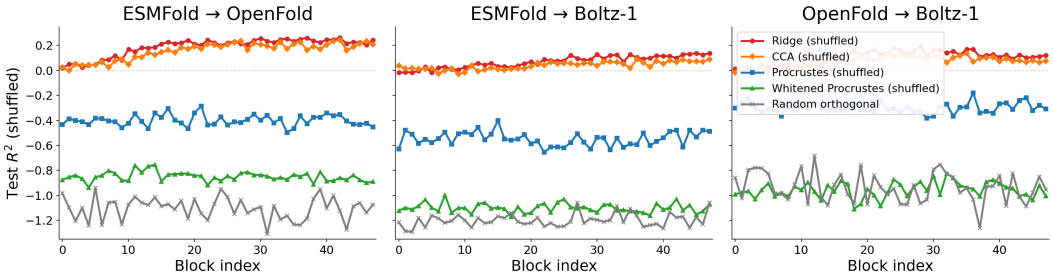


Figure 25: **Test R^2 under shuffled correspondences for five alignment methods.** Ridge and CCA achieve mildly positive shuffled R^2, indicating they fit marginal statistics. Procrustes-family methods produce negative R^2 under shuffling because the rotation constraint cannot recover the target mean; the more constrained the method, the more negative the shuffled R^2 can go. Whitened Procrustes sits near the random-orthogonal floor, indicating its rotation contains essentially no information transferable to shuffled data.

Ridge and CCA achieve shuffled R^2 around 0.1–0.25 across blocks. Whitened Procrustes sits near the random-orthogonal floor ($R^2 \approx -1$), with unwhitened Procrustes intermediate. Combined with the high true-correspondence R^2 reported in §6 ($R^2 \approx 0.9$ for ESMFold $\leftrightarrow$ OpenFold), this is why we use whitened Procrustes: it is the most constrained method that still aligns representations with high fidelity, and its near-floor shuffled control indicates the learned rotation is specific to genuine cross-model structure rather than model-specific artifacts.

M Scope, Conditioning, and Extended Interventions

All-pair evaluation. The main single-block analysis uses donor–target pairs for which full patching succeeds. Table 2 compares that subset with all donor–target pairs. Each entry is the maximum per-block induction rate within the specified early (0–11) or late (36–47) region, rather than the maximum across the entire trunk. These regional peaks therefore need not equal the global peaks in other tables. Including all pairs lowers success rates. Sequence patches still act early in ESMFold and OpenFold, while pairwise patches remain more effective in Boltz-1. In Boltz-1, pairwise patches act over a broad interval rather than only in late blocks.

Larger motifs and tertiary contacts. For four-strand sheets, we use 200 PISCES donor proteins and 100 size-matched targets of 400–600 residues. Donor sheets span 40–60 residues and are patched into non-beta target regions. We report intended-topology formation separately from DSSP beta content: increased beta content alone does not establish the intended sheet. For tertiary contacts, we use 300 donor–target pairs with contacting donor strands and non-contacting target segments. The median sequence separation is 132 residues (range 59–254), and the mean initial spatial separation is 24.5Å. Sequence patches replace the selected segment states; pairwise patches replace cross-segment

Table 2: Single-block induction rates (%) with and without conditioning on full-patch success. S: sequence; P: pairwise.

Model	Cases	S early	S late	P early	P late
ESMFold	Transferable subset	40.0	0.0	0.2	25.0
	All pairs	36.8	0.0	0.1	15.1
OpenFold	Transferable subset	40.0	0.0	0.2	34.2
	All pairs	33.4	0.0	0.2	29.6
Boltz-1	Transferable subset	8.7	0.0	30.3	35.4
	All pairs	5.7	0.0	25.6	27.5

states. Contact success requires the intended Cα pair to lie within 8Å. Table 3 reports both tasks, with matched contact controls in Tables 7–8.

Table 3: Extended interventions. Entries separated by a slash are sequence/pairwise results. Sheet beta-content increases and strict topology success are distinct outcomes.

