CO-DIRECT
ANTICIPATION AND DIRECTIONAL ROUTING FOR JOINT PROTEIN GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Joint generative models aim to produce compatible outputs by allowing their predictions to inform one another. In protein design, this requires coordinating an amino-acid sequence with its three-dimensional structure. Yet stronger influence between these predictions does not necessarily make that influence beneficial. We investigate this distinction in a joint protein generator and find that the benefit of an additional prediction varies across residues and tasks, while the resulting response can increase reference-based prediction error. We introduce Co-DIRECT to adaptively coordinate sequence and structure through two components: anticipation and a router. Anticipation probes how the model's predictions respond to a tentative update, providing corrections that account for the coupled response. The router uses information from both sequence and structure representations to determine which representation receives these corrections at each residue. Together, these components regulate generation without querying an external structure predictor during sampling. Across binder design, enzyme scaffolding, and antibody CDR redesign, Co-DIRECT improves the reported mean assessments of refolded designs relative to frozen and fully adapted Proteina-Complexa. Component ablations and routing controls support the combined approach, with improved outcomes accompanied by fewer local error-increasing responses. These findings support adaptive coordination of sequence and structure as a strategy for improving joint protein generation.

1 INTRODUCTION

Joint generation aims to produce compatible outputs by allowing their representations to inform one another throughout generation. Joint diffusion and flow models pursue this goal through cross-modal interactions, so that each representation can guide updates to the others as they evolve (Ruan et al., 2023; Bao et al., 2023; Li et al., 2025). The same motivation is driving a shift in protein engineering. A widely used approach first generates a backbone with RFdiffusion and then assigns a sequence with ProteinMPNN (Watson et al., 2023; Dauparas et al., 2022). This ordering allows structure to guide sequence design but leaves the backbone fixed while the sequence is generated. Joint protein generators instead update backbone and sequence together, allowing each to inform the other during sampling (Campbell et al., 2024; Chu et al., 2024; Geffner et al., 2026). The intended benefit is a mutually consistent sequence and backbone that also satisfy the geometric constraints of a ligand, target protein, or antigen (Figure 1a).

Making these interactions available does not guarantee that they improve the generated pair. ProtComposer and base Proteina-Complexa report improvements when their generated sequences are redesigned (Stärk et al., 2025b; Didi et al., 2026b), although other comparisons favor co-generation under different search and evaluation settings (Didi et al., 2026a; Lin et al., 2026). These findings leave open how to make better use of coupled predictions during generation. We examine a narrower question at the prediction level. When a joint generator is evaluated again after a small change to its inputs, does the resulting correction improve its prediction? On held-out proteins, applying this correction uniformly produces local benefits and costs of comparable magnitude, including at molecular contacts, although the absolute changes are small (Figure 2). This diagnostic measures the

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

usefulness of the response without isolating cross-modal information transfer. It motivates explicitly controlling which input is changed and which channel uses the resulting response.

We introduce **Co-DIRECT**, which combines *anticipation* with *directional routing* to control this response (Figure 1b). At each residue, routing selects whether backbone or sequence leads. Anticipation modifies the leading channel's input using partner geometry or a short lookahead and queries the shared generator again. The update combines the first prediction for the leading channel with the second prediction for the responding channel. Training teaches the response to improve the update at the original state. Our contributions are **1)** this geometry-dependent anticipation method with learned directional routing, **2)** a paired diagnostic that holds the state, reference, and scoring rule fixed to distinguish local benefits, costs, and deterioration frequency, and **3)** an evaluation of task-specific quality and generated–refold consistency under common candidate and refolding budgets, supported by component ablations and inference-policy comparisons. Co-DIRECT achieves competitive mean scores against pipelines using inverse folding, exceeding evaluated RF pipelines on binder and enzyme primary endpoints.

2 Related Work

Joint protein generation. Multiflow, Protpardelle, and La-Proteina jointly generate protein representations, allowing sequence and structure to influence one another during sampling (Campbell et al., 2024; Chu et al., 2024; Geffner et al., 2026). Complexa extends joint generation with molecular conditioning and predictor-guided search (Didi et al., 2026b), while BoltzGen also supports post-generation inverse folding (Stärk et al., 2025a). Co-DIRECT builds on a pretrained joint generator and learns how to query and select its responses during generation. Its contribution concerns the use of coupled predictions within the generator. Sequence redesign and search provide complementary ways to improve designs, whose effects depend on the evaluation setting (Appendix E.1).

Controlling interactions between modalities. Cross-modal attention and joint attention enable representations to exchange information (Ruan et al., 2023; Bao et al., 2023; Esser et al., 2024; Li et al., 2025). Noise schedules and guidance further control their interaction during generation (Li et al., 2025; Qiu et al., 2025; 2026), while gradient balancing addresses unequal contributions during training (Wang et al., 2020; Peng et al., 2022). These approaches motivate explicit control over coupled generation. Co-DIRECT makes this control residue-specific by selecting which channel supplies a modified input and which channel receives the resulting prediction. Its router also weights self-conditioning, while the response objective teaches the selected prediction to update the original state.

Anticipation and additional model evaluations. AMED learns intermediate evaluation times and direction scales from teacher trajectories (Zhou et al., 2024). LOLA accounts for another learner's anticipated parameter update (Foerster et al., 2017). Co-DIRECT shares the idea of evaluating a response before committing to an update. It constructs an auxiliary input through geometry snap or latent lookahead, evaluates the same generator again, and retains the second prediction only for the selected follower. The response is trained against the original-state velocity. This differs from fitting a teacher's sampling trajectory or anticipating another learner's parameter update.

Measuring the usefulness of interactions. EMAP examines predictive cross-modal interactions, while pointwise $\mathcal{V}$-information measures instance-level predictive value (Hessel & Lee, 2020; Ethayarajh et al., 2022). Our diagnostic asks whether a specified correction improves a joint generator's prediction under a fixed scoring state and reference. Separating local benefits from costs reveals changes that aggregate error can obscure. The diagnostic measures response utility, including within-channel effects, without identifying cross-modal information transfer alone.

3 When Does an Additional Prediction Help?

Joint generation lets backbone and sequence influence both predicted updates, but these interactions need not improve predictions. We ask whether a correction from an additional model evaluation reduces error and whether its overall effect conceals local improvements and deterioration.

We evaluate the frozen Proteina-Complexa checkpoints used to initialize Co-DIRECT, retaining ordinary self-conditioning. We corrupt a reference at each evaluation time, advance both inputs

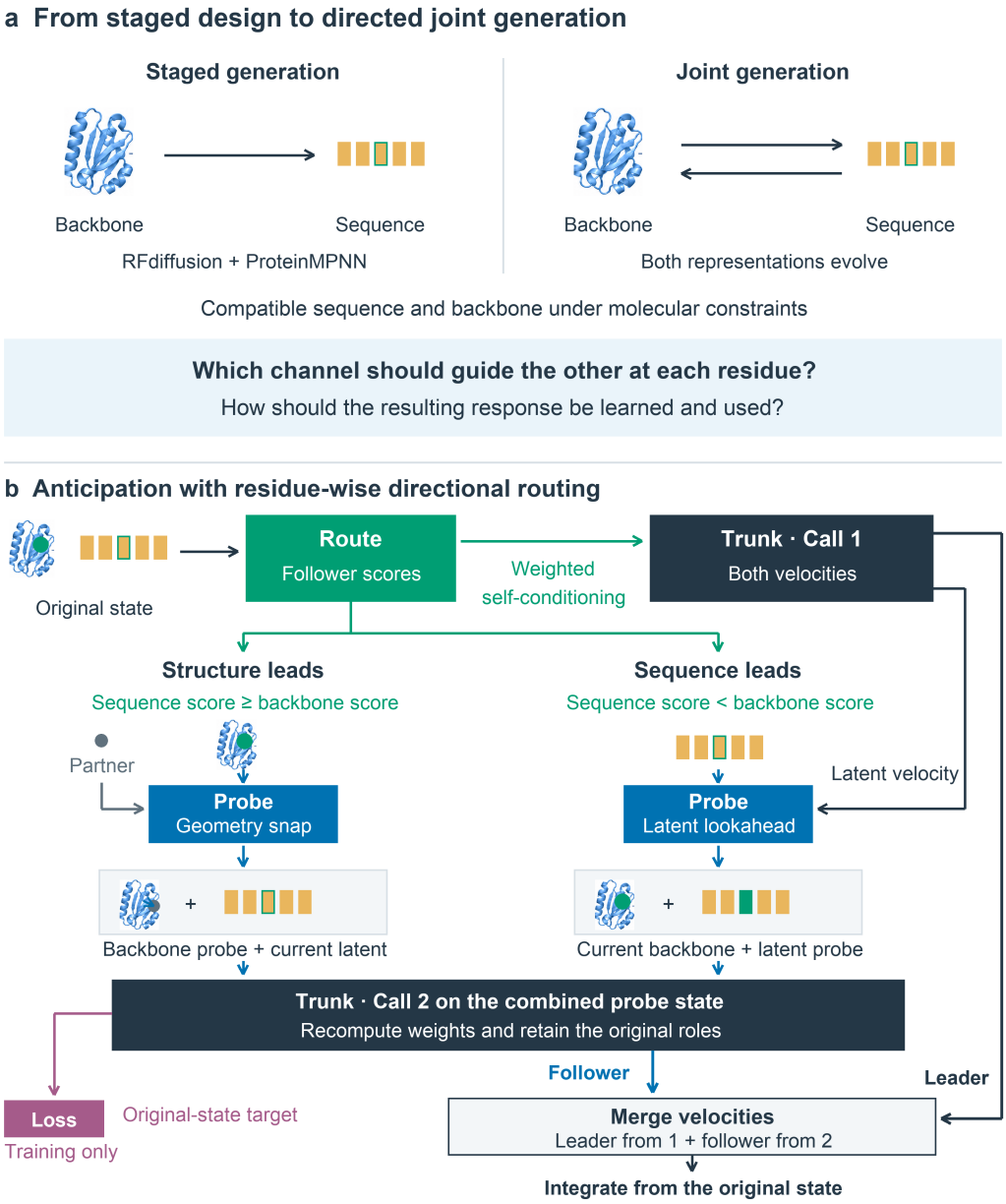


Figure 1: **From joint generation to directed responses.** **(a)** Joint generation enables two-way interactions between backbone and sequence, raising the question of which channel should guide the other and how its response should be used. **(b)** Co-DIRECT routes the response at each residue and probes the leader through geometry snap or latent lookahead. The update combines the first-call leader prediction with the second-call follower prediction and integrates from the original state. The response loss supervises the follower against the original-state target.

slightly along their predicted velocities, and evaluate again with the same times, context, and self-conditioning. Comparing original and corrected predictions against the same reference velocity measures whether the clipped response improves the original prediction (Figure 2a).

