MECHANISMS OF AI PROTEIN FOLDING IN ESMFOLD

Anonymous authors
Paper under double-blind review

ABSTRACT

How do protein structure prediction models fold proteins? We investigate this question by tracing how ESMFold folds a beta hairpin, a prevalent structural motif. Through counterfactual interventions on model latents, we identify two computational stages in the folding trunk. In the first stage, early blocks initialize *pairwise biochemical signals*: residue identities and associated biochemical features like charge flow from sequence representations into pairwise representations. In the second stage, late blocks develop *pairwise spatial features*: distance and contact information accumulate in the pairwise representation. We demonstrate that the mechanisms underlying structural decisions of ESMFold can be localized, traced through interpretable representations, and manipulated with strong causal effects.

1 INTRODUCTION

How do protein folding models fold proteins? Neural networks such as AlphaFold (Jumper et al., 2021) and ESMFold (Lin et al., 2023a) predict three-dimensional structures directly from amino acid sequences, but the internal computations that produce a fold remain poorly understood. Like AlphaFold2 and Boltz-2 Passaro et al. (2025), ESMFold is built around a *folding trunk* that iteratively refines a sequence representation while building a pairwise representation encoding relationships between residues; a structure module then decodes these into 3D coordinates.

We conduct the first mechanistic analysis of a folding trunk, focusing on how ESMFold folds beta hairpins: a minimal motif requiring coordination between sequence-distant residues. Using activation patching Vig et al. (2020); Meng et al. (2022), we transplant representations from a donor protein containing a hairpin into the forward pass of a target protein containing a helical region, varying *what* is patched (sequence versus pairwise representations) and *where* (early versus late blocks). (Fig. 1).

We find two computational stages in the folding trunk. In the first stage, early blocks initialize *pairwise biochemical signals*: residue identities and features like charge flow from sequence into pairwise representations. In the second stage, late blocks develop *pairwise spatial features*: distance and contact information accumulate in the pairwise representation. We show that molecular features

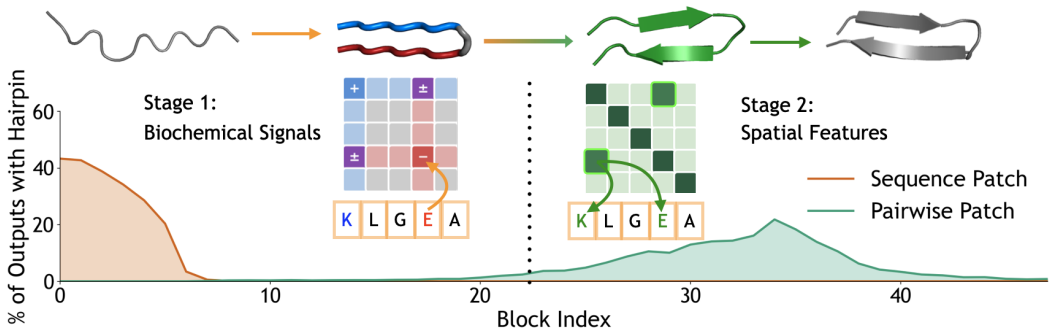


Figure 1: **Two computational stages in the ESMFold folding trunk.** We identify which latent representations in the model influence hairpin formation by patching activations from a hairpin protein into a helical protein at each block of the trunk, then measuring whether the output folds as a hairpin. Sequence patches (orange) induce hairpin formation in early blocks (0–7); pairwise patches (green) are effective in late blocks, with results aggregated over 2000 experiments.

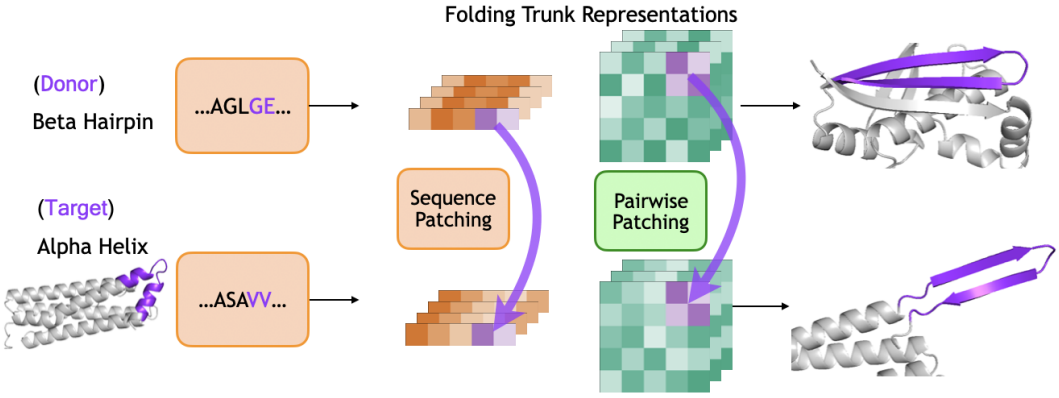


Figure 2: **Activation patching setup.** We extract sequence (orange) and pairwise (green) representations of a donor hairpin and patch them into the corresponding positions of a helical target during its forward pass. Purple highlights the intervened region.

such as charge are encoded in interpretable linear directions and causally influence folding, and that the pairwise representation functions as a distance map controlling output geometry. Together, these results demonstrate that ESMFold's structural decisions can be localized, traced through interpretable representations, and manipulated with strong causal effects.

2 When Does the Model Fold a Hairpin?

We transplant hairpin representations into helical proteins via activation patching (Fig. 2). We run both a donor (containing a beta hairpin) and target (containing a helix-turn-helix) through ESMFold, then replace the target's representations in the helical region with the donor's hairpin representations at specific blocks, aligning at loop positions. We disable recycling, which has minimal impact for the short proteins considered (App. F.1).

Dataset and feasibility. We curated 95 alpha-helical targets and ∼80,000 donor hairpins from the PDB (Berman et al., 2000). Each hairpin consists of two strands (5–10 residues each) connected by a short loop (2–5 residues). Using DSSP (Kabsch & Sander, 1983), we identified internal loop regions in each target protein, then sampled 10 donor hairpins per loop, aligning loop positions to define a target region of 15–20 residues for intervention. This yielded about 5,000 patching experiments in total, across 95 targets, 10 donors per loop, and a median of 5 loops per protein (App. E). Patching both sequence and pairwise representations across all 48 blocks induces hairpin formation in ≈40% of cases, as verified by DSSP (App. D). Trunk patching is more effective than encoder or structure module patching (App. F.2.1).

Single-block patching reveals two regimes. To localize the computation, we patch either sequence or pairwise representations at a single block k, restricting to the ≈2,000 pairs where full patching succeeded. The result (Fig. 1) reveals two distinct regimes: sequence patches are effective in early blocks ($k \in \{0, \ldots, 7\}$), peaking at ∼40% success at block 0, while pairwise patches become effective starting around block 25, reaching ∼20% by block 35.

Reverse patching (helix into hairpin targets) produces a similar two-regime structure (App. F.2.3). This symmetry suggests that the two computational stages reflect general properties of the folding trunk—early biochemical feature propagation and late geometric refinement—rather than mechanisms specific to hairpin formation. Next, we investigate these stages and their computations.

3 Early Blocks: Building Pairwise Chemistry

In Appendix F.3, we trace information flow from sequence into pairwise representations, finding that the main transfer occurs in early blocks: after sequence patching at block 0, z rapidly shifts toward the donor representation within the first ≈10 blocks, and freezing `seq2pair` during this window

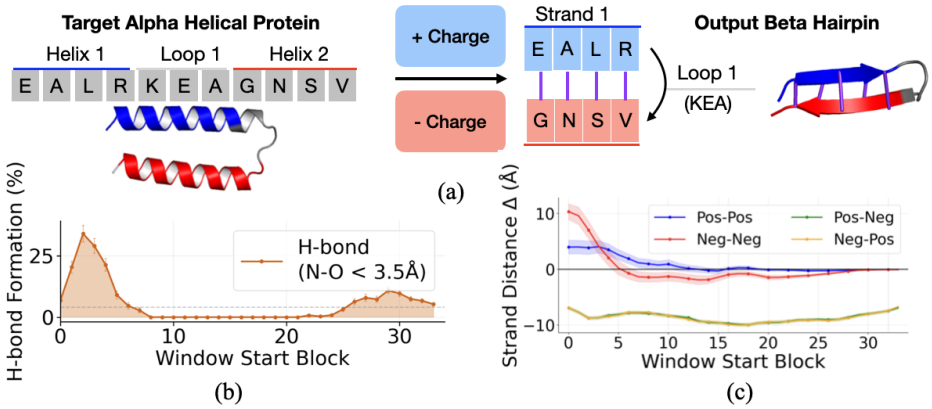


Figure 3: **Electrostatic complementarity steering induces hairpin formation.** (a) Intervention setup: we steer the sequence representation toward opposite charges on the two helical regions flanking the loop. (b) We measure hairpin induction by the percentage of cross-strand residue pairs forming backbone hydrogen bonds (N-O distance < 3.5Å). Steering with window size = 15 applied starting in early blocks (0–5) is most effective, with a smaller effect in later blocks. (c) Control: same-charge steering (Pos-Pos, Neg-Neg) on existing hairpins increases cross-strand distance, while opposite-charge steering (Pos-Neg, Neg-Pos) on helical targets decreases it, confirming the charge direction causally influences geometry.

prevents hairpin formation entirely. We now investigate whether the model leverages and propagates biochemical features from the sequence to guide folding.