Task	Model	Effective blocks	Peak effect	Success (%)
Sheet	ESMFold	0–6 / 21–42	$+72\%/+65\%$ beta	32 / 21
	OpenFold	0–4 / 19–45	$+74\%/+81\%$ beta	37 / 25
	Boltz-1	0–3 / 7–47	$+52\%/+83\%$ beta	15 / 28
Contact	ESMFold	0–11 / 24–47	$-15.5/-17.3$Å	47 / 46
	OpenFold	0–15 / 22–47	$-15.2/-17.7$Å	55 / 52
	Boltz-1	0–2 / 10–47	$-16.0/-18.5$Å	12 / 73

Length and motif-independent timing. The original target proteins contain 100–400 residues. A length-stratified extension uses five bins with 60, 60, 40, 30, and 20 proteins, respectively, up to 1,000 residues. Comparing the shortest (75–250) and longest (801–1,000) bins, full-patch success declines from 43% to 28% for ESMFold, 41% to 26% for OpenFold, and 38% to 25% for Boltz-1. ESMFold's sequence window changes from 0–7 to 0–9 and its pairwise window from 27–37 to 25–35; OpenFold has the same reported windows. Their longest-bin sequence/pairwise peaks are 16.2%/3.6% and 11.3%/11.8%, respectively. Boltz-1 has no detected sequence-patching effect in this extension; its pairwise effects occupy 0–3 and 24–36 in the shortest bin and 0–3 and 23–34 in the longest, with a longest-bin peak of 18.2%. Thus, longer proteins retain some intervention timing but provide weaker evidence for sequence sufficiency, especially in Boltz-1.

Separately, pathway-update norms measured across all residues of 1,000 diverse proteins (75–1,000 residues, including repeat, coiled-coil, and multidomain proteins) have sequence-to-pair peak/center-of-mass block indices of 2/6.4, 2/5.2, and 1/1.1 for ESMFold, OpenFold, and Boltz-1. Corresponding pair-to-sequence indices are 34/25.6, 32/26.3, and 11/22.5. This motif-independent measurement supports earlier sequence-to-pair updates without establishing causal use from update magnitude alone.

Boltz-1 indexing and interpretation. The upstream MSA module is indexed as blocks 0–3 in this analysis. After filtering, Boltz-1 pairwise patches transfer hairpins from blocks 3–47. The effect begins near the MSA–Pairformer boundary and continues through the subsequent pairwise processing. Sequence-to-pair freezing and ablation, donor-directed changes in z, and early biochemical steering provide evidence for functional sequence-to-pair transfer in the upstream module. Sequence patches produce the donor structure less often than in ESMFold or OpenFold. Thus, the models can perform related computations in the same order while differing in the blocks involved and the success of an intervention.

N Structural Quality, Designability, and Contact Specificity

Evaluation criteria. A chain is intact when no adjacent Cα–Cα distance exceeds 4.2Å. A structure is clash-free when it has at most two non-adjacent heavy-atom pairs within 2Å. Model confidence

is reported as mean pLDDT on a 0–1 scale. ProteinMPNN [Dauparas et al., 2022] designs eight sequences per backbone; the best-of-eight design/refold self-consistency TM-score is denoted scTM, with scTM ≥ 0.5 used for the designability filter. The cross-model check refolds designed sequences with the two non-source folding models and reports their mean scTM. Motif recovery is conditional on successful patches and measures recovery within the intervention window. These checks ask whether the output has plausible geometry and whether folding models can recover it from designed sequences. They do not establish experimental stability.

Table 4: Independent-refolding evaluation, $n = 2,000$ outputs per model/condition. Rates are percentages; brackets give reported 95% intervals. "Cross" is cross-model scTM. All-block patching is more disruptive than a single-block intervention.

Model	Condition	Success	Intact	Clash-free	Cross
ESMFold	Baseline	0 [0.0, 0.2]	100 [99.8, 100]	93 [91.8, 94.0]	.97 [.967, .973]
	Single-block	40 [37.9, 42.2]	97 [96.2, 97.7]	80 [78.2, 81.7]	.93 [.925, .935]
	All-block	42 [39.9, 44.2]	83 [81.3, 84.6]	63 [60.9, 65.1]	.91 [.904, .916]
OpenFold	Baseline	0 [0.0, 0.2]	97 [96.2, 97.7]	80 [78.2, 81.7]	.96 [.956, .964]
	Single-block	40 [37.9, 42.2]	90 [88.6, 91.2]	63 [60.9, 65.1]	.94 [.935, .945]
	All-block	38 [35.9, 40.1]	67 [64.9, 69.0]	50 [47.8, 52.2]	.90 [.893, .907]
Boltz-1	Baseline	0 [0.0, 0.2]	100 [99.8, 100]	87 [85.5, 88.4]	.98 [.977, .983]
	Single-block	40 [37.9, 42.2]	100 [99.8, 100]	83 [81.3, 84.6]	.96 [.956, .964]
	All-block	41 [38.9, 43.2]	100 [99.8, 100]	70 [68.0, 72.0]	.95 [.945, .955]