Perturbing both channels measures response utility without isolating cross-modal transfer. We separate local benefits from costs, including at molecular contacts. This diagnostic motivates response control in the initial models; ablations then test its effect after adaptation.

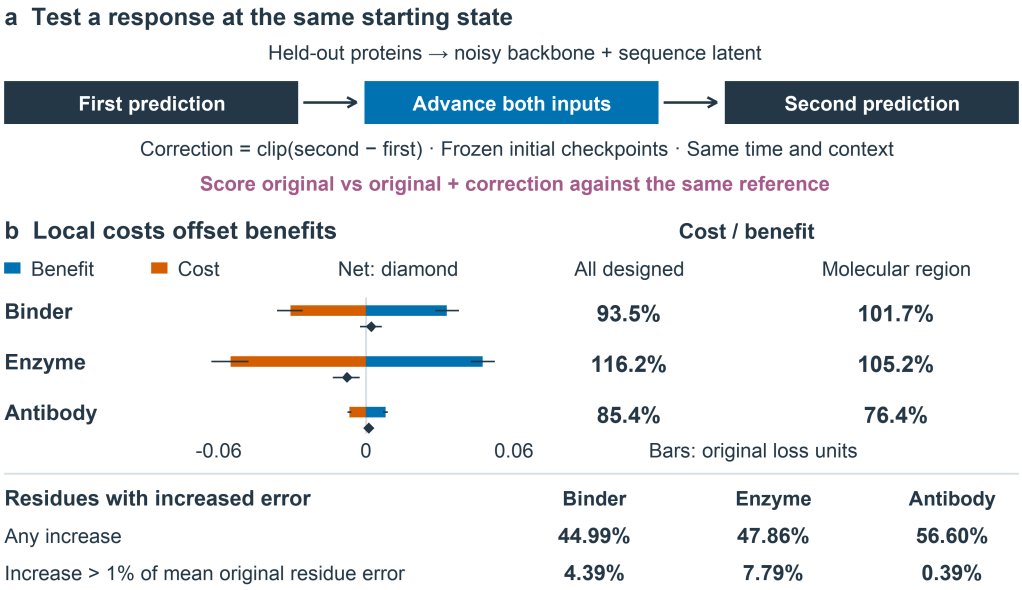


Figure 2: **Local costs offset much of an additional prediction's benefit.** **(a)** Predictions from frozen initial checkpoints share the scoring state and reference. **(b)** Bars separate benefits from costs (left); diamonds show net effects. Absolute changes remain small, including at molecular contacts. Error bars are 95% bootstrap intervals over source proteins. Bottom rows report any deterioration and deterioration exceeding 1% of the same evaluation's mean original error per designed residue. Appendix C defines regions and reports further statistics.

We use 153 binder, 216 enzyme, and 1,042 antibody reference proteins from benchmark test splits, prioritizing proteins absent from Proteina-Complexa training. Sequence identity between training, validation, and test splits is at most 30%. We evaluate eight times with four noise draws each, retaining the original loss, masks, weights, and normalization. The same correction rule is applied across eligible residues without routing or adaptation. Appendix C specifies the probes, exclusions, and averaging procedure.

Let $v = f_\theta(s, t, c)$ denote the original prediction with self-conditioning included in c, and d the clipped correction. For component m and residue i, the error reduction is $\Delta_{m,i} = \ell_{m,i}(v; s) - \ell_{m,i}(v + d; s)$. Positive values indicate improvement and negative values indicate deterioration. We separate their contributions before aggregation

$$B_s = \sum_{m,i} [\Delta_{m,i}]_+, \qquad H_s = \sum_{m,i} [-\Delta_{m,i}]_+, \qquad G_s = B_s - H_s, \tag{1}$$

where $[q]_+ = \max(q, 0)$. We average these quantities with equal weight for each source protein, and their difference exactly recovers the net error reduction. We also report the frequency and magnitude of residue-level deterioration to distinguish widespread small changes from larger local costs.

Figure 2 shows that the correction yields benefits alongside costs of comparable magnitude. Costs amount to 93.5%, 116.2%, and 85.4% of benefits for binders, enzymes, and antibodies, respectively. Absolute changes are small, with benefits and costs together accounting for 0.363%, 0.251%, and 0.113% of the original error. Although 44.99%, 47.86%, and 56.60% of designed residues show some deterioration, only 4.39%, 7.79%, and 0.39% exceed 1% of the mean original error per designed residue. Thus, most local changes are small, but their accumulated costs offset much of the benefit.

The same pattern appears in regions relevant to molecular interactions. Costs amount to 101.7% of benefits at binder interfaces, 105.2% near enzyme ligands, and 76.4% at antigen contacts within designed CDR-H3. An additional response can therefore improve predictions while introducing substantial costs relative to its benefits, including at molecular contacts. These observations moti-

vate constructing and selecting responses explicitly. Co-DIRECT implements this approach through anticipation and residue-wise directional routing.

4 Anticipation and Directional Routing with Co-DIRECT

Co-DIRECT learns how to obtain and use a response from a joint generator before updating the actual state. At each designed residue, routing assigns a leading channel and a responding channel. Anticipation modifies the leader input and queries the shared generator again. The update then combines the first-call leader prediction with the second-call follower prediction. Figure 1b illustrates routing, probe construction, the two generator calls, and response supervision.

4.1 Route selects and conditions the responding channel

Let $s_t = (x_t, z_t)$ contain Cα coordinates x_t in nanometers and eight-dimensional residue latents z_t encoding sequence and non-Cα atomistic information. The modality times are $t = (t_x, t_z)$, and molecular context c supplies the ligand or protein partner and any motif conditions. Let D_i indicate designed residues and M_i indicate valid, non-padding residues. Anticipation acts on their intersection $A_i = D_i M_i$.

The router uses the current minimum Cα–partner distance $d_i(t)$ and residue latent $z_{t,i}$. An MLP with two 64-unit SiLU hidden layers (Elfwing et al., 2018) produces a contact logit and two nonnegative follower scores

$$(q_i^{(0)}, b_{x,i}^{(0)}, b_{z,i}^{(0)}) = g_\phi([d_i(t), z_{t,i}]),$$

$$w_{m,i}^{(0)} = D_i \,\mathrm{softplus}(b_{m,i}^{(0)}), \qquad m \in \{x, z\}, \tag{2}$$

$$F_{z,i}^{(0)} = A_i \mathbf{1}[w_{z,i}^{(0)} \geq w_{x,i}^{(0)}], \qquad F_{x,i}^{(0)} = A_i - F_{z,i}^{(0)}. \tag{3}$$

Superscript (0) denotes evaluation at the original state. The channel with the larger score becomes the follower, with ties assigned to the latent channel. Thus, $F_{z,i}^{(0)} = 1$ selects a structure leader and a sequence follower, while $F_{x,i}^{(0)} = 1$ selects the reverse direction. The contact logit provides auxiliary supervision during training.

The scores also weight the generator's existing self-conditioning input (Chen et al., 2022). Let h denote the incoming, unweighted clean-state prediction or zero. The first generator call uses

$$c_{\mathrm{sc}}^{(0)} = w^{(0)} \odot h, \qquad v^{(0)} = f_\theta(s_t, t, c, c_{\mathrm{sc}}^{(0)}). \tag{4}$$

During generation, h is initially zero and subsequently contains the previous step's clean prediction. During training, a separate preparatory call supplies a detached self-prediction on half the batches, while the remaining batches use zero. Prediction losses can train the router through the continuous weighting of h. The hard role assignment is not differentiated (Appendix A.4).

4.2 Probe constructs the leader input

When structure leads, the probe places nearby partner geometry in the auxiliary backbone input. Let $p_{k^*(i)}$ be the nearest valid partner atom to $x_{t,i}$ and let $r = 0.5$ nm. The geometry snap is

$$x_{t,i}^{\mathrm{probe}} = \begin{cases} p_{k^*(i)}, & F_{z,i}^{(0)} = 1 \text{ and } \|x_{t,i} - p_{k^*(i)}\|_2 \leq r, \\ x_{t,i}, & \text{otherwise.} \end{cases} \tag{5}$$

The latent input at that residue remains unchanged. Snap uses partner geometry already supplied in c to construct an auxiliary query. The resulting coordinates need not form a physically realizable intermediate structure because integration starts from the original state.