We focus on charge, which is biochemically relevant for hairpin stability: antiparallel beta strands are often stabilized by salt bridges between oppositely charged residues (e.g., lysine (K) and glutamate (E)) on facing positions Ciani et al. (2003).

Charge is linearly encoded. To test whether charge is encoded, we used a difference-in-means approach to identify a "charge direction" in the sequence representation space. We collect the sequence representations of these residues 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:

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

and analogously for $\bar{s}_{\mathcal{N}}$. Projecting residue representations onto v_{charge} shows clean separation between charge classes at early blocks (App. F.5), confirming that charge is linearly encoded. Using linear probing, we find that charge and residue identity are transmitted through `seq2pair` and can be decoded from z by block 10–15 (App. F.4). However, is this information used to drive folding decisions?

Electrostatic complementarity steering. Because this direction is linear, we can manipulate it: adding v_{charge} to a representation should make it "more positive," and subtracting should make it "more negative." For a target helix-turn-helix region with residue indices $\mathcal{S}_1$ (helix 1) and $\mathcal{S}_2$ (helix 2), we steer the sequence representation as:

$$s_i' = \begin{cases} s_i + \alpha \cdot v_{\text{charge}} & \text{if } i \in \mathcal{S}_1 \\ s_i - \alpha \cdot v_{\text{charge}} & \text{if } i \in \mathcal{S}_2 \\ s_i & \text{otherwise} \end{cases} \tag{2}$$

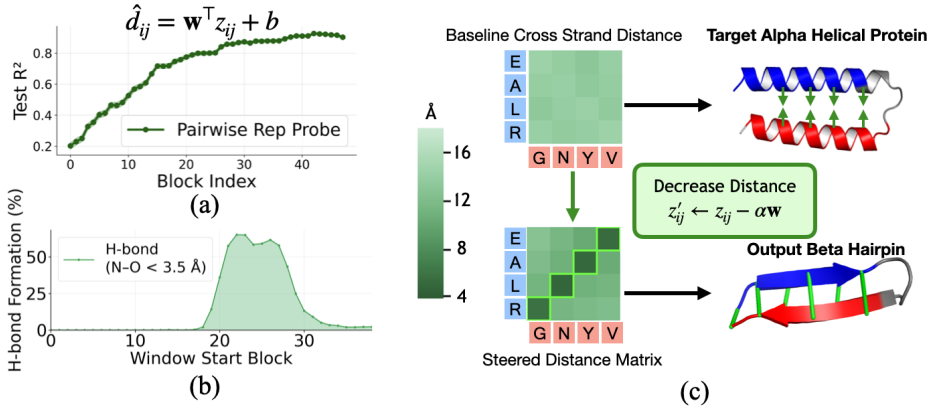


Figure 4: **Distance is linearly encoded in z and can be steered to induce contacts.** (a) Linear probe R^2 for predicting Cα distance from z, reaching ≈ 0.9 by late blocks. (b) H-bond formation peaks when steering blocks 20–35. (c) Setup: we steer cross-strand pairs toward 5.5Å by subtracting the probe weight direction ($z'_{ij} \leftarrow z_{ij} - \alpha \mathbf{w}$), decreasing cross-strand distances and inducing hydrogen bonds.

where α controls the steering strength. Our steering method is designed to mimic the pattern of cross-strand electrostatic complementarity found in natural hairpins (Fig. 3(a)).

Steering induces hairpin formation. We apply this intervention to all loop regions in our 95 target proteins, yielding approximately 500 cases. We set the steering strength $\alpha = 3\sigma$, where σ is the standard deviation of residue projections onto the charge direction. We quantify structural induction by the percentage of cross-strand residue pairs forming backbone hydrogen bonds (N–O distance < 3.5Å), a graded metric that captures partial structural changes from weaker steering perturbations. We filter out cases of nonspecific global compaction using radius of gyration (Appendix F.7).

Our complementarity steering (positive on helix 1, negative on helix 2) induces H-bond formation, with the effect concentrated with windows starting in early blocks (0–5), coinciding with stage 1 (Fig. 3b). A smaller effect persists in later blocks.

Same-charge steering confirms the mechanism. To further validate that charge causally influences folding, we perform a control experiment with same-charge steering to 500 randomly sampled beta-hairpins from our 80,000 protein dataset. Rather than inducing complementary charges, we steer both strands of existing beta-hairpins to the same charge (both positive or both negative), expecting electrostatic repulsion to increase cross-strand distance.

We observe the predicted effect: same-charge steering (Pos-Pos and Neg-Neg) increases mean cross-strand distance, particularly in early blocks where the effect peaks around +10Å before decaying toward zero (Fig. 3c). Conversely, opposite-charge steering on α-helical targets (Pos-Neg and Neg-Pos) decreases cross-strand distance, consistent with electrostatic attraction pulling the helices together.

4 Late Blocks: Pairwise Geometry Emerges

By the late blocks, sequence patching no longer induces hairpin formation, but pairwise patching does. This suggests that the pairwise representation z has taken over as the primary carrier of folding-relevant information.

4.1 Pairwise Representation Encodes Distance

What information does z carry in late blocks that enables it to control folding? A natural hypothesis is that z encodes spatial relationships between residues: particularly pairwise distances, which directly determine whether residues can form the contacts required for a hairpin. 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 (details in App. E). Even at block 0, z contains some distance information: it is initialized with relative positional embeddings that encode sequence separation, which correlates weakly with distance. But, as z is populated with sequence information and refined through triangular updates, probe accuracy increases substantially. By late blocks, R^2 reaches approx. 0.9 (Fig. 4a), indicating z contains accurate pairwise distances.

Steering z induces cross-strand hydrogen bond formation. Since the probe is linear, its gradient with respect to z_{ij} is simply the weight vector: $\nabla_{z_{ij}} \hat{d}_{ij} = \mathbf{w}$. We can therefore steer the representation toward a target distance by moving along this direction. We choose 5.5Å as our target, the typical Cα–Cα spacing for cross-strand contacts in antiparallel β-sheets Brändén & Tooze (1999). 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. We set the steering strength to $\alpha = 20\sigma$, where σ is the standard deviation of residue-pair representations projected onto this distance direction. We apply this intervention symmetrically to both z_{ij} and z_{ji} across trunk blocks with window size = 10.

We measure success by the formation of backbone hydrogen bonds (N–O distance < 3.5Å). Steering produces measurable structural changes: cross-strand distances decrease and β-sheet hydrogen bonds form in the target region (Fig. 4c). The effect depends on when steering is applied. Steering in blocks 20–35 is most effective, with H-bond formation peaking around 40% (Fig. 4b). Steering in early blocks is less effective, both because the distance probes are less accurate early in the trunk (Fig. 4a) and because subsequent computation can overwrite the perturbation.

We also find that z feeds back into the sequence representation via the `pair2seq` pathway: in late blocks, the pairwise representation produces attention biases that preferentially connect residues in structural contact, enabling structure-aware communication in the sequence update (Appendix F.8). Additionally, scaling z by a constant factor before the structure module proportionally scales output distances, while scaling s has virtually no effect on geometry (App. F.9). Thus, the direction of z encodes specific pairwise distances while its magnitude controls global scale, confirming that z acts as a geometric blueprint that the structure module renders into 3D coordinates

5 Discussion

Our analysis reveals a two-stage computational structure in ESMFold's folding trunk: early blocks propagate biochemical features (e.g., charge) from sequence into pairwise representations, while late blocks develop geometric features encoding distances and contacts. These representations are causally linked to structural outcomes: steering the charge direction induces changes consistent with electrostatic attraction and repulsion, and manipulating the distance direction enables targeted hydrogen bond formation. We note that our analysis focuses on a single motif and model; we discuss limitations in Appendix C.

Intriguingly, the two computational stages we observe in ESMFold echo proposals from physical studies of hairpin folding, where early folding is driven by interactions between sequence-distant residues and stable backbone hydrogen bonds form only later as the structure consolidates Dinner et al. (1999). Whether this parallel reflects a shared computational logic or is coincidental remains an open question, but it suggests that interpretability of folding models may yield insights that connect back to folding mechanisms themselves.