Model	Condition	pLDDT	scTM	Motif recovery (%)
ESMFold	Baseline	.92 [.916, .924]	.99 [.988, .992]	–
	Single-block	.87 [.865, .875]	.95 [.946, .954]	92 [89.9, 93.7]
	All-block	.82 [.814, .826]	.93 [.925, .935]	100 [99.5, 100]
OpenFold	Baseline	.85 [.844, .856]	.98 [.977, .983]	–
	Single-block	.90 [.895, .905]	.96 [.956, .964]	94 [92.1, 95.4]
	All-block	.81 [.803, .817]	.94 [.935, .945]	100 [99.5, 100]
Boltz-1	Baseline	.92 [.916, .924]	.99 [.988, .992]	–
	Single-block	.84 [.834, .846]	.98 [.977, .983]	96 [94.4, 97.2]
	All-block	.83 [.824, .836]	.97 [.966, .974]	100 [99.5, 100]

Rate intervals in Table 4 are Wilson 95% intervals; continuous-metric intervals are obtained by resampling independent protein–target cases. Table 5 gives the separate steering and cross-model-transplant evaluations. Sample sizes and interventions differ, so its rates should not be interpreted as a matched comparison with the single-block cohort above.

Table 5: Quality of steering and transplantation outputs. Success is shown with 95% interval half-widths. Intact, clash-free, and recovery are percentages; pLDDT and scTM are means. Recovery is conditional on successful interventions.

Model	Intervention	N	Success (%)	Intact	Clash-free	pLDDT	scTM	Recovery
ESMFold	Charge	400	20 ± 3.9	72	68	.83	.75	77
	Distance	600	18 ± 3.1	67	72	.85	.78	89
	Cross-model	4,000	16 ± 1.1	83	75	.78	.95	85
OpenFold	Charge	400	20 ± 3.9	77	65	.74	.90	75
	Distance	600	17 ± 3.0	80	68	.72	.90	72
	Cross-model	4,000	14 ± 1.1	91	72	.81	.96	100
Boltz-1	Charge	400	8 ± 2.7	95	83	.82	.84	62
	Distance	600	16 ± 2.9	92	75	.79	.93	87
	Cross-model	4,000	9 ± 0.9	93	91	.84	.97	91

Targeted contact transfer versus collapse. The long-range experiment is repeated with a donor whose corresponding segments do not contact, and with a random pairwise perturbation matched to the donor patch's norm and mask. Tables 7 and 8 report the same conditions. S and P denote sequence and pairwise patching. Donor-contact precision is the fraction of newly formed cross-segment contacts present in the donor. Off-target density measures tertiary contact-density change outside the targeted cross-segment pairs. "Contacting" denotes the original donor condition; "non-contacting"

Table 6: Localization after structural-validity and designability filtering. Peaks are induction rates (%). The Boltz-1 pairwise interval is 3–47. Each interval applies to the stated intervention and mask.

Model	Patch	Raw peak	Validity-filtered	Designability-filtered	Window
ESMFold	Sequence	40	37	35	0–6
	Pairwise	25	21	22	19–42
OpenFold	Sequence	41	39	34	0–7
	Pairwise	34	26	24	18–43
Boltz-1	Sequence	15	10	7	0–3
	Pairwise	35	33	33	3–47

denotes the matched donor control. Continuous quantities are shown with 95% interval half-widths; rates use Wilson intervals.

Table 7: Long-range contact controls: targeted and global distance changes. Rg is radius of gyration; all distances are in Å.