When sequence leads, the latent input takes a short lookahead along the first-call latent velocity while the backbone input remains unchanged. We detach the first velocity, clip the predicted endpoint displacement to norm $\sqrt{3}$ in latent space, and scale that displacement by the probe increment relative to the remaining flow time. Equation 11 in Appendix A.2 gives the exact construction and its training and generation increments. Together, these residue-wise choices define a single auxiliary state $s_t^{\mathrm{probe}} = (x_t^{\mathrm{probe}}, z_t^{\mathrm{probe}})$ for the second generator call.

4.3 TRUNK PREDICTS AND SELECTS THE FOLLOWER RESPONSE

The second call uses the same generator parameters, modality times, and molecular context. The router recomputes scores from the probe's distances and latents, while retaining the original role assignments. Writing $w_\phi(s)$ for the channel scores computed at state s with the supplied partner coordinates and design mask held fixed, we obtain

$$w^{(1)} = w_\phi(s_t^{\text{probe}}), \qquad c_{\text{sc}}^{(1)} = w^{(1)} \odot h,$$

$$v^{(1)} = f_\theta(s_t^{\text{probe}}, t, c, c_{\text{sc}}^{(1)}), \tag{6}$$

$$v_{m,i}^{\text{out}} = (1 - F_{m,i}^{(0)}) v_{m,i}^{(0)} + F_{m,i}^{(0)} v_{m,i}^{(1)}. \tag{7}$$

Both calls weight the same unweighted h independently. The recomputed scores change the second call's conditioning, while $F^{(0)}$ determines which predictions are retained. At each eligible residue, the leader keeps its first-call velocity and the follower receives its second-call velocity. Different residues can select different directions within the same second call.

Generation updates the original state using $s_{t+\Delta t,m} = s_{t,m} + \Delta t_m v_m^{\text{out}}$. Outside A, probe inputs retain their original values and output velocities are taken from the first call. The sampler uses two generator calls at each of 400 steps, giving 800 calls per candidate.

4.4 LOSS TRAINS THE RESPONSE AT THE ORIGINAL STATE

For the linear flow path, let $s_{1,m}$ denote the reference endpoint for channel m. At $t_m < 1$, the target velocity is $u_m = (s_{1,m} - s_{t,m})/(1 - t_m)$ (Lipman et al., 2022). The first prediction is trained with the generator's flow-matching objective and additional contact-region supervision. Because the selected velocities update the original state, the second prediction is trained against the same original-state target on follower residues

$$\mathcal{L}_{\text{ant}} = \sum_{m \in \{x,z\}} \frac{\sum_i F_{m,i}^{(0)} \| v_{m,i}^{(1)} - u_{m,i} \|_2^2}{\max(1, \sum_i F_{m,i}^{(0)})}. \tag{8}$$

The squared norm sums errors over each channel's features. We average over its follower residues and then sum the two channel losses. A channel with no follower residues contributes zero.

Router supervision combines contact classification with regression on the difference between its follower scores. The regression target is derived from reference Cα–partner and sidechain–partner distances, favoring a latent follower when the sidechain lies closer to the partner. Reference geometry supplies these training labels, while the router inputs remain the current distance and latent. Contact-region supervision trains the first call's clean prediction at designed residues whose reference Cα lies within 0.5 nm of the partner. Appendix A.5 specifies these auxiliary objectives and their reductions.

The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{FM}}(v^{(0)}) + 2\mathcal{L}_{\text{interface}} + \mathcal{L}_{\text{ant}} + \mathcal{L}_{\text{router}}. \tag{9}$$

We adapt all trunk parameters, including base weights and inherited unmerged LoRA parameters (Hu et al., 2021), using AdamW (Loshchilov & Hutter, 2017). The autoencoder remains fixed. Adam (Kingma & Ba, 2014) trains the router through geometric supervision and weighted self-conditioning. Appendix A provides initialization, optimization, and software details. The same computation applies to ligand, target-protein, and antigen contexts.

5 RESULTS

5.1 CO-DIRECT IMPROVES BINDER AND ENZYME METRICS OVER RF PIPELINES

We first assess whether Co-DIRECT improves design outcomes under a common evaluation protocol. The benchmark contains 30 binder, 41 enzyme, and 32 antibody targets. Every method submits three candidates per target and five refolds per candidate using the same ESMFold2-Fast settings (Candido et al., 2026) and family-specific selection rule, without additional reward-guided

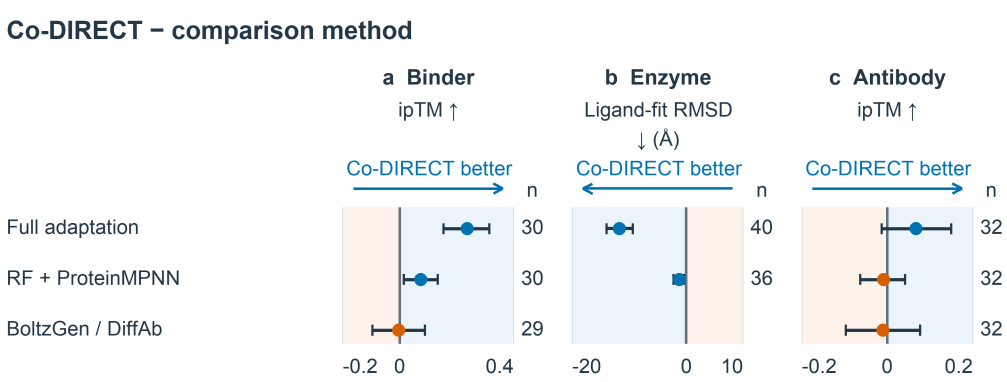


Figure 3: **Design quality against key baselines.** Target-paired means and 95% bootstrap intervals show Co-DIRECT minus comparator, with paired counts n. Positive ipTM and negative ligand-fit RMSD favor Co-DIRECT (arrows and blue shading). RF pipelines pair ProteinMPNN with RFdiffusion3, RFdiffusion2, and RFantibody for binder, enzyme, and antibody tasks. The final row uses BoltzGen and DiffAb. Appendix E.1 gives full results.

search. All endpoints use the same selected candidate and refold. Ordinary full adaptation tests improvement within the same joint generator, while RF–ProteinMPNN pipelines and BoltzGen establish performance relative to existing design approaches (Watson et al., 2023; Ahern et al., 2026; Butcher et al., 2025; Bennett et al., 2026; Dauparas et al., 2022; Stärk et al., 2025a). Appendix E.1 provides complete results, including Chroma and DiffAb (Ingraham et al., 2023; Luo et al., 2022).

Co-DIRECT raises mean binder ipTM from adaptation's 0.309 to 0.547, reduces enzyme ligand-fit RMSD from 19.89 to 8.16 Å, and raises antibody ipTM from 0.474 to 0.541 (Figure 3). Paired gains are 0.2379 ipTM ($[0.1545, 0.3154]$), 11.70 Å ($[9.36, 13.99]$), and 0.0674 ipTM ($[-0.0125, 0.1498]$). Binder and enzyme intervals favor Co-DIRECT; the antibody ipTM interval includes zero. All three primary comparisons with frozen Complexa favor Co-DIRECT. Co-DIRECT also outperforms RFdiffusion1/3 on binder ipTM and RFdiffusion2 on enzyme ligand-fit RMSD, while BoltzGen and antibody-specialist ipTM contrasts remain uncertain. Antibody ipAE improves over adaptation from 19.10 to 13.32 Å, and its secondary comparison with RFantibody also favors Co-DIRECT.

Refolding uses the designed sequence and task conditions. We assess backbone consistency by comparing generated and refolded structures. Mean Cα scRMSD falls from adaptation's 16.90 to 4.61 Å for binders and from 18.02 to 7.42 Å for enzymes. Antibody framework-aligned CDR-H3 RMSD falls from 2.12 to 1.96 Å. All three paired intervals favor Co-DIRECT. BoltzGen retains better binder consistency; Co-DIRECT improves enzyme consistency over RFdiffusion2, while antibody-specialist contrasts include zero. Appendix B.7 defines alignments, and Appendix E reports paired comparisons.

5.2 Anticipation and routing jointly improve design outcomes

To test whether extra computation explains the gains, we increase ordinary-adaptation sampling from 400 to 799 steps, nearly matching Co-DIRECT's 800 calls without routing or anticipation (Figure 4a). The extra steps improve binder ipTM by 0.0088, enzyme ligand-fit RMSD by 0.49 Å, and antibody ipTM by 0.0042. Co-DIRECT retains paired advantages of 0.2291 binder ipTM ($[0.1433, 0.3066]$), 11.20 Å enzyme RMSD ($[8.84, 13.55]$), and 0.0632 antibody ipTM ($[-0.0176, 0.1464]$). Binder and enzyme intervals favor Full; the antibody ipTM difference remains uncertain. The binder and enzyme gains therefore persist at a nearly matched generator-call budget.