To our knowledge, our work is the first to analyze the computational mechanisms of the folding trunk of a protein structure prediction model. By pinpointing where and how folding models represent structural features, we have laid a foundation for future discovery. In other domains, interpretation of AI models has yielded insights previously unknown to humans (Davies et al., 2021; 2024; Wong et al., 2024; Schut et al., 2025; Wang et al., 2026). Protein folding models contain generalized knowledge that may be just as valuable as their ability to output structure. Interpretability offers a way to unlock this knowledge: whether by extracting learned principles, understanding the origins of model errors, or enabling us to redirect model capabilities toward tasks they were not explicitly trained for.

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 *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=zdOGBRQEbz.

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.

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.

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

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/ja0300741. URL https://doi.org/10.1021/ja0300741. PMID: 15369359.

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.

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

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.

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.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, et al. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26 (83):1–64, 2025.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. Distributional vectors encode referential attributes. In Lluís Màrquez, Chris Callison-Burch, and Jian Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 12–21, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1002. URL https://aclanthology.org/D15-1002/.

Jannik Adrian Gut and Thomas Lemmin. Dissecting alphafold2's capabilities with limited sequence information. *Bioinformatics Advances*, 5(1):vbae187, 11 2024. 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. *bioRxiv*, 2025. doi: 10.1101/2025.04.11.648460. URL https://www.biorxiv.org/content/early/2025/04/17/2025.04.11.648460.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. *arXiv preprint arXiv:1909.03368*, 2019.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Dmitry Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Zidek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stoian Nikolov, Rishub Jain, Jonathan Adler, Tim Back, Steffen Petersen, Daniel Reiman, Liz Clancy, Michal Zielinski, Martin Steinegger, Magdalena Pacholska, Tobias Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, and Andrew W Senior. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583–589, 2021. doi: 10.1038/s41586-021-03819-2.

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

Nobuyasu Koga, Rie Tatsumi-Koga, Gaohua Liu, Rong Xiao, Thomas B. Acton, Gaetano T. Montelione, and David Baker. Principles for designing ideal protein structures. *Nature*, 491:222–227, 2012. doi: 10.1038/nature11600.

Arne Köhn. What's in an embedding? analyzing word embeddings through multilingual evaluation. In Lluís Màrquez, Chris Callison-Burch, and Jian Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2067–2073, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1246. URL https://aclanthology.org/D15-1246/.

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, 2023a. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/abs/10.1126/science.ade2574.

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, 2023b. doi: 10.1126/science.ade2574.

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. *Advances in Neural Information Processing Systems*, 36, 2022. arXiv:2202.05262.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, Eric Todd, David Bau, and Yonatan Belinkov. The quest for the right mediator: Surveying mechanistic interpretability for nlp through the lens of causal mediation analysis. *Computational Linguistics*, pp. 1–48, 09 2025. ISSN 0891-2017. doi: 10.1162/COLI.a.572. URL https://doi.org/10.1162/COLI.a.572.

Victor Munoz, 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.

Taihei Murakami, Yuki Hashidate, and Yasuhiro Matsunaga. Mechanistic interpretability of fine-tuned protein language models for nanobody thermostability prediction. *bioRxiv*, pp. 2025–12, 2025.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Nithin Parsan, David J. Yang, and John J. Yang. Towards interpretable protein structure prediction with sparse autoencoders. In *ICLR 2025 Workshop on Generative and Experimental Perspectives for Biomolecular Design (GEM)*, 2025. URL https://arxiv.org/abs/2503.08764.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, David Kwabi-Addo, Dominique Beaini, Tommi Jaakkola, and Regina Barzilay. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, 2025. doi: 10.1101/2025.06.14.659707. URL https://www.biorxiv.org/content/early/2025/06/18/2025.06.14.659707.

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.

Elana Simon and James Zou. Interplm: discovering interpretable features in protein language models via sparse autoencoders. *Nature Methods*, pp. 1–11, 2025.

Nishant Subramani, Nivedita Suresh, and Matthew Peters. Extracting latent steering vectors from pretrained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 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 (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

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

Sarah Wiegreffe and Naomi Saphra. Mechanistic? In *The 7th BlackboxNLP Workshop*, 2024. URL https://openreview.net/forum?id=schAf4BPtD.

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.

A Related Work

Interpretability. Mechanistic analyses of neural networks increasingly focus on understanding representations and mechanisms that drive model behaviors (Geiger et al., 2025; Mueller et al., 2025; Wiegreffe & Saphra, 2024). Common approaches include probing classifiers (Gupta et al., 2015; Köhn, 2015; Belinkov, 2022), linear representation analysis (Park et al., 2024), causal interventions via activation patching (Vig et al., 2020; Meng et al., 2022) and steering vectors (Subramani et al., 2022; Marks & Tegmark, 2024). We generalize these methods to the ESMFold folding trunk.

AI protein folding interpretability. Jumper et al. (2021) use intermediate decoding to visualize intermediate results of the trunk, but recent research has dissected only the sequence encoders. Simon & Zou (2025); Adams et al. (2025); Garcia & Ansuini (2025); Murakami et al. (2025) have applied sparse autoencoders to protein language models, and Parsan et al. (2025) steered ESM encoder SAE features to induce structural changes. Gut & Lemmin (2024) analyzed AlphaFold2 behaviorally. Unlike previous work on encoders, we analyze the folding trunk directly through causal interventions, localizing computation across pairwise attention blocks rather than only encoder features.

Beta-hairpin folding. Beta hairpins have long served as a model system in protein folding research, where their small size made them a natural choice for early experimental, theoretical, and simulation-based studies Munoz et al. (1998); Blanco et al. (1998). Prior work explored, for example, how local sequence properties give rise to nonlocal structure via physics-based models (Dinner et al., 1999). Others suggested stages in hairpin formation, where early folding is biased by interactions

between sequence-distant residues, while stable backbone hydrogen bonds form later as the structure consolidates Dinner et al. (1999). Beyond that, Koga et al. (2012) derive a rule for inferring hairpin orientation from its vicinity. We observe analogous stages inside a neural structure prediction model.

B Background

B.1 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.

Figure 5: **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. 5) 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. 6), while *tertiary structure* describes the overall arrangement of these motifs within a single chain. We focus on a specific secondary-structure motif: the beta hairpin.

Beta-hairpins. A beta-hairpin (Fig. 6) 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.

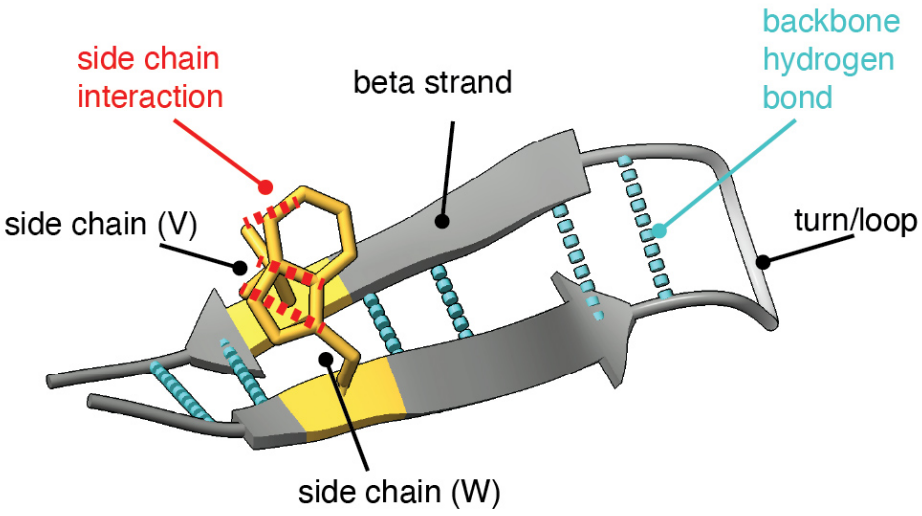
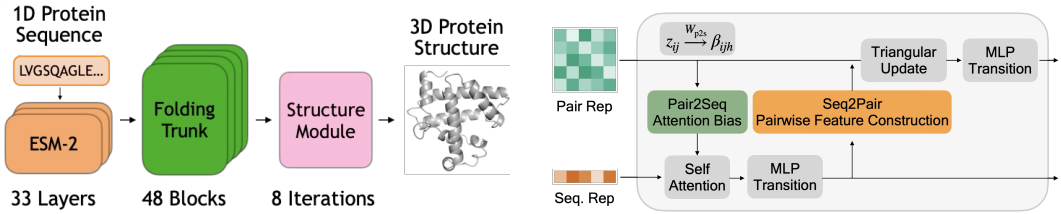


Figure 6: **3D cartoon diagram of a beta-hairpin**, a common secondary-structure motif consisting of two antiparallel β-strands (side chain V and side chain W) connected by a short turn/loop. Interactions between side chains on opposing strands and backbone hydrogen bonding together determine the folded hairpin geometry.



(a) ESMFold architecture. (b) Folding block.