Model	Condition	$\Delta d_{\text{intended}}$	Contact success (%)	ΔRg	Δd_{chain}
ESMFold	Contacting S	-15.51 ± 5.13	47.7 ± 5.8	$-.65 \pm .74$	$-.46 \pm .21$
	Contacting P	-17.32 ± 3.56	46.1 ± 4.9	$-.15 \pm .17$	$-.14 \pm .19$
	Non-contacting S	6.72 ± 1.54	0 [0, 3]	$1.87 \pm .50$	$2.65 \pm .55$
	Non-contacting P	5.28 ± 1.31	0 [0, 2]	$1.01 \pm .66$	2.01 ± 1.06
	Random P	-5.63 ± 2.82	7.8 ± 4.3	$-3.19 \pm .23$	$-2.21 \pm .28$
OpenFold	Contacting S	-15.20 ± 3.17	55.5 ± 4.2	$.22 \pm .27$	$.35 \pm .33$
	Contacting P	-17.70 ± 3.28	52.3 ± 6.6	$-.41 \pm .24$	$-.05 \pm .28$
	Non-contacting S	$4.20 \pm .96$	0 [0, 6]	$.51 \pm .27$	$.69 \pm .35$
	Non-contacting P	4.36 ± 1.02	0 [0, 7]	$1.05 \pm .19$	$1.06 \pm .24$
	Random P	-8.79 ± 3.14	9.2 ± 3.1	-2.44 ± 2.21	-1.79 ± 2.84
Boltz-1	Contacting S	-16.04 ± 4.47	12.4 ± 4.7	$-.42 \pm .14$	$-.12 \pm .05$
	Contacting P	-18.53 ± 5.85	73.2 ± 7.3	$-.22 \pm .11$	$-.27 \pm .12$
	Non-contacting S	$3.04 \pm .04$	0 [0, 3]	$1.05 \pm .13$	$1.07 \pm .13$
	Non-contacting P	3.03 ± 1.92	0 [0, 4]	$2.31 \pm .97$	$1.21 \pm .57$
	Random P	-6.03 ± 1.23	6.0 ± 2.4	$-3.24 \pm .07$	-1.32 ± 1.08

Table 8: Long-range contact controls: specificity for the donor pattern and off-target changes. Density changes are in percentage points (pp). Rows correspond to Table 7.

Model	Condition	Donor-contact precision	Δ off-target density (pp)
ESMFold	Contacting S	$.92 \pm .06$	$.3 \pm .2$
	Contacting P	$.87 \pm .08$	$.1 \pm .1$
	Non-contacting S	$.00 \pm .00$	$-1.2 \pm .4$
	Non-contacting P	$.00 \pm .00$	$-.9 \pm .5$
	Random P	$.12 \pm .04$	$4.1 \pm .9$
OpenFold	Contacting S	$.88 \pm .07$	$-.1 \pm .2$
	Contacting P	$.79 \pm .13$	$.1 \pm .2$
	Non-contacting S	$.00 \pm .00$	$-.4 \pm .3$
	Non-contacting P	$.00 \pm .00$	$-.5 \pm .3$
	Random P	$.13 \pm .09$	$-.8 \pm .8$
Boltz-1	Contacting S	$.76 \pm .12$	$.4 \pm .2$
	Contacting P	$.83 \pm .22$	$.2 \pm .2$
	Non-contacting S	$.00 \pm .00$	$-.6 \pm .2$
	Non-contacting P	$.00 \pm .00$	$-.7 \pm .4$
	Random P	$.19 \pm .09$	$.8 \pm .3$

Contacting donors recover the donor contact pattern with precision .76–.92 and modest whole-chain changes ($|\Delta\text{Rg}| \leq .65$Å). Random perturbations produce lower contact success and precision, with larger whole-chain contraction. Non-contacting donors increase segment separation. The donor patches therefore do more than pull the protein together: they preferentially recover the intended contacts. Local distortions caused by the intervention can still occur.

O Biochemical Steering Controls

Charge controls. Norm- and variance-matched random directions use the same steering strengths, residue positions, and block windows as the charge intervention. Charge-steering peaks are 20%, 20%, and 8% for ESMFold, OpenFold, and Boltz-1, whereas random directions have median peak effects at most .2% and 95th-percentile effects below 1% across models. Together with sign-sensitive attraction and repulsion, this argues against arbitrary perturbation as the sole explanation. The direction can nevertheless encode charge together with correlated residue properties.

An uncharged-residue hydrophobicity construction. To separate hydrophobicity from charged-residue labels, we construct a difference-of-means direction from two sets containing only uncharged residues: $\{\mathrm{I}, \mathrm{V}, \mathrm{L}, \mathrm{F}, \mathrm{M}, \mathrm{A}\}$ and $\{\mathrm{N}, \mathrm{Q}, \mathrm{S}, \mathrm{T}, \mathrm{G}\}$. On 500 helical targets, we steer the originally hydrophobic face toward the polar direction and the originally polar face toward the hydrophobic direction, then measure DSSP-derived solvent exposure. Table 9 gives the results. Late-block and matched-random interventions do not produce the systematic exposure reversal. The effect persists without charged residues in either label set, supporting a hydrophobicity-related feature. Other properties of these residues may also contribute.

Table 9: Steering from uncharged residue sets, $N = 500$ helical targets. Exposure rates are percentages. Parentheses give change in percentage points with reported 95% interval half-widths.