Component controls test anticipation, routing, and trunk adaptation (Figure 4b). Removing anticipation retains routing but removes the auxiliary query and response objective, lowering binder ipTM by 0.0931, increasing enzyme RMSD by 3.362 Å, and lowering antibody ipTM by 0.0551. No router uses symmetric anticipation with ordinary self-conditioning and contact/response coefficients of 2/1, replacing both channels at designed residues with the second prediction. Mean losses are

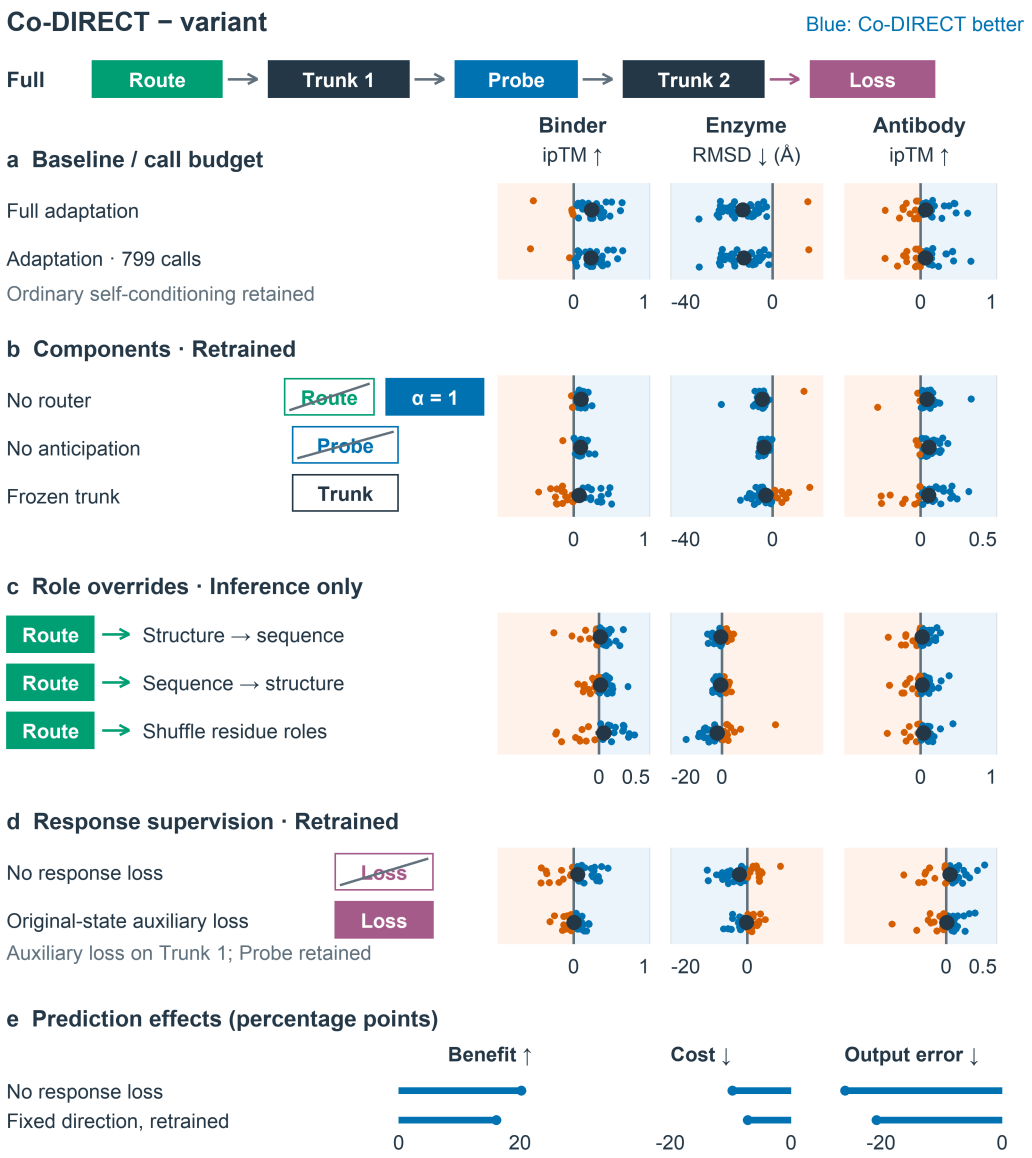


Figure 4: **Separate baseline gains from component effects.** **(a)** Ordinary adaptation and its 799-call control. **(b)** Retrained component comparisons on separate axes. Crossed modules are removed; outlined Trunk weights are frozen. No router replaces both channels with Call 2 ($\alpha = 1$), retaining contact and anticipation losses and ordinary self-conditioning. **(c)** Full-checkpoint role overrides at inference. **(d)** Retrained loss controls retain Probe at generation. **(e)** Prediction effects on 22 enzyme parents, normalized by shared reference error. Differences are Co-DIRECT minus variant; blue favors Co-DIRECT. Points are targets; dark summaries show means and nominal 95% intervals. Appendix D specifies controls and counts.

0.1001, 4.020 Å, and 0.0427. Both controls have intervals favoring Full across all three tasks, supporting their combined use. No router jointly changes role selection, the probe, and conditioning, so its effect represents their combined contribution.

Freezing the initial trunk tests the added value of generator adaptation with routed anticipation retained. It lowers binder ipTM by 0.0737, increases enzyme RMSD by 2.590 Å, and lowers antibody ipTM by 0.0527. The enzyme and antibody intervals favor Full, while the binder interval includes

zero. Routed anticipation with a frozen trunk retains much of the mean gain over ordinary adaptation; the additional benefit of trunk adaptation is task dependent.

To examine role assignments, we reuse the Full checkpoint and retain learned conditioning while changing roles only at inference (Figure 4c). Fixed leadership tests whether one direction can serve all residues, while shuffling within distance strata tests their assignment to particular residues. Shuffling incurs mean losses of 0.0483 binder ipTM, 1.732 Å enzyme RMSD, and 0.0379 antibody ipTM. All fixed-policy and shuffled-policy intervals include zero. Their mean trends favor Full, but do not resolve a primary-endpoint advantage for learned residue assignments.

Finally, we test the value and placement of response supervision (Figure 4d). Both retrained controls retain the probe during generation. Removing response loss incurs mean losses of 0.0543 binder ipTM, 1.987 Å enzyme RMSD, and 0.0419 antibody ipTM. Only the enzyme primary-endpoint interval excludes zero. Applying the auxiliary objective to the first prediction leaves smaller gaps of 0.0055, 0.148 Å, and 0.0092, with intervals spanning zero throughout. Primary endpoints support response supervision most clearly for enzymes but do not distinguish its placement from the original-state auxiliary objective. Appendix D specifies the controls.

5.3 Response supervision increases local benefits and reduces costs

We examine local benefits, costs, and final error near the ligand on corrupted complexes from 22 enzyme source proteins, using a common reference-error denominator (Figure 4e). No response loss tests explicit response supervision. The retrained fixed-direction control tests adaptation to a fixed policy, complementing inference-only overrides. Its ordinary self-conditioning means this comparison includes both role selection and learned conditioning.

Relative to the no-response-loss control, Full increases benefits from 16.35% to 36.63% of the shared reference error and reduces costs from 17.66% to 7.96%. Final error falls from 94.99% to 69.05%, despite Full having a higher first-call error (97.72% versus 93.68%). The fraction of residues with increased error also falls from 32.46% to 19.54%. Response supervision therefore improves the deployed prediction through both larger benefits and smaller local costs.

The retrained fixed-direction control yields benefits of 20.50%, costs of 15.10%, and final error of 89.77%. Full improves all three quantities, showing that its advantage persists against a model trained with a fixed direction. These results examine the learned response at reference states and complement the design outcomes above. Appendix E reports the full diagnostic results.

6 Discussion and Conclusion

Co-DIRECT makes joint generation competitive in mean task scores with pipelines using inverse folding. It improves binder and enzyme primary endpoints over adaptation and the evaluated RF pipelines, and generated–refold agreement over adaptation across all three families. Binder ipTM approaches BoltzGen's mean, but BoltzGen retains better backbone consistency. Antibody ipTM differences from adaptation remain uncertain, while secondary ipAE improves. The binder and enzyme gains persist with nearly matched generator-call budgets. Removing anticipation or using a symmetric extra-pass reduces primary performance across families. Trunk-adaptation effects vary by task, while primary inference-policy comparisons remain unresolved.

The broader contribution is to make interactions within a joint generator measurable, trainable, and controllable. Separating local benefits from costs reveals effects hidden by aggregate error, including near functional regions. Co-DIRECT specifies the auxiliary state presented to the generator and how its response enters the update. This improves joint sequence–backbone generation for protein engineering and motivates studying the direction and consequences of interactions alongside the architectures that enable them.

The evidence covers one generator family and computational evaluations. Target bootstrap intervals omit training-seed variability, and binding and catalytic activity require experimental validation. Co-DIRECT adds a model call and uses different channel probes. Future work could improve efficiency, test stability across seeds and architectures, and extend anticipation and routing to other joint generators with complementary constraints, using domain-appropriate probes and outcome assessments.

AI USE STATEMENT

We used generative AI tools to assist with drafting the implementation code of the proposed method and auxiliary scripts (e.g., data processing and plotting), and to improve the readability and grammar of the manuscript. The research ideas, method design, experimental design, and interpretation of results are entirely the authors' own; we did not use generative AI tools for ideation, hypothesis generation, synthetic data generation, or formulating or proving mathematical claims. All AI-drafted code was reviewed, modified, and tested for correctness by the authors, and all reported results were produced and verified by the authors. We take full responsibility for the final content of this work, including all text, claims, and code.

ETHICS STATEMENT

This work studies computational protein design. The reported refolding confidence, geometric agreement, and gate outcomes do not establish experimental binding, catalytic activity, therapeutic efficacy, or biological safety. Protein design methods can support beneficial engineering but may also be misused. Any experimental development requires assessment of the specific target and intended application. The present evidence supports computational benchmark claims. Experimental validation is required before application.