Figure 7: **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. (Right) A folding block consisting of sequence and pairwise updates, including triangular updates Jumper et al. (2021). Adapted from Lin et al. (2023b).

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 Munoz et al. (1998); Blanco et al. (1998); Dinner et al. (1999); Koga et al. (2012). 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.

B.2 Protein Structure Prediction with ESMFold

We briefly outline ESMFold's architecture. For readability, we make some simplifications, e.g., we omit normalization layers and other pre-processing of the folding trunk inputs.

ESMFold. Let $x = (x_1, \ldots, x_L)$ be an amino-acid sequence of length L, with $x_i \in \{\text{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}\}$. ESMFold consists of three learned modules (see Fig. 7a) and approximately maps x to its folded protein structure.

The first module is a protein language model $f^{(1)} : (x_1, \ldots, x_L) \mapsto s^{(0)} \in \mathbb{R}^{L \times d_s}$, implemented as the BERT-style ESM-2 Lin et al. (2023b). It maps the input sequence to an initial sequence representation, which associates each residue with a vector of dimension d_s.

The second module is the folding trunk $f^{(2)} : (s^{(0)}, z^{(0)}) \mapsto (s^{(K)}, z^{(K)})$. The folding trunk starts from the initial sequence representation and an initial pair representation $z^{(0)} \in \mathbb{R}^{L \times L \times d_z}$, where d_z is the dimensionality of the pairwise representation. The pair representation $z^{(0)}$ is initialized using a learned positional embedding. The trunk then iteratively refines both representations over K blocks while preserving their shapes: $s^{(k)} \in \mathbb{R}^{L \times d_s}$ and $z^{(k)} \in \mathbb{R}^{L \times L \times d_z}$ for all $k \in \{0, \ldots, K\}$ with $K = 48$. The trunk is reapplied up to four times; each application is referred to as a recycle.

The third is a structure module $f^{(3)} : (s^{(K)}, z^{(K)}) \mapsto X \in \mathbb{R}^{L \times 14 \times 3}$ based on invariant point attention (IPA, Jumper et al. (2021)), which predicts 3D coordinates for up to 14 atoms per residue (backbone and side chain).

In this section, we treat $f^{(1)}$ and $f^{(3)}$ as black boxes and focus on the folding trunk $f^{(2)}$.

Folding trunk. The folding trunk consists of $K = 48$ blocks $g^{(1)}, \ldots, g^{(K)}$ with identical architecture (see Fig. 7, b) but separate parameters indicated by the superscript,

$$\left(s^{(k)}, z^{(k)}\right) = g^{(k)}\left(s^{(k-1)}, z^{(k-1)}\right), \qquad k = 1, \ldots, K. \tag{3}$$

Each trunk layer consists of a *sequence update* g_1 and a *pairwise update* g_2, i.e., $g(s, z) = g_2(g_1(s, z))$, with residual connections throughout. We omit the superscripts when they are clear from the context.

Sequence update. The sequence update g_1 consists of a modified self-attention layer incorporating pairwise information followed by an MLP. This is done by creating a bias term (one per attention

head) and adding it to the unnormalized attention scores. For head h, this takes the form

$$A_{ij}^{(h)} = \frac{\langle q_i^{(h)}, k_j^{(h)} \rangle}{\sqrt{d_h}} + \beta_{ijh}(z_{ij}), \tag{4}$$

in which q refers to the query, k to the key, d_h to the dimension of the head and $\beta_{ijh}(z_{ij})$ to the bias term computed from the pairwise representations z via linear projection $\beta_{ijh}(z_{ij}) = \big(W_\beta(z_{ij})\big)_h$, where $W_\beta \in \mathbb{R}^{H \times d_z}$ and H refers to the number of attention heads.

Pairwise update. The pairwise update g_2 refines the pair representation z in two steps. First, sequence-level information is incorporated through a *sequence-to-pair* operation, in which pairwise features are constructed from sequence features via elementwise multiplication and difference:

$$\phi_{ij} = \big[u_i \odot v_j \; ; \; u_i - v_j\big] \in \mathbb{R}^{2d}, \tag{5}$$

in which $[\cdot;\cdot]$ denotes vector concatenation and $u, v \in \mathbb{R}^{L \times d}$ are computed from the sequence representations via linear projection $u_i = W_u s_i$ and $v_i = W_v s_i$ for $i \in \{1, \ldots, L\}$, where $W_u, W_v \in \mathbb{R}^{d \times d_z}$. These sequence-derived pairwise features are incorporated into z via projection and addition: $z' = z + W_z \phi$, in which $W_z \in \mathbb{R}^{d_z \times 2d}$.

Second, z' is refined via triangular multiplicative updates and triangular attention Jumper et al. (2021), followed by an MLP, all with residual connections.

C Limitations

Single structural motif. Beta hairpins are among the simplest folding units, involving relatively local interactions within a 12–25 residue window. Whether the two-stage structure extends to other motifs (alpha helices, beta sheets, long-range domain contacts) remains to be tested. Helices involve primarily backbone hydrogen bonds and might depend less on pairwise representations; long-range contacts might require different mechanisms.

Single folding model. While ESMFold shares architectural features with AlphaFold2, it differs in important ways. ESMFold uses single-sequence input and an ESM-2 encoder, whereas AlphaFold2 uses MSAs and an Evoformer. The computational stages we observe may not directly transfer to other architectures.

Physical viability of counterfactual structures. Our patching experiments demonstrate that representations can be transplanted, but translating patched structures back to viable sequences remains open. The counterfactual structures we produce show low pLDDT, and inverse folding fails to recover sequences that fold to these structures. Interventions can also create unrealistic proteins with many steric clashes or chain breaks.

D 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).

E Experiment Datasets

Alpha-helical target dataset (patching targets). For activation patching experiments, we curated a separate set of 95 all-α proteins from the PDB spanning diverse structural families (globins,

helix bundles, helix–turn–helix proteins, coiled coils, EF-hand proteins, ferritins, and helical repeat proteins). As above, we verified each structure using DSSP and excluded any chain containing β-strand content. Sequence lengths were restricted to 100–400 residues. These proteins serve as targets into which hairpin representations are patched.

Block 0 and block 27 patching dataset. After running single-block patching across all donor–target pairs, we constructed two subsets of successful interventions: 400 cases of sequence patching at block 0 and 200 cases of pairwise patching at block 27. These subsets are used in the analyses of information flow and attention redirection (Section F.3, Fig. 13).

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. From the 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 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 below.

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. These constraints ensure that probing reflects general representational structure rather than sequence overlap with patching donors or targets.

F Additional Experiments

F.1 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.

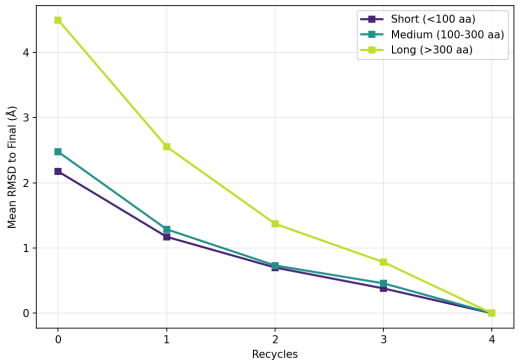


Figure 8: Mean Cα RMSD to the highest-recycle ESMFold prediction as a function of the number of recycles, shown for short ($<$ 100 aa), medium (100–300 aa), and long ($>$ 300 aa) proteins sampled from UniProt. For short sequences, the RMSD drops sharply between 0 and 1 recycle and decreases only modestly thereafter, indicating that most structural convergence occurs within the first recycle. In contrast, longer proteins continue to show substantial refinement with additional recycling.

F.2 Activation Patching Experiments

F.2.1 Patching 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. 9), 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 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 establishes the folding trunk as the critical site where structural decisions are made and motivates our detailed analysis of its computational stages in the main text.

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. 10):

- **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.