Model	Effective blocks	Originally hydrophobic face	Originally polar face
ESMFold	0–18	$23 \to 66$ $(+43 \pm 6.1)$	$49 \to 26$ (-23 ± 2.3)
OpenFold	0–14	$23 \to 62$ $(+39 \pm 5.6)$	$49 \to 22$ (-27 ± 3.7)
Boltz-1	0–3	$23 \to 47$ $(+24 \pm 4.4)$	$49 \to 17$ (-32 ± 4.9)

P Coherent Scale Sensitivity and Directional Geometry Controls

Multiplying the entire pairwise input by a scalar tests the effect of its overall scale; it does not isolate one feature direction. In a magnitude-matched comparison, coherent rescaling changes mean output pairwise distance by 6.89Å, versus .05Å for Gaussian noise. The decoder therefore responds differently to uniform scaling and to noise of the same magnitude.

The learned distance direction is tested separately on the cross-strand pairs targeted in distance steering. In the reverse intervention, steering toward greater separation changes targeted distances by $+10.86$Å averaged across the three models, compared with $+.20$Å for a matched random direction. Preserving the global norm of z gives $+10.85$Å. The distance change is therefore not explained solely by an increase in the overall norm, although individual pair vectors can still change in norm. Separately, contact readouts from the attention bias show that attention has access to contact information; they do not establish how much the output depends on its use.

Q Calibration, Decoder Swaps, and Disagreement Analysis

Matched intermediate-state calibration. Each off-diagonal donor–receiver direction is evaluated on approximately 2,000 matched protein–target cases across the 48 blocks. We divide peak cross-model success by the receiver's peak within-model success, using matched cases, receivers, and masks. Bootstrap intervals resample independent matched cases and recompute the peak ratio within each replicate; block evaluations are not treated as independent samples. Table 10 shows that success depends strongly on the receiving model. Patches from the other models recover approximately 64% of ESMFold's within-model peak, 35% of OpenFold's, and 22.5% of Boltz-1's. The mean of the six cross-model ratios is 40.3%.

Transferring final states between decoders. For each source–decoder combination, we use 500 cases in which the source intervention produces a hairpin. For each case, we finish the source trunk, align its final z to the receiver basis, and pass it to the receiving decoder. This bypasses subsequent receiver-trunk computation. Table 11 reports the median Cα RMSD to the source prediction and the fraction of outputs that retain the hairpin. The 65–78% retention across off-diagonal cells

Table 10: Peak cross-model/within-model success ratios with case-bootstrap 95% intervals. Rows are donor models; columns are receivers. Diagonals are normalized references.

Donor	ESMFold	OpenFold	Boltz-1
ESMFold	1.00	.33 [.29, .37]	.20 [.17, .23]
OpenFold	.75 [.69, .81]	1.00	.25 [.21, .29]
Boltz-1	.52 [.47, .57]	.37 [.33, .42]	1.00

supports partial compatibility across IPA-based and diffusion-based readouts. It does not imply that intermediate states can be substituted freely throughout another model's trunk. Approximately 15% of failed transfers involving Boltz-1 have conformational heterogeneity, but this does not resolve the overall asymmetry. Alignment errors and differences in how models process mixtures of donor and target states may also contribute.

Table 11: Final-state decoder swaps, $n = 500$ cases per cell. Reported 95% intervals accompany RMSD (Å) and hairpin retention.

Source	Decoder	Median RMSD	Hairpin retained (%)
ESMFold	OpenFold	4.47 [4.18, 4.79]	66 [61.7, 70.0]
	Boltz-1	3.27 [3.04, 3.52]	69 [64.8, 72.9]
OpenFold	ESMFold	1.77 [1.61, 1.95]	78 [74.2, 81.4]
	Boltz-1	3.84 [3.69, 4.11]	74 [70.0, 77.7]
Boltz-1	ESMFold	3.09 [2.84, 3.36]	72 [67.9, 75.8]
	OpenFold	4.64 [4.29, 5.02]	65 [60.7, 69.1]

ESMFold2 decoder check. ESMFold2 [Candido et al., 2026] provides an additional test with a language-model encoder and a diffusion decoder. With $n = 500$ cases per source, its decoder retains hairpins from ESMFold, OpenFold, and Boltz-1 in 79% [75.2, 82.3], 70% [65.8, 73.9], and 82% [78.4, 85.1] of outputs. Median source-relative RMSDs are 2.84Å [2.60, 3.10], 3.62Å [3.34, 3.93], and 2.41Å [2.21, 2.63], respectively (95% intervals). Within-model retention is 89%. This provides an additional decoder comparison; the main analysis remains focused on ESMFold, OpenFold, and Boltz-1.