REPRODUCIBILITY STATEMENT

Section 4 and Appendix A describe the model and training procedure. Appendix B specifies the data splits and evaluation protocol, and Appendix D defines the model variants. Code and evaluation scripts are available anonymously at https://anonymous.4open.science/r/Co-DIRECT-0641/README.md.

REFERENCES

Woody Ahern, Jason Yim, Doug Tischer, Saman Salike, Seth M Woodbury, Donghyo Kim, Indrek Kalvet, Yakov Kipnis, Brian Coventry, Han Raut Altae-Tran, et al. Atom-level enzyme active site scaffolding using rfdiffusion2. *Nature Methods*, 23(1):96–105, 2026.

Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning*, pp. 1692–1717. PMLR, 2023.

Nathaniel R Bennett, Joseph L Watson, Robert J Ragotte, Andrew J Borst, DéJenaé L See, Connor Weidle, Riti Biswas, Yutong Yu, Ellen L Shrock, Russell Ault, et al. Atomically accurate de novo design of antibodies with rfdiffusion. *Nature*, 649(8095):183–193, 2026.

Jasper Butcher, Rohith Krishna, Raktim Mitra, Rafael I Brent, Yanjing Li, Nathaniel Corley, Paul T Kim, Jonathan Funk, Simon Mathis, Saman Salike, et al. De novo design of all-atom biomolecular interactions with rfdiffusion3. *bioRxiv*, 2025.

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

Salvatore Candido, Thomas Hayes, Alexander Derry, Roshan Rao, Zeming Lin, Robert Verkuil, Bryan Z Wu, Jin Sub Lee, Elise S Bruguera, Jehan A Keval, et al. Language modeling materializes a world model of protein biology. *bioRxiv*, pp. 2026–06, 2026.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022.

Alexander E Chu, Jinho Kim, Lucy Cheng, Gina El Nesr, Minkai Xu, Richard W Shuai, and Po-Ssu Huang. An all-atom protein generative model. *Proceedings of the National Academy of Sciences*, 121(27):e2311500121, 2024.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

Kieran Didi, Danny Reidenbach, Matthew Penner, Supriya Ravichandran, Marshall Case, Mike Nichols, Erik Swanson, Alex Reis, Maggie Prescott, Yue Qian, et al. Latent generative search unlocks de novo design of untapped biomolecular interactions at scale. *bioRxiv*, pp. 2026–09, 2026a.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael Bronstein, et al. Scaling atomistic protein binder design with generative pretraining and test-time compute. In *International Conference on Learning Representations*, volume 2026, pp. 61056–61109, 2026b.

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pp. 569–593. Springer, 1992.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *International Conference on Machine Learning*, pp. 5988–6008. PMLR, 2022.

William Falcon, Jirka Borovec, Adrian Wälchli, Nic Eggert, Justus Schock, Jeremy Jordan, Nicki Skafte, Vadim Bereznyuk, Ethan Harris, Tullie Murrell, et al. Pytorchlightning/pytorch-lightning: 0.7. 6 release. *Zenodo*, 2020.

Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

Tomas Geffner, Kieran Didi, Zhonglin Cao, Danny Reidenbach, Zuobai Zhang, Christian Dallago, Emine Kucukbenli, Karsten Kreis, and Arash Vahdat. La-proteina: Atomistic protein generation via partially latent flow matching. In *International Conference on Learning Representations*, volume 2026, pp. 67350–67395, 2026.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *nature*, 585(7825):357–362, 2020.

Jack Hessel and Lillian Lee. Does my multimodal model learn cross-modal interactions? it's harder to tell than you might think! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 861–877, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, 623(7989):1070–1078, 2023.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 32(5):922–923, 1976.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Zichun Liao, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Omniflow: Any-to-any generation with multi-modal rectified flows. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13178–13188. IEEE, 2025.

Yeqing Lin, Minji Lee, Aakarsh Vermani, Ellena Jiang, Sebastiaan De Cooman, Matej Špetko, and Mohammed AlQuraishi. Fast and ultra-capable protein design: Advancing the frontier through atomistic se (3)-equivariance with genie 3. *bioRxiv*, 2026.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems*, 35:9754–9767, 2022.

Martin Pacesa, Lennart Nickel, Christian Schellhaas, Joseph Schmidt, Ekaterina Pyatova, Lucas Kissling, Patrick Barendse, Jagrity Choudhury, Srajan Kapoor, Ana Alcaraz-Serna, et al. One-shot design of functional protein binders with bindcraft. *Nature*, 646(8084):483–492, 2025.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. Balanced multimodal learning via on-the-fly gradient modulation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8237. IEEE, 2022.

Keyue Qiu, Yuxuan Song, Zhehuan Fan, Peidong Liu, Zhe Zhang, Mingyue Zheng, Hao Zhou, and Wei-Ying Ma. Piloting structure-based drug design via modality-specific optimal schedule. *arXiv preprint arXiv:2505.07286*, 2025.

Keyue Qiu, Xintong Wang, Zhilong Zhang, Hao Zhou, and Wei-Ying Ma. Demystifying multimodal biomolecular co-design with intrinsic geodesic coupling. *arXiv preprint arXiv:2606.01628*, 2026.

Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10219–10228. IEEE, 2023.

Hannes Stärk, Felix Faltings, MinGyu Choi, Yuxin Xie, Eunsu Hur, Timothy O'Donnell, Anton Bushuiev, Talip Uçar, Saro Passaro, Weian Mao, et al. Boltzgen: Toward universal binder design. *BioRxiv*, 2025a.

Hannes Stärk, Bowen Jing, Tomas Geffner, Jason Yim, Tommi Jaakkola, Arash Vahdat, and Karsten Kreis. Protcomposer: Compositional protein structure generation with 3d ellipsoids. In *International Conference on Learning Representations*, volume 2025, pp. 11061–11089, 2025b.

Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12692–12702. IEEE, 2020.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Zhenyu Zhou, Defang Chen, Can Wang, and Chun Chen. Fast ode-based sampling for diffusion models in around 5 steps. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7777–7786. IEEE, 2024.

A DETAILED CO-DIRECT METHOD

This section specifies the computation shared by the three task families, including the two generator calls, self-conditioning, training objectives, and optimization settings.

A.1 INPUTS AND COMPUTATION AT EACH STEP

The generator state contains Cα coordinates x_t in nanometers and eight-dimensional residue latents z_t, which encode sequence and non-Cα atomistic information. The fixed autoencoder decodes the latents after sampling. Molecular context comprises ligand atoms for enzymes, target-protein atoms for binders, and antigen atoms for antibodies, together with the supplied motif or framework conditions. The router receives only the current minimum Cα–partner distance and the eight latent features at each residue. Molecular context enters the generator separately.

Table 1: Inputs and their uses. Both generator calls share context and unweighted self-conditioning. Their router scores can differ. Reference endpoints supply training labels, not forward inputs.

Quantity	Source	Destination
x_t, z_t, t	Current state and modality times	Generator; z_t also to router
d_i	Minimum live Cα–partner distance	Router, recomputed at the probe
Context c	Supplied motif and ligand/target features	Both generator calls
h	Clean self-prediction or zero	Generator after weighting by $w^{(j)}$
D, M	Existing design and occupancy masks	$A = D \wedge M$ for anticipation
s_1, reference atoms	Training complex	Velocity and geometric targets

Each sampling step uses the following two-call computation.

1. The inputs are s_t, t, c, h and the residue masks. The self-conditioning input is the previous clean prediction when available and zero otherwise.
2. The router g_ϕ evaluates s_t to produce scores $w^{(0)}$ and follower assignments $F^{(0)}$ within A. The generator then predicts $v^{(0)}$ using self-conditioning $w^{(0)} \odot h$.
3. The assignments $F^{(0)}$ determine the probe. Backbone leaders near valid partner atoms use snap, while latent leaders use the clipped displacement in Equation 11. Each follower's own local state remains unchanged, as do both channels outside A.
4. The router computes new scores $w^{(1)}$ from the probe state. The second generator call uses the original times and context with self-conditioning $w^{(1)} \odot h$. Follower assignments remain fixed. Both calls weight the same unweighted h independently, so weights are not applied twice.
5. The second velocity replaces the first only at positions selected by $F^{(0)}$. Training compares these responses with the original-state velocity target. Generation integrates the selected velocities from the original state.

Training and generation use the same probe construction. Training uses $\delta = (1 - \bar{t})/400$, where $\bar{t}$ is the training time, whereas generation uses the actual mean increment of the two modality schedules. At the first generation step the latent displacement is zero, but the geometry snap and second generator call still run. The latent displacement is clipped before it is added to the current state, as specified below.

Each candidate uses 400 sampling steps with one original-state call and one probe call per step, for 800 generator evaluations. Snap requires the task-provided partner coordinates. Nearest-atom queries exclude absent atoms, and a backbone leader is snapped only when a valid partner atom lies within 0.5 nm. The probe supplies an auxiliary input and does not replace the state from which integration proceeds.

A.2 LATENT-LEADER PROBE AND STEP SIZE

When the latent channel leads, its input takes a short prediction-based step. We clip the predicted *displacement* separately for each residue's eight latent features

$$d_{z,i} = (1 - t_z)\,\mathrm{sg}(v^{(0)}_{z,i}), \qquad \bar{d}_{z,i} = d_{z,i}\min\{1, R/\|d_{z,i}\|_2\}, \tag{10}$$

$$z^{\mathrm{probe}}_{t,i} = z_{t,i} + F^{(0)}_{x,i}\frac{\delta}{1 - t_z}\bar{d}_{z,i}, \qquad R = \sqrt{3}. \tag{11}$$

Zero displacements remain zero. Here sg stops gradients through the first velocity, and R is a latent-space norm threshold, not a length in nanometers. Training and generation use the same probe function with different increments δ (Appendix A.1). Snap itself does not depend on δ.