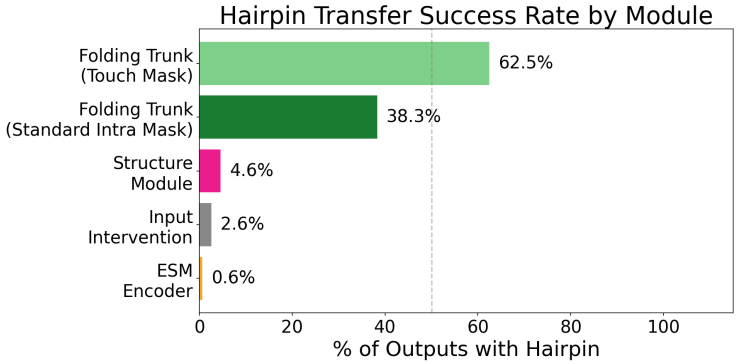


Figure 9: We apply activation patching to the ESM protein language model encoder, folding trunk, and structure module. Patching in the folding trunk is the only place where hairpins consistently transfer. Input intervention: alter the sequence residue text to match hairpin residues before inputting into ESMFold model

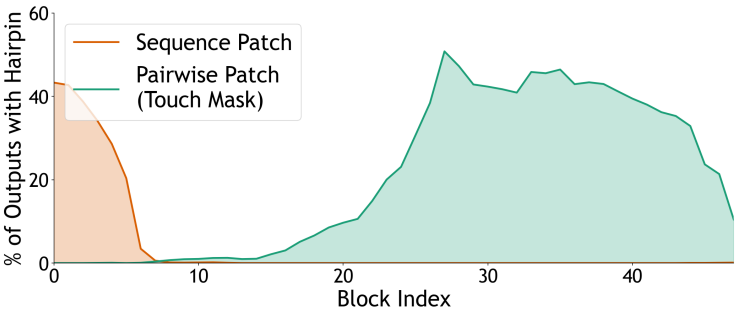


Figure 10: **Single-block patching with touch mask.** Pairwise patching using the touch mask (green) shows a broader effective window extending into late blocks (25–47), compared to sequence patching (orange) which is effective only in early blocks (0–8). The touch mask patches interactions between hairpin residues and flanking regions, bypassing the need for information to propagate from the patched region to neighboring residues.

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 "input intervention" provides an upper bound on what sequence information alone can achieve. As shown in Fig. 9, 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.

F.2.2 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. 10). 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

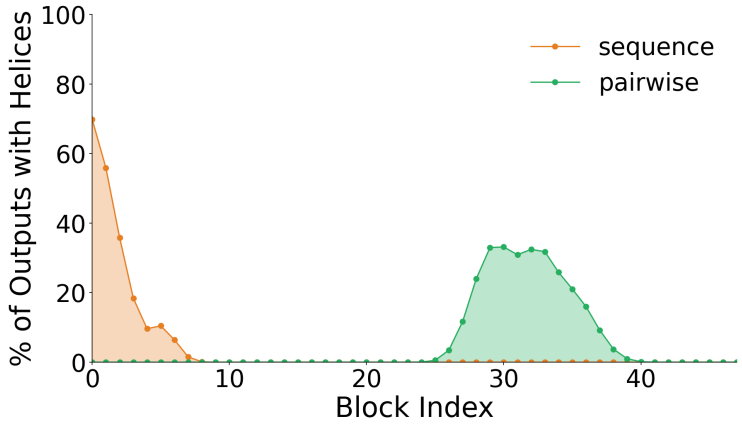


Figure 11: **Reverse patching reproduces the two-stage structure.** We patch alpha-helical representations into beta-hairpin targets and measure successful conversion ($\geq$50% alpha-helix content in the patched region). Sequence patching (orange) is effective in early blocks (0–7), while pairwise patching (green) is effective in late blocks (25–40), mirroring the pattern observed in forward (hairpin $\rightarrow$ helix) patching. This symmetry suggests the two computational stages reflect general properties of the folding trunk rather than hairpin-specific mechanisms.

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.

F.2.3 Reverse Patching: Helix to Hairpin

To test whether the two-stage computational structure is specific to hairpin formation or reflects a general property of the folding trunk, we performed the reverse experiment: patching alpha-helical representations into beta-hairpin targets to induce helix formation.

Experimental setup. We swapped the roles of donor and target proteins from our main experiments. The donor proteins now contain alpha-helical regions (the original targets), and the target proteins contain beta hairpins (the original donors). We extracted helix representations from the donor and patched them into the hairpin region of the target, measuring whether the output structure converted to alpha-helical content. We defined successful conversion as achieving $\geq$50% alpha-helix content in the patched region, as determined by DSSP secondary structure assignment.

Results. The reverse patching experiments reveal a similar two-stage structure (Fig. 11). Sequence patching is most effective in early blocks (0–7), with success rates peaking around 70% at block 0 and declining rapidly thereafter. Pairwise patching shows the complementary pattern: ineffective in early blocks but increasingly effective starting around block 25, peaking at approximately 33% success in blocks 28–32.

This symmetry suggests that the two computational stages we identified—early biochemical feature propagation and late geometric refinement—are general properties of the folding trunk rather than artifacts specific to hairpin formation. The trunk uses the same computational strategy regardless of the target secondary structure: early blocks establish sequence-derived features in the pairwise representation, while late blocks refine the geometric relationships that determine the final fold.

F.3 Sequence Information Flows Into Pairwise Space

Within the folding trunk, sequence representations s and pairwise representations z interact through two pathways (Figure 7b). The `seq2pair` pathway converts the sequence representation into a pairwise update via Eq. equation 5. The `pair2seq` pathway projects z into a scalar bias for each attention head and residue pair, which modulates sequence self-attention, via Eq. equation 4.

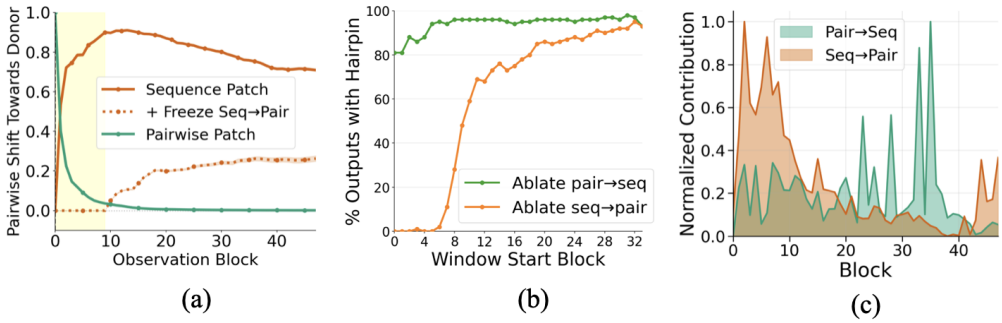


Figure 12: (a) Shift in representations in the target forward pass under different interventions (sequence patching at block 0, sequence patching at block 0 plus freezing `seq2pair` at blocks 0-10, pairwise patching at block 0). Sequence patching leads to representation alignment between intervened and donor forward pass as measured using Eq. equation 6. **Freezing `seq2pair` while doing so prevents the representation alignment.** Pairwise patching does not lead to alignment. (b) Sliding window ablation of `seq2pair` or `pair2seq` (with window size = 15) during sequence patching at block 0. As can be seen by the drop in the fraction of outputs containing hairpins in the early windows, **`seq2pair` is necessary for hairpin formation during sequence patching.** (c) The `seq2pair` pathway (orange) contributes most strongly to pairwise updates in early blocks (0–10), while the `pair2seq` pathway (green) contributes minimally in early blocks but increases substantially in late blocks (30–40).

Why does sequence patching only work in early blocks? We hypothesize that early blocks serve the critical role of transferring information from s into z via `seq2pair`, populating z before downstream computations begin. To test this, we take 400 successful cases of block-0 sequence patching and track how the patches affect the pairwise representation.

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

$$\alpha = \frac{\langle z_{\text{patched}} - z_{\text{target}}, z_{\text{donor}} - z_{\text{target}} \rangle}{\|z_{\text{donor}} - z_{\text{target}}\|^2} \tag{6}$$

Intuitively, α measures how far the patched representation has moved toward the donor relative to the original target. $\alpha = 0$ indicates the patched representation equals the target, $\alpha = 1$ indicates it equals the donor, and negative values would indicate movement away from the donor.

After patching sequence at block 0, we notice that the pairwise representation shift happens almost entirely in the first $\approx$10 blocks: z rapidly becomes donor-like early on, and then changes only gradually afterward (Fig. 12a). This suggests that the model has a short early "write-in" window during which sequence-level information is transferred into and consolidated within the pairwise representation. To test whether this early window is necessary, we blocked `seq2pair` updates during blocks 0–10. Under this intervention, z no longer makes the rapid early jump and never reaches the donor-like regime, even after the trunk resumes normal computation. In contrast, directly patching the output z at block 0 does not persist: the injected pairwise state quickly returns toward the target trajectory.