Disagreement with experimental structures. Across 1,000 proteins, 332 have at least one pair of final predicted distance maps with correlation $r < .90$. Of these, 71% have two agreeing models and one outlier; 29% have three-way disagreement. Consensus is not treated as ground truth: each prediction is compared with experimental structure. On 200 proteins with substantially divergent predictions, final-state swaps produce distance maps closer to the source than to the decoder host in every direction (Table 12). Within-model controls give $r \geq .9997$ and shuffled projections $r = .05$. The final pairwise state therefore carries substantial information about the differing predicted fold, including across decoder families.

Table 12: Decoder swaps on 200 proteins with divergent model predictions. Correlations compare output distance maps; RMSDs are in Å.

Source → decoder	r to source	r to host	RMSD to source	RMSD to host
OpenFold → ESMFold	.978	.562	1.8	19.2
Boltz-1 → ESMFold	.900	.614	3.1	9.9
OpenFold → Boltz-1	.935	.634	5.4	13.5
ESMFold → Boltz-1	.874	.605	7.1	15.4
ESMFold → OpenFold	.899	.394	8.8	19.9
Boltz-1 → OpenFold	.856	.507	11.6	17.2

A localized late-refinement failure. In a follow-up set of 100 failures arising during middle or late trunk refinement, protein 1d2o illustrates how subsequent computation can overwrite an aligned state. ESMFold predicts the experimental fold, while OpenFold and Boltz-1 predict the same alternative

fold. OpenFold's baseline agreement with experimental distances is .52. Patching ESMFold's state into OpenFold at blocks 0, 6, 12, 18, and 24 gives agreement .53, .55, .54, .54, and .54; patching at blocks 28, 36, and 47 gives .98, 1.00, and 1.00. Boltz-1's state, which encodes the same incorrect fold as OpenFold, gives .52–.53 throughout. In this example, the correct donor geometry survives only when introduced at block 28 or later. That boundary is specific to this protein and intervention. On 39 proteins where ESMFold2 disagrees with the original models, swaps in both directions likewise yield outputs closer to the source fold (median RMSD 1.88–4.86Å) than the host fold (9.17–12.17Å); shuffled controls give 15–31Å.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the paper's contributions: a recurring functional ordering across tested interventions, biochemical and distance-directed steering, and partial cross-model transfer.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Sec. 8, on the following topics: diversity of structural motifs analyzed, analysis of pairwise-representation-free architectures, and physical viability of counterfactual structures.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper is empirical and mechanistic rather than theoretical, and it does not present formal theorems requiring assumptions or proofs.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: The paper describes the principal protocols, datasets, and evaluation criteria. Some settings for the added validation controls and per-cell sample counts are not fully specified, so complete reproduction is not yet supported by this revision.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We provide a public code repository (App. C) that reproduces the main ESMFold experiments via a `reproduce.sh` entrypoint. We release code for a single model to keep the reproduction environment lightweight; the OpenFold and Boltz-1 replications apply the same methodology to publicly available model checkpoints, and full coverage of the added multi-model controls is not established by that release.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No]

   Justification: The original interventions include models, datasets, and strengths. Several added controls lack complete exposure criteria, decoder-state settings, or rescaling parameters in the available records.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Tables 4, 10, and 11 report 95% intervals. Rates use Wilson intervals where specified; transfer calibration resamples matched cases rather than blocks. Some exploratory results lack intervals.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: App. C reports original A6000 hardware and approximate costs, and identifies the H100 length extension. A complete budget for added validation experiments is not reported.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics, as it analyzes existing protein-folding models and public protein-structure data while discussing responsible-use considerations.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The Impact Statement in Appendix B discusses positive uses for mechanistic understanding and reliability, as well as possible dual-use concerns if interpretability insights accelerate harmful protein engineering.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a new high-risk model or dataset requiring special safeguards, and the methods are analytical rather than a new predictive or generative capability.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Appendix E documents the licenses and usage terms for the existing models and data resources used in the paper, including ESMFold/ESM, OpenFold, Boltz-1, PDB, CATH, UniProt, and DSSP

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper curates experimental subsets but does not introduce or release a new standalone asset requiring separate documentation.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, human-subject experiments, participant instructions, or compensation.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, so IRB approval or equivalent review is not applicable.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as an important, original, or non-standard component of the core research methodology described in the paper.