A.3 INTERVENTION MASKS AND MOTIF HANDLING

The mask $A = D \wedge M$ intersects the designed-residue mask D with the valid-residue mask M. It determines where probes are applied, which residues enter the response loss, and which velocities are replaced. Outside A, both output velocities come from the first call and the residue contributes no response loss. Other residues can still influence these predictions through the joint generator.

The method retains each task's design masks. In enzyme training, motif conditions are supplied as separate tokens, and valid chain residues can remain designed. During enzyme generation, specified catalytic positions are excluded from anticipation. Thus the masking rule is shared across training and generation, while the task-specific sets can differ.

Exclusion from anticipation does not freeze coordinates. The integrator continues to update valid residues outside A using their first-call velocities. In enzyme generation, export restores the specified catalytic amino-acid identities using the motif-position mapping while retaining the generated Cα coordinates of the designed chain. This sequence operation does not impose catalytic geometry. In antibody generation, export restores the supplied framework sequence and atoms while retaining the sampled Cα coordinates of the designed loops. These steps preserve the task's sequence and framework conditions before refolding.

A.4 SELF-CONDITIONING IN TRAINING AND GENERATION

Half of the training batches obtain h from an additional generator prediction on the same noisy state, converted to a clean-state prediction and detached. The other half use zero. This preparation call is additional to the two calls defining anticipation. It supplies a model prediction rather than a reference endpoint (Chen et al., 2022).

During generation, the first step uses $h = 0$, and later steps use the previous clean prediction. Both generator calls weight this same unweighted prediction independently with their current router scores. The second call recomputes the scores from the probe state while retaining the follower assignments from the first call.

Self-conditioning when the router is removed. The Full model passes $w^{(j)} \odot h$ to call j. The no-router control removes the learned weights and supplies the generator's ordinary clean-prediction or zero input h. Full adaptation also retains ordinary self-conditioning. The no-router control additionally removes residue-wise roles and replaces the directional probe with a symmetric scalar extra-pass. Inference-only fixed and shuffled policies instead retain the learned conditioning weights and change the roles used by the directional probe. These controls therefore test different interventions.

A.5 LOSS DEFINITIONS AND GEOMETRIC LABELS

Router supervision uses reference geometry only as training labels. Let $y_i = \mathbf{1}[d^{\mathrm{ref}}_{\mathrm{CA},i} \le r]$ and let S_i indicate that a sidechain atom is present. With $\langle\cdot\rangle_B$ denoting a mean over mask B,

$$\begin{aligned}\gamma_i &= 4\,\mathrm{clip}(d^{\mathrm{ref}}_{\mathrm{CA},i} - d^{\mathrm{ref}}_{\mathrm{sidechain},i}, -1, 1),\\ \mathcal{L}_{\mathrm{router}} &= \langle \mathrm{BCEWithLogits}(q^{(0)}, y)\rangle_A\\ &\quad + \langle (w^{(0)}_z - w^{(0)}_x - \gamma)^2\rangle_{A\cap S}.\end{aligned} \tag{12}$$

Distances are expressed in nanometers. A closer sidechain favors a latent follower. This is a geometric heuristic, not an error-minimizing direction label. Missing sidechains are excluded from direction regression but not contact classification.

For a row mask B, define $\langle a \rangle_B = \sum_i B_i a_i / \max(1, \sum_i B_i)$, so an empty set contributes zero. Each microbatch contains one training complex. Flow matching supervises the first prediction. The additional interface objective uses its reconstructed clean prediction $\widehat{s}^{(0)}_{1,m}$

$$\mathcal{L}_{\text{interface}} = \frac{1}{2} \sum_{m \in \{x,z\}} \left\langle \frac{\|\widehat{s}^{(0)}_{1,m} - s_{1,m}\|_2^2}{(1-t_m)^2 + \varepsilon} \right\rangle_{A \cap C}, \qquad C_i = \mathbf{1}[d^{\text{ref}}_{\text{CA},i} \le 0.5\,\text{nm}]. \tag{13}$$

Here ε stabilizes the denominator. The squared norm sums feature errors. Interface loss averages contact residues and then the two modalities, with coefficient 2 in the total objective. Anticipation instead adds the two follower-specific residue means with coefficient 1 and no division by feature dimension. The router classification and direction terms each have coefficient 1.

Contact and direction labels use reference geometry during training, whereas the router input uses the current noisy geometry. Sidechain distance is the minimum over present atoms excluding $\{\text{N}, \text{C}\alpha, \text{C}, \text{O}\}$, using the inherited atom37 representation (Jumper et al., 2021). Residues without a present sidechain atom are excluded from direction regression but remain eligible for contact classification. Both losses are averaged only over their valid designed residues.

A.6 Why the auxiliary query is an anticipation

Anticipation means querying a channel's response to a constructed partner input before updating the actual state. The latent-led probe takes a short step along the prediction. The backbone-led probe queries nearby contact geometry, not a Taylor approximation to the backbone trajectory. The supplied partner geometry is already in the context, so snap changes its use rather than adding an observation. Neither branch requires a physically realizable intermediate structure.

For an Euler update $s_{t+\Delta t,m} = s_{t,m} + \Delta t_m v^{\text{out}}_m$, the state being updated is s_t. This explains the response target $(s_{1,m} - s_{t,m})/(1 - t_m)$. Substituting the probe state into that target would define a different prediction task. Both state blocks can change across the collection of probed residues, and the conditioning scores are recomputed, so the response does not isolate a single cross-modal derivative. No equilibrium or optimal-leadership assumption is required.

A.7 Gradient paths and the two roles of the router

For fixed follower masks, probe inputs, and incoming h, the direct conditioning contribution of a prediction loss to the router gradient is

$$\left.\frac{\partial \mathcal{L}_{\text{pred}}}{\partial \phi}\right|_{h, F^{(0)}, s_t, s_t^{\text{probe}}} = \sum_{j \in \{0,1\}} \sum_{m,i} \left\langle \frac{\partial \mathcal{L}_{\text{pred}}}{\partial c^{(j)}_{\text{sc},m,i}}, h_{m,i} \right\rangle \frac{\partial w^{(j)}_{m,i}}{\partial \phi}. \tag{14}$$

Each loss uses the appropriate prediction. Flow matching and interface supervision use the first prediction, whereas anticipation uses the second. This expression describes the continuous conditioning path, not differentiation through a sampling history. It is zero when $h = 0$. The geometric router losses still provide supervision. The Boolean role comparison has no straight-through estimator. Detaching the first velocity when making the latent displacement prevents the response loss from returning through that probe-construction path. It does not detach either call's weighting of h. Generation runs without gradient tracking.

The score difference determines the role, while the individual score magnitudes scale self-conditioning. A common offset to already computed scores would preserve the role but change conditioning. Geometric difference regression therefore does not calibrate the common score magnitude by itself. Likewise, recomputing $w^{(1)}$ changes conditioning without selecting a second role. Ablations of direction and conditioning must be interpreted separately.

A.8 TRAINING AND GENERATION CONFIGURATION

Binder, enzyme (AME), and antibody models use the same Co-DIRECT optimization procedure and forward computation. Task data, molecular conditions, and masks vary, with shared binder/antibody and separate AME initial checkpoints. The shared training procedure uses 800 optimizer updates, microbatch size 1, accumulation of up to eight microbatches per update, and a 50-update warmup. These are optimizer updates, not the 400 sampling steps. AdamW updates trainable trunk parameters at 10^{-5} with zero weight decay and gradient clipping at 1.0. Adam updates the router at 10^{-3}. Optimizer moments follow PyTorch defaults. All trunk parameters are trainable unless the variant freezes them, including base weights, biases, unmerged LoRA factors, and layers outside LoRA. The autoencoder stays fixed. Inherited LoRA uses rank 32 and scaling parameter 64, with its weights kept separate from the base weights.

Full uses flow matching, contact loss with coefficient 2, response loss with coefficient 1, and geometric router supervision with coefficient 1. The same learning rates and update budget apply to Full, ordinary full adaptation, and retrained controls across all three families, with only the designated parameters and objectives changed. These shared settings were selected using validation results from both Full and the baselines. Family-specific inputs are summarized in Table 2, and changes for each control are specified in Appendix D.

Binder and antibody adaptation start from the same pretrained Proteina-Complexa checkpoint, while enzyme adaptation starts from the enzyme-specialized Proteina-Complexa checkpoint and uses the AME v6 training split. These are also the frozen initial checkpoints evaluated in Figure 2, with their saved LoRA parameters retained. Each task uses its own training split and molecular conditions. The enzyme training configuration uses seed 1401. Training-seed uncertainty is discussed in Appendix D.6.

Table 2: Family inputs under the common optimization procedure. Variant changes are in Table 15.

Family	Data and molecular context	Generator initialization
Binder	Binder training split; target protein and supplied motif conditions	Shared binder/antibody Proteina-Complexa checkpoint
Enzyme (AME)	AME v6 training split; ligand and theozyme conditions	Enzyme-specialized Proteina-Complexa checkpoint
Antibody	Antibody training split; antigen, framework, and designed H3 mask	Shared binder/antibody Proteina-Complexa checkpoint

The shared router is an MLP with dimensions $9 \to 64 \to 64 \to 3$ and SiLU hidden activations. Its outputs are a contact logit and two softplus channel scores. No amino-acid identity, reference sidechain distance, or pooled context features are included in its input. The same architecture is used for all three task families.