To confirm this asymmetry, we measure the relative magnitude of `seq2pair` versus triangular updates to z at each block. We quantify the relative contribution of `seq2pair` as:

$$\texttt{seq2pair contribution} = \frac{\|\phi\|}{\|\phi\| + \|\Delta z_{\text{tri}}\|} \tag{7}$$

where ϕ is the `seq2pair` update (Eq. 5) and Δz_{tri} is the combined triangular update. Similarly, the sequence representation s receives updates from both standard self-attention and the `pair2seq` bias. We measure the `pair2seq` contribution by computing the norm of the bias term relative to the total attention output. All norms are computed globally across the full representation, and we report values averaged across a test set of proteins. As expected, `seq2pair` contributions peak in early

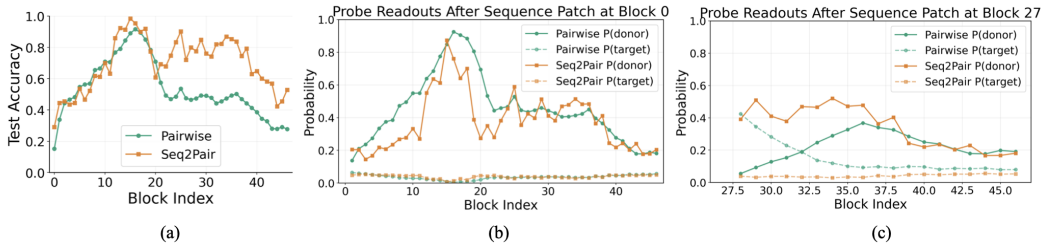


Figure 13: **Tracking residue identity transfer into the pairwise representation.** (a) We train linear probes to predict residue i's identity from the `seq2pair` output and pairwise representation z_{ij}. Probing accuracy on z (green) starts near chance and rises through blocks 0–15 as information transfers from s; ablating `seq2pair` (yellow) prevents this rise. (b) Sequence patching at block 0 (top) causes probes to read donor identities in both `seq2pair` and z, confirming the transfer pathway. Patching at block 27 (bottom) shows weaker effects: z retains more target identity because it has already been populated.

blocks and decline thereafter; `pair2seq` contributions show the opposite pattern, increasing in the second half (12c).

Pathway ablations. Finally, we confirm that `seq2pair` is necessary for hairpin formation in sequence patching. We perform sliding-window ablations, disabling `seq2pair` for consecutive blocks while patching sequence at block 0 (Fig. 12b). Ablating `seq2pair` in blocks 0-20 dramatically reduces hairpin formation; ablating it in later blocks has little effect, suggesting that sequence patches work because they propagate into z via `seq2pair` during early blocks.

F.4 Tracking Residue Identity Transfer via Seq2Pair

To characterize the timing of information flow through `seq2pair`, we tracked when residue identity information appears in the pairwise representation z. We trained linear probes to predict amino acid identity from z_{ij} and from the `seq2pair` output at each block.

Probes as tracking tools. We emphasize that these probes serve as **tracking tools** rather than evidence of learned representations. Because `seq2pair` constructs pairwise features from projected sequence representations via elementwise multiplication and difference (Eq. 5), and because the sequence representation s contains explicit residue identity embeddings, high probe accuracy is architecturally expected. A selectivity analysis following Hewitt & Liang (2019) confirms this: selectivity is near zero for all components (Table 1), indicating probes detect the embedding signature rather than learned structural features. Nevertheless, tracking **when** this signature appears in z provides useful information about computational timing.

Timing of transfer. Probing accuracy on z starts near chance at block 0 (since z is initialized with only positional embeddings) and rises through blocks 0–15, plateauing at approximately 70% accuracy (Fig. 13a). This timing aligns with our patching results: early blocks constitute a "write-in" window during which sequence information populates z.

Verification via patching. We further verified this transfer by examining probe readouts after sequence patching. When patching at block 0, probes read donor residue identities in both `seq2pair` outputs and z, confirming that the patched sequence information propagates through the expected pathway (Fig. 13b, top). When patching at block 27, the effect is weaker: z has already been populated with target identity information, making it harder to overwrite (Fig. 13b, bottom).

F.5 Charge Direction Projections

To verify that charge is linearly encoded in the sequence representation, we computed a charge direction using the difference-of-means approach (Eq. 1) and projected residue representations onto this direction across all blocks of the folding trunk.

Table 1: Probe selectivity analysis following Hewitt & Liang (2019). Selectivity (linguistic accuracy minus control accuracy) is near zero for all components, indicating probes track the residue embedding signature rather than learned structural features.

Component	Ling. Acc.	Ctrl. Acc.	Selectivity
s	0.99 ± 0.01	0.99 ± 0.01	0.00 ± 0.01
z	0.74 ± 0.21	0.73 ± 0.21	-0.02 ± 0.05
`seq2pair`	0.76 ± 0.20	0.71 ± 0.19	0.06 ± 0.04
bias	0.18 ± 0.06	0.24 ± 0.04	-0.10 ± 0.03

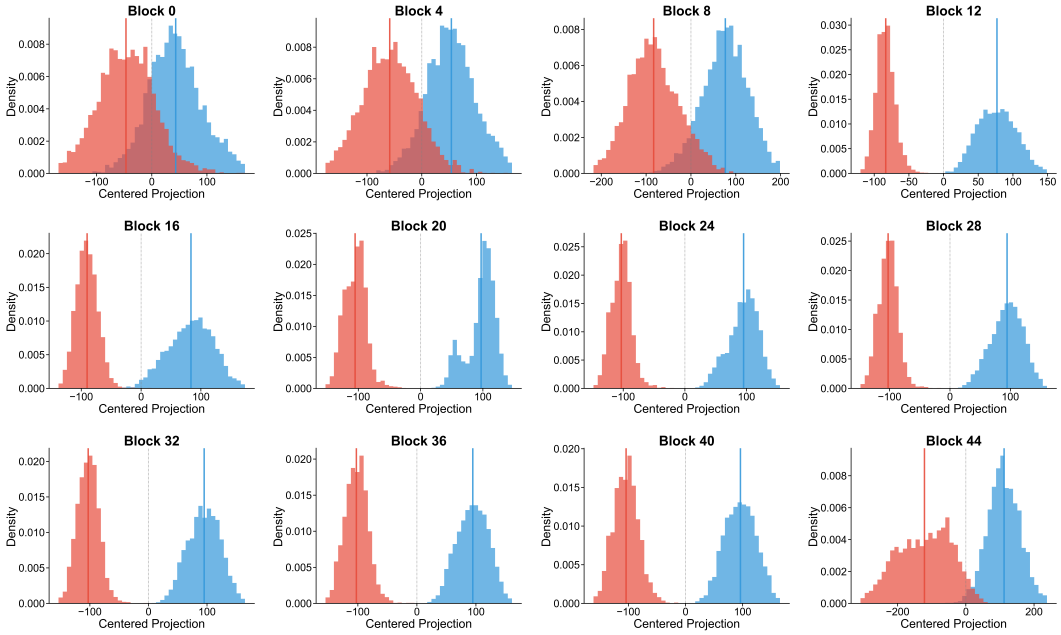


Figure 14: **Charge is linearly encoded in the sequence representation.** Projections of positively charged residues (K, R, H; blue) and negatively charged residues (D, E; red) onto the charge direction (difference of means) at each block of the folding trunk. Vertical lines indicate group means. Clean separation emerges by block 4 and is maintained throughout the trunk, confirming that charge information is encoded in a linearly accessible format.

Fig. 14 shows the distribution of projections for positively charged residues (lysine, arginine, histidine) and negatively charged residues (aspartate, glutamate) at blocks 0–44. At block 0, the distributions overlap substantially, but by block 4 clean separation emerges, with positive residues projecting to positive values and negative residues projecting to negative values. This separation is maintained throughout the remainder of the trunk.

The persistence of linear separability across all blocks confirms that charge information remains accessible in a simple linear format throughout the folding computation. This property enables our charge steering intervention (Section 3): because charge is encoded along a consistent linear direction, we can manipulate it by adding or subtracting the charge vector from sequence representations.

F.6 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.

Probe setup. For each block k, 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

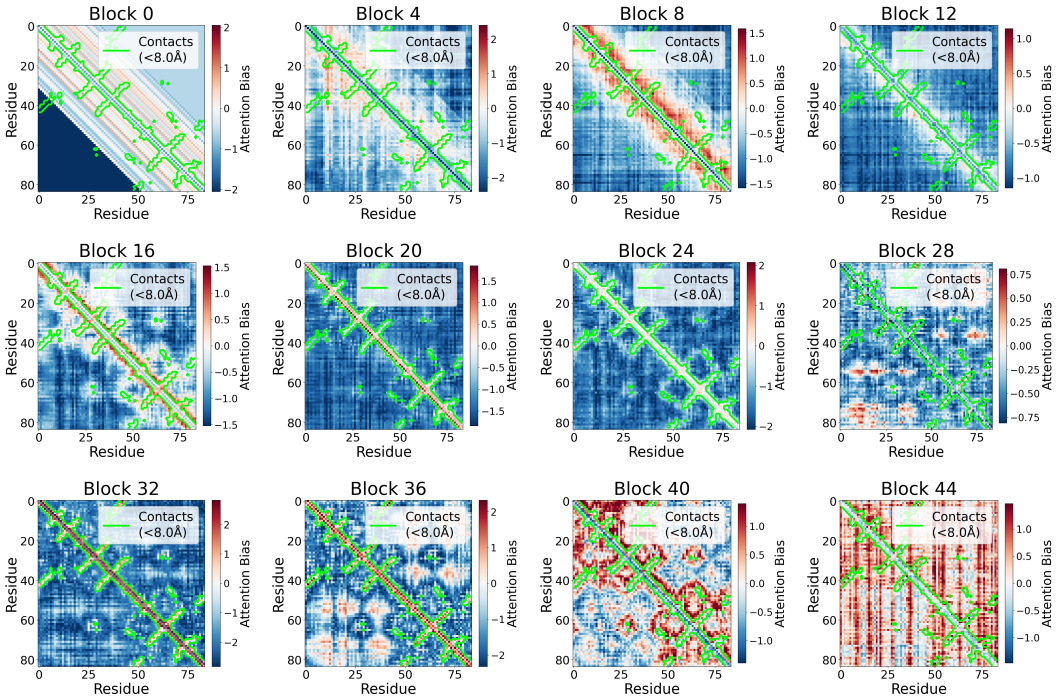


Figure 15: **`Pair2seq` attention bias across blocks (head-averaged).** Each panel shows the bias term β_{ij} from Eq. equation 4, averaged across all 8 attention heads, for protein 6rwc at selected blocks of the folding trunk. Red indicates positive bias (encouraging attention); blue indicates negative bias. Green contours mark structural contacts (Cα $<$ 8Å). In early blocks, the bias is near-uniform; by the middle blocks, it begins to align with the contact map, and by late blocks, contacting residue pairs receive substantially higher bias than non-contacts.