Enzyme generation uses a chain length of 180. Candidate counts, refolding, and selection are defined in Appendix B.

A.9 SOFTWARE AND MODEL COMPONENTS

The generator uses the Proteina-Complexa implementation built on La-Proteina (Didi et al., 2026b; Geffner et al., 2026).[1] PyTorch provides tensor operations, automatic differentiation, the MLP, and the optimizers (Paszke et al., 2019). The inherited low-rank layers use `loralib`, the reference LoRA implementation (Hu et al., 2021).[2] Because base weights are also trainable, using this library does not make the adaptation LoRA-only. Lightning supplies the upstream model/training infrastructure (Falcon et al., 2020), and OmegaConf handles hierarchical configuration. AdamW, Adam, and SiLU are attributed separately to their method references (Loshchilov & Hutter, 2017; Kingma & Ba, 2014; Elfwing et al., 2018).

Refolding uses ESMFold2-Fast through the `esm` package (Candido et al., 2026), distinct from the original ESMFold release (Lin et al., 2023). RDKit enumerates ligand automorphisms for

[1] https://github.com/NVIDIA-BioNeMo/Proteina-Complexa

[2] https://github.com/microsoft/LoRA

symmetry-aware atom matching. NumPy provides summary statistics and singular value decomposition for proper Kabsch alignment (Harris et al., 2020; Kabsch, 1976).

Table 3: Software versions used in the experiments and analysis. Versions identify the training, generation, and molecular evaluation environments.

Stage	Component	Version
Training/generation	CPython; PyTorch	3.12.13; 2.7.0+cu126
Training/generation	`loralib`; Lightning	0.1.2; 2.5.6
Training/generation	OmegaConf; NumPy	2.3.1; 1.26.4
Refolding	`esm`; ESMFold2-Fast model	3.4.0
Molecular evaluation	RDKit	2026.03.5

B Benchmark Sets and Evaluation Procedures

The study uses two complementary evaluations. Figure 2 measures prediction changes at corrupted reference structures, while Figures 3 and 4 evaluate generated designs and method variants. Their cohorts, sampling units, and outcomes are specified separately below.

B.1 Sequence-separated data partitions

Training, validation, and test sets are constructed to prevent sequence leakage, with sequence identity between partitions at or below 30%. Test examples do not enter adaptation or validation-based model selection. This split restriction concerns protein sequences. It does not imply that ligand chemotypes or broad task categories are disjoint. Figure 2 uses the held-out diagnostic cohort, prioritizing proteins not present in the original Proteina-Complexa training set. This upstream-training criterion is distinct from the train/validation/test separation for Co-DIRECT. Loader coverage and exclusions are reported in Appendix C.

B.2 Figure 2 cohort and paired predictions

Figure 2 evaluates corrupted reference structures from the held-out benchmark set. Proteins absent from the original Proteina-Complexa training set are prioritized, and every reference accepted by the data loader is measured. Table 4 reports coverage. Loader exclusions concern unsupported atom or residue encodings, cropping that removes the partner, or missing selected chains. No predicted structure replaces a missing reference.

Table 4: Diagnostic coverage. References and source parents differ from the generation benchmark.

Task	In split	No structure	Loader excluded	Measured	Parents
Binder	198	18	27	153	92
AME	292	6	70	216	111
Antibody	1106	55	9	1042	430

The diagnostic evaluates the frozen pretrained initialization used for each task. Binder and antibody evaluations use the same Proteina-Complexa checkpoint, while enzymes use the enzyme-specialized Proteina-Complexa checkpoint. These checkpoints also initialize the corresponding Co-DIRECT models (Appendix A.8). We retain the saved LoRA parameters, matching autoencoder, and ordinary Proteina self-conditioning. These initial models use neither a router nor geometry snap. The uniform correction below is a diagnostic intervention on its predictions, restricted to designed, valid residues. This directly measures response benefits and costs in the models from which adaptation starts.

At step 0, self-conditioning keys are omitted. At every later step, both channels receive the detached clean estimates from the preceding original-state (Jacobi) call. The reference endpoint and noisy state are never substituted for these estimates. Omitting a coordinate key gives zero self-conditioning pair features in Proteina, whereas supplying zero coordinates creates zero-distance bin features. The diagnostic therefore preserves key omission when no preceding prediction exists. The primary

diagnostic follows this sampling-time rule, with self-conditioning present after step 0, rather than the 50% dropout used in training.

The original sampling schedule has 400 calls, indexed 0–399. Coordinate time follows a logarithmic schedule and latent time a power schedule, both with parameter 2. At each scheduled time, the same reference structure and residue latents are forward-corrupted to form the current state. Self-conditioning follows the Proteina loop across these states, while the states themselves are constructed from the reference rather than integrated from predicted velocities. Eight indices with a preceding time increment were fixed before measurement (Table 5), with four corruption draws per reference and index. The preceding schedule index supplies the probe increment. Original and corrected predictions share the scoring state, reference, times, masks, and incoming self-conditioning.

Table 5: Prespecified call indices, evaluation times from the original schedule, and mean channel increment h_k. The eight-index mean is neither a training-time expectation nor a complete average over all sampler calls.

Call	t_x	t_z	h_k
1	0.011563	0.000006	0.005784
58	0.492059	0.021025	0.003359
115	0.741341	0.082656	0.002272
172	0.870668	0.184900	0.001879
228	0.936926	0.324900	0.001846
285	0.972138	0.507656	0.001998
342	0.990406	0.731025	0.002248
399	0.999883	0.995006	0.002550

Corruption seeds are fixed by reference, call index, and draw. Backbone interpolation includes stochastic centering with scale 0.2 nm. The same original prediction and clipped correction are used for $\alpha \in \{0, 0.25, 0.5, 1\}$, with clipping applied before scaling. The primary measurement uses $\alpha = 1$. Enzyme motif selection uses a fixed seed of 20260922 so that conditioning and design masks are reproducible.

B.3 Uniform probe and prediction-error accounting

The diagnostic in Figure 2 uses the frozen initial checkpoint for each task. Let q_k denote its incoming self-conditioning at schedule index k, with $q_0 = \varnothing$ denoting omitted inputs. It advances both blocks by $h_k v$, re-evaluates with q_k held fixed, and clips the response separately by residue and block

$$h_k = \tfrac{1}{2} \sum_{m \in \{x,z\}} (t_m^{(k)} - t_m^{(k-1)}), \tag{15}$$

$$\delta = f_\theta(s + h_k v, t, c, q_k) - v, \tag{16}$$

$$\mathcal{C}_{m,i}(\delta) = \begin{cases} \delta_{m,i} \min\{1, 0.5\|v_{m,i}\|_2 / \|\delta_{m,i}\|_2\}, & \|\delta_{m,i}\|_2 > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

The corrected diagnostic prediction is $v + \alpha D\mathcal{C}(\delta)$, where $D_i = 1$ for designed residues and $D_i = 0$ otherwise, with the mask broadcast over both blocks and their feature dimensions. No probe is applied at the first call or when the time increment is nonpositive. For a smooth field, $\delta_x = h_k[(\partial_x f_x)v_x + (\partial_z f_x)v_z] + O(h_k^2)$, which includes both self- and cross-modal dependence. These definitions do not describe the snap update in Section 4, and the clipping bound above does not apply when the snap method replaces a follower prediction.

We implement this comparison with frozen Proteina-Complexa (Didi et al., 2026b). Its state $s = (x_{t_x}, z_{t_z})$ contains backbone coordinates x and per-residue latents z, which encode sequence and non-Cα atomistic information. Given times $t = (t_x, t_z)$, molecular context c, and incoming self-conditioning q_k, the model predicts

$$v_x = f_\theta^x(x_{t_x}, z_{t_z}, t, c, q_k), \qquad v_z = f_\theta^z(x_{t_x}, z_{t_z}, t, c, q_k). \tag{18}$$

Both predictions depend on both states through shared residue representations and pair-biased attention (Geffner et al., 2026, Appendix H).

Writing $v = (v_x, v_z)$, the response is $f_\theta(s + h_k v, t, c, q_k) - v$. Per-residue, per-component clipping and the design mask give correction d (Equation 17). We compare v and $v + \alpha d$ at $\alpha = 1$, with $\alpha \in \{0, 0.25, 0.5, 1\}$ used for sensitivity analysis. These are scored diagnostic predictions; the self-conditioning history continues to use the uncorrected Jacobi clean estimates. Co-DIRECT instead uses the directional probe and selected follower replacement in Section 4.

Let $\ell_{m,i}(a; s)$ be the original flow-matching loss contribution for prediction a, component m, and residue i at input s. It preserves the model's conversion to clean-reference predictions, feature weights, valid-residue denominator, masks, and time weights. Define

$$\Delta_{m,i}(s) = \ell_{m,i}(v; s) - \ell_{m,i}(v + \alpha d; s), \tag{19}$$

$$B_s = \sum_{m,i} [\Delta_{m,i}]_+, \qquad H_s = \sum_{m,i} [-\Delta_{m,i}]_+, \qquad G_s = B_s - H_s, \tag{20}$$

where $[q]_+ = \max(q, 0)$. Benefit B_s and cost H_s exactly recover the total error reduction through $G_s = B_s - H_s$. Costs refer to a component, residue, and noisy state, not a fixed set of bad residues. Signs are separated before averaging over noise, time, references, and source parents. Aggregation E gives each parent equal weight. Uncertainty jointly resamples parents.

Molecular contacts impose local design requirements (Pacesa et al., 2025; Ahern et al., 2026; Bennett et al., 2026). Using heavy atoms in reference structures, we inspect binder interfaces within 5 Å of the target, enzyme residues within 6 Å of the ligand, and antigen contacts within 5 Å among designed antibody CDR-H3 residues. The enzyme region is a proximity proxy, not an annotated catalytic site. These post-hoc regions interpret the diagnostic. They do not add router inputs or training weights.

For region R, restrict the sums to obtain B_R, H_R, G_R and original error L_R. Table 10 reports $100E[Q_R]/E[L_R]$ for $Q \in \{B, H\}$; Figure 2 reports the descriptive ratio $100E[H_R]/E[B_R]$. References must contain both regional and other designed residues. Table 6 gives sample counts, and Appendix C.4 reports comparisons with other designed residues.

B.4 Loss reduction and uncertainty

We score both predictions through the model's original conversion from velocities to clean-sample predictions. For block $m \in \{x, z\}$, reference $y_{1,m}$, and candidate velocity a, write this conversion as $\widehat{y}_{1,m}(a; s)$. The measured loss is the sum of the following terms

$$\ell_{m,i}(a; s) = \frac{\sum_j \left[M_{m,i}(y_{1,m,i,j} - \widehat{y}_{1,m,i,j}(a; s))\right]^2}{n_m\left((1 - t_m)^2 + 10^{-5}\right)}, \quad n_m = \sum_i M_{m,i}, \tag{21}$$

where M is the model's valid-residue mask and j indexes features. The design mask restricts the correction, not the loss denominator. Thus $\ell = \sum_{m,i} \ell_{m,i}$ recovers the model's total loss, with no auxiliary terms in this configuration.

The diagnostic retains the original feature sums, valid-residue denominators, clean-prediction conversion, and stabilized time weights. Residues outside the design mask retain their original prediction and have zero error change, although their original error remains part of the full loss.

For each state and draw, we compute $\Delta_{m,i}$, B_s, H_s, and G_s without thresholds. Positive/negative splitting occurs before any averaging. Let reference e belong to parent p, with n_p references, $K = 8$ indices and $R = 4$ draws. For any state statistic q the task summary is

$$E[q] = \frac{1}{P} \sum_{p=1}^{P} \frac{1}{n_p} \sum_{e \in p} \frac{1}{K} \sum_{k=1}^{K} \frac{1}{R} \sum_{r=1}^{R} q_{e,k,r}. \tag{22}$$

This average with equal weight per source protein is distinct from recovering the original loss of a particular batch. We draw 10,000 parent-bootstrap replicates with seed 20260921, preserving all observations and policy values within each sampled parent. Percentile 95% intervals use the same resamples for B, H, G, decompositions, and ratios. Residues, noise draws, and related references are not independent bootstrap units. Per-index intervals are marginal. No simultaneous coverage across the eight indices is claimed.

The design-region baseline is the sum of original loss terms inside the design mask, with the original valid-residue denominator retained. Relative effects divide the parent-weighted effect mean by the corresponding baseline mean. They are not averages of statewise ratios. Table 7 reports the resulting scale. An undefined ratio is not replaced with zero.

B.5 FUNCTIONAL-REGION DEFINITIONS AND PAIRED COMPARISONS

The regional analysis was conducted after inspecting the aggregate diagnostic. We fixed region definitions before associating them with loss changes, but the analysis remains post hoc. Regions use clean reference structures, independent of corrupted states and corrected predictions. We restrict each geometric region to valid designed residues using the same mask as the correction. Fixed target, ligand-context, and framework residues may define the partner geometry but do not contribute designed-region numerators or baselines.

For binders, the partner comprises the complete target chains specified by the task. For antibodies, it comprises the whole declared antigen. We define contact by minimum heavy-atom distance of at most 5 Å, with 4 and 6 Å sensitivity thresholds. The antibody strata divide the designed CDR-H3 window into antigen-contacting and non-contacting residues. The fixed framework is not a comparison group. For enzymes, the partner is the ligand instance identified by the PLINDER system identifier, requiring at least six observed heavy atoms. The primary proximity threshold is 6 Å, with 4 and 8 Å sensitivities.

Regions use heavy-atom distances to the whole declared partner in the source structure, rather than a cropped partner. Source chain and residue identities map these annotations to model residues. Coordinate checks require at least 95% matching of valid residues and maximum discrepancy below 0.01 Å after unit conversion. Ambiguous or missing mappings are treated as unknown rather than non-contact. All reported regional references pass these checks.

The enzyme region denotes ligand proximity rather than annotated catalytic activity. Conditioning motifs in this reference cohort are not catalytic annotations. Binder interfaces and antibody antigen contacts likewise describe observed geometric contacts in the reference structure.

Table 6: Sample counts for the regional analysis. Reference/source-protein counts are shown together. Region/design counts give residue positions before repeated draws and times. Every designed position could be assigned a region using observed source atoms. Regions are defined in this subsection.

Task	Full refs/parents	Paired refs/parents	Region/design slots
Binder	153/92	148/90	4,359/18,779
Enzyme	216/111	185/99	2,588/26,915
Antibody	1042/430	1017/421	6,574/15,370

Additive attribution and paired regional contrasts. Partitioning the full cohort by region preserves each original residue–channel loss contribution and its original valid-residue denominator. It introduces no region-size renormalization. Summing the disjoint strata recovers the design-mask totals at each state and draw. This attribution uses all 153/216/1,042 binder/enzyme/antibody references from 92/111/430 parents.

For comparisons, we restrict to references containing at least one valid contribution in both the interest region R and other designed residues O. This shared subset contains 148/185/1,017 references from 90/99/421 parents. The numerator and baseline of each regional ratio, and both sides of its contrast, use these same references and the original nested averaging procedure with equal parent weights. With $L_R = \sum_{m,i\in R} \ell_{m,i}(v; s)$, we compute

$$r_Q(R) = 100\frac{E[Q_R]}{E[L_R]}, \qquad \Delta r_Q = r_Q(R) - r_Q(O), \qquad Q \in \{B, H, G\}. \tag{23}$$

These are ratios of aggregated losses, not averages of statewise ratios. A zero baseline leaves a ratio undefined. All ratios and contrasts are recomputed within each shared parent-bootstrap replicate. The reported 95% intervals are nominal percentile intervals without multiplicity adjustment. Since $G_R = B_R - H_R$, $\Delta r_G = \Delta r_B - \Delta r_H$.

B.6 GENERATED-DESIGN BENCHMARK AND COMMON REFOLDING PROTOCOL

The generation benchmark contains 30 binder, 41 enzyme, and 32 antibody targets. Each method contributes three candidates per target, and each candidate receives five ESMFold2-Fast refolds (Candido et al., 2026). Refolding uses eight loops, 50 steps, seed 1, and no MSA. The refolder receives sequence and the applicable ligand CCD, not generated coordinates. A common family-specific selection rule chooses one candidate and one refold for every reported metric. Equal candidate and refolding budgets do not imply equal generator-call or training costs.

Binders. We assess binder–target interface ipTM, binder–target interface ipAE, and mean pLDDT over the generated binder chain. Interface confidence and local confidence are widely used alongside structural consistency in binder-design evaluation (Watson et al., 2023; Pacesa et al., 2025). Restricting pLDDT to the binder avoids dilution by an unchanged target. We use ipAE in Å and pLDDT on the 0–100 scale, with any predictor-specific normalization explicitly converted. Pair-specific interface scores refer to the declared binder–target pair rather than a global score that also includes other interfaces. The term interface identifies the interacting pair. It does not assert that the underlying PAE matrix is symmetric. Interpreting an interface PAE score also depends on how cross-partner matrix entries and directional summaries are averaged.

Evaluated generation stages. For BoltzGen, we use diffusion generation followed by inverse folding, with its native Boltz-2 refolding and filtering/ranking stages disabled. This is distinct from the full released pipeline (Stärk et al., 2025a). Proteina-Complexa is likewise evaluated without reward-guided search or an additional downstream screening pipeline. Each method contributes three candidates per target. The shared ESMFold2-Fast evaluation produces five refolding samples per candidate and applies the common family-specific selection rule. The selected structure supplies all reported metrics. No additional method-specific reward, refolding, or filtering budget is used before this common evaluation.

A common selected refold for enzyme evaluation. We report motif-fit RMSD and ligand-fit RMSD first, followed by protein–ligand interface ipTM as a complementary confidence measure. All three enzyme endpoints are obtained from the same selected ESMFold2-Fast refold of the designed protein with its ligand, without an MSA. Each of $G = 3$ design candidates receives five refolding samples. The family-specific selection rule first favors candidates passing every gate, then the minimum normalized endpoint score, and finally the mean normalized score, with higher normalized scores preferred. The hierarchy selects a refolding sample and a design candidate for each task. The reported values are task means over the selected structures, not separately optimized means for each metric. Generation coordinates are not substituted into either RMSD. Gate failure does not automatically exclude a task from the reported mean when no passing candidate is available.

Catalytic-atom correspondence. The scored motif contains the non-hydrogen atoms named in the task's theozyme, the specified catalytic-site model. Generator metadata map each reference motif residue to a zero-based design-chain index. The corresponding refold residue uses the one-based sequence identifier. Atom pairs are matched by reference motif key and atom name, not by sequence alignment or native residue numbering. Missing atoms are omitted and their counts recorded. Missing