(D, E). Specifically, 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 the `seq2pair` output ϕ_{ij} (Eq. 5) to track when charge information first enters the pairwise pathway.

We train probes on the probing dataset E, reporting balanced accuracy to account for class imbalance (charged residues comprise approximately 15% of natural proteins).

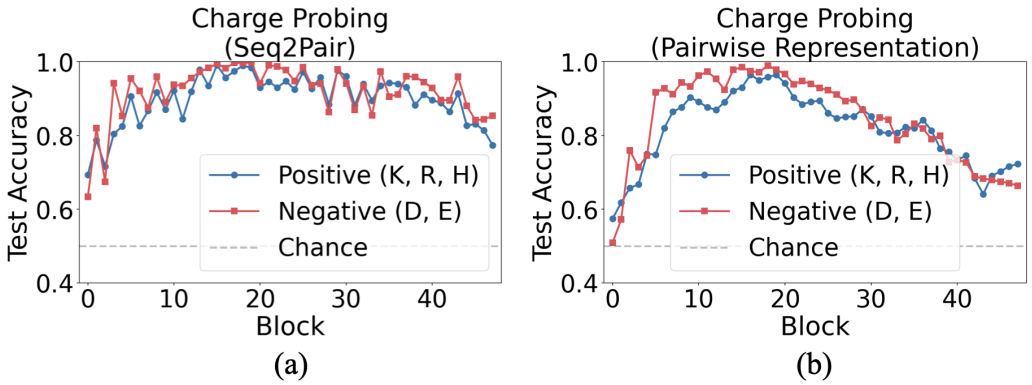


Figure 16: **Charge information is linearly accessible in pairwise representations.** Balanced accuracy of linear probes predicting positive charge (K, R, H; blue) and negative charge (D, E; red) from (a) the `seq2pair` output and (b) the pairwise representation z. Probes predict the charge of residue i from pairwise features at position (i, j) for sequence-distant pairs ($|i - j| \geq 4$). The `seq2pair` output maintains high accuracy across all blocks. In contrast, probes on z start lower at block 0 and rise through blocks 0–15, consistent with charge information transferring into z via `seq2pair` during early blocks. Both representations show declining probe accuracy in late blocks (35–47), suggesting charge information is progressively transformed into more geometric features as the trunk shifts to stage 2 computation.

Results. Figure 16 shows probe accuracy across blocks for both representation types. The `seq2pair` output maintains high probe accuracy throughout all blocks, confirming that charge information is consistently available at the interface between sequence and pairwise representations. In contrast, probes on z start near chance at block 0 (since z is initialized only with positional embeddings) and rise through blocks 0–15, plateauing at accuracies comparable to the `seq2pair` probes.

This timing aligns with our patching results (Section F.3): the early blocks constitute a "write-in" window during which biochemical features including charge are transferred from sequence representations into z via the `seq2pair` pathway. The convergence of z probe accuracy to `seq2pair` accuracy by block 15 indicates that this transfer is largely complete by the end of stage 1.

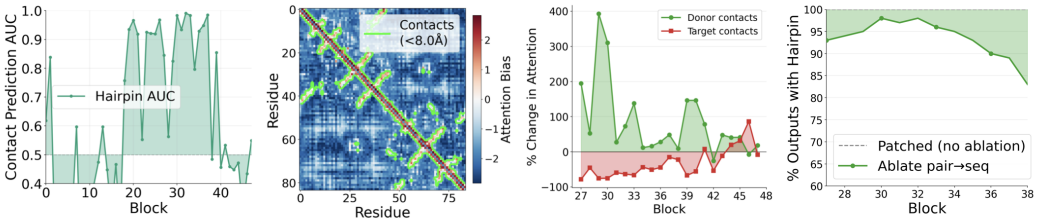
F.7 Structural Induction Metric and Compaction Filtering

Hydrogen bond metric. We quantify structural induction under steering using the percentage of cross-strand residue pairs forming backbone hydrogen bonds (N–O distance < 3.5Å). Because steering applies weaker perturbations than full activation patching, we use this graded metric rather than binary DSSP classification to capture partial structural induction. Backbone hydrogen bonding is the defining physical interaction of beta-sheets: antiparallel beta-strands are stabilized by inter-strand hydrogen bonds between backbone amide (N–H) and carbonyl (C=O) groups. Because these interactions emerge as strands begin to align—even before full secondary-structure geometry is established—hydrogen bond formation provides a sensitive and mechanistically direct measure of cross-strand pairing Munoz et al. (1998); Dinner et al. (1999).

Compaction filtering. Steering perturbations applied to intermediate blocks can cause the predicted structure to collapse globally rather than form specific cross-strand contacts. To distinguish genuine contact formation from nonspecific compaction, we compute the radius of gyration

$$R_g = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{r}_i - \bar{\mathbf{r}}\|^2} \tag{8}$$

over Cα coordinates and exclude cases where R_g falls below 0.7 times the baseline (unsteered) value. This removes cases in which the hydrogen bond metric would be inflated by global collapse rather than specific strand pairing.



(a) (b) (c) (d)

Figure 17: **The `pair2seq` bias encourages communication among structural contacts during the sequence update.** (a) ROC-AUC for classifying contacts (Cα $<$ 8Å) vs. non-contacts using bias values alone. The `pair2seq` bias cleanly separates contacts in hairpins in middle and late blocks. (b) **Pair-to-sequence bias vs. structural contacts (Block 32)**. Heatmap: attention bias values averaged across heads (red = positive, blue = negative). Green contours: residue pairs within 8Å. High bias correlates with spatial contacts. (c) For each block k after patching, we compute the percent change in mean attention probability $\bar{A}_{ij}^{(k)}$ (averaged across heads) to residue pairs (i, j) that are contacts uniquely in the donor structure (green) versus uniquely in the target structure (red). Percent change is computed relative to the unpatched baseline. After patching, attention shifts toward donor contacts and away from target contacts. (d) Ablating the `pair2seq` bias after patching results in minor drops in hairpin formation, suggesting that this bias is not fully critical.

F.8 Pair2Seq Promotes Contact Information

How does distance information in z influence the rest of the computation? One pathway is `pair2seq`: the pairwise representation is projected to produce a scalar bias that modulates sequence self-attention (Fig. 7b). We examine what this bias encodes.

We use the 300 hairpin sequences from our probing dataset and extract the `pair2seq` bias values at each block. For each sequence, we compute whether the bias cleanly separates contacting residue pairs (Cα distance $<$ 8Å) from non-contacts, measuring this with ROC-AUC: an AUC of 1.0 would indicate that all contacts receive higher bias than all non-contacts, while 0.5 indicates no separation.

In middle and late blocks, the `pair2seq` bias cleanly isolates structural contacts (Fig. 17a). Visualizing bias values alongside contact maps reveals that residue pairs in contact receive substantially more positive bias than non-contacts (Fig. 17b). We visualize the bias values across blocks in Appendix F.10, along with per-head biases which show distinct patterns and suggest head specialization as an area for future investigation.

This provides a mechanism for late-block coordination: z broadcasts contact information back into s, allowing the sequence representation to update in a structure-aware manner before the final output. Specifically, `pair2seq` bias encourages the sequence attention to preferentially mix information between contacting residues in late blocks.

Causal role of `pair2seq`. Is this relationship causal? We tested this by patching the pairwise representation at block 27 for 200 successful cases and measuring how sequence attention changes in subsequent blocks. For each block k after patching, we computed the percent change in mean attention probability $\bar{A}_{ij}^{(k)}$ (averaged across heads) relative to the unpatched baseline, separately for residue pairs that are contacts uniquely in the donor versus uniquely in the target structure. After patching, attention to donor-unique contacts increases substantially (up to 400% at block 29), while attention to target-unique contacts decreases (Fig. 17c). The pairwise representation causally redirects which residues attend to each other.

However, ablating the `pair2seq` pathway with a sliding window of size 10 after patching only reduces hairpin formation by 10–20% (Fig. 17d), suggesting that while `pair2seq` contributes to structure-aware communication, it is not the sole mechanism by which z influences the final output.

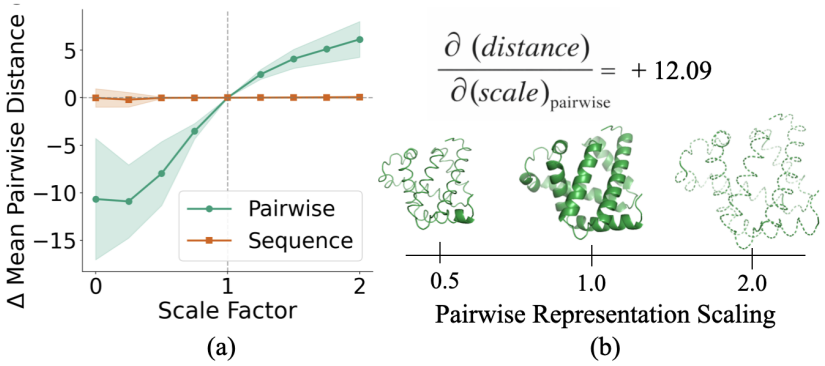


Figure 18: **Scaling the pairwise representation produces proportionally scaled 3D structures.** (a) Effect of scaling the pairwise (z) versus sequence (s) representations on mean pairwise CA distance, shown as deviation from baseline (scale=1.0). Shaded regions indicate standard deviation across 600 proteins. (b) Example structures at different pairwise scaling factors, with the gradient $\partial(\text{distance})/\partial(\text{scale})$ computed at the normal operating point.

F.9 Z Functions As A Distance Map

The `pair2seq` pathway is one route by which z influences the output. But z also feeds directly into the structure module, which produces the final 3D coordinates (Fig. 7b). How does the structure module use this information?

Scaling experiments reveal z functions as a distance map. To isolate how the structure module uses z, we scaled the pairwise representation by factors ranging from 0 to 2 before it enters the structure module, while holding s fixed and vice versa. We applied these interventions to all 600 proteins in our probing dataset and measured the mean Cα–Cα distance in the output structure.

The effect is striking: scaling z monotonically scales the mean pairwise distance between residues in the output structure. Scaling up causes the protein to expand; scaling down causes it to contract (Fig. 18a, b). In contrast, scaling s has virtually no effect on output geometry. This result complements our steering experiments (Section 4.1): there we showed that the direction of z encodes specific pairwise distances, here we show that its magnitude controls global scale. Together, these findings confirm that z acts as a geometric blueprint that the structure module renders.

F.10 Bias Maps

We visualize the `pair2seq` attention bias (Section F.8) across blocks and across individual attention heads. For a single protein (PDB: 6rwc), we extract the bias term $\beta_{ij}(z_{ij})$ from Eq. equation 4 and average over heads. Green contours indicate structural contacts (Cα distance < 8Å).

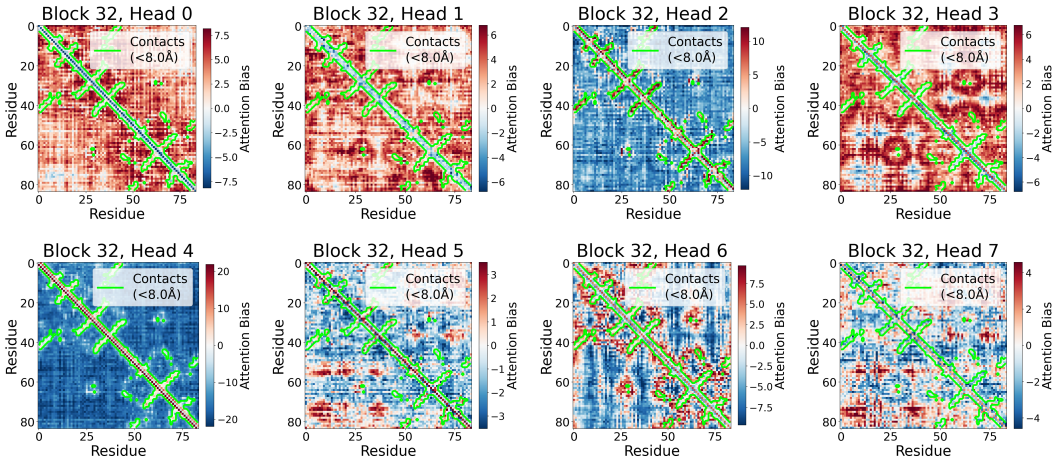


Figure 19: **Per-head `pair2seq` bias at block 32.** Individual attention head bias values for beta sheet protein 6rwc at block 32. Green contours mark structural contacts (Cα $<$ 8Å). Different heads exhibit distinct patterns: some heads show strong contact-aligned bias (e.g., heads that highlight the off-diagonal contact structure), while others capture different spatial relationships or show more diffuse patterns. This specialization suggests that individual heads attend to complementary aspects of pairwise geometry.

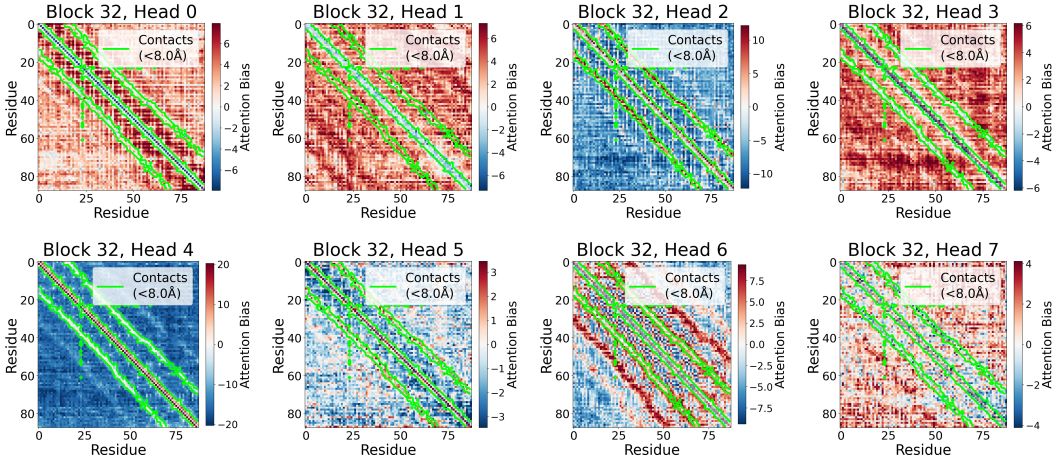


Figure 20: **Per-head `pair2seq` bias at block 32.** Individual attention head bias values for protein beta helix protein 1l0s at block 32. Green contours mark structural contacts (Cα $<$ 8Å). Different heads exhibit distinct patterns: some heads show strong contact-aligned bias (e.g., heads that highlight the off-diagonal contact structure), while others capture different spatial relationships or show more diffuse patterns. This specialization suggests that individual heads attend to complementary aspects of pairwise geometry.

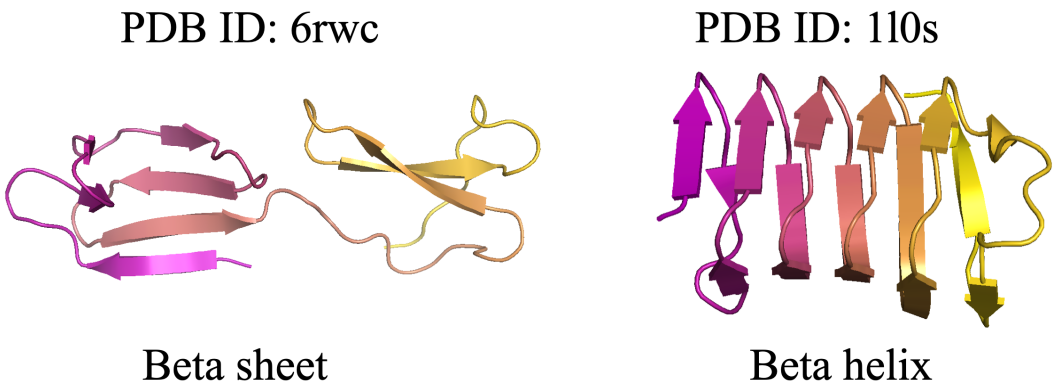


Figure 21: **Example proteins used throughout the analysis.** PyMOL visualization of PDB 6rwc, a beta-sheet protein with antiparallel strands connected by turns, and PDB: 1l0s, a left-handed beta-helix with repeating triangular coils. These two structures illustrate distinct beta-rich folds and serve as representative examples for the pair2seq bias visualizations in Appendix F.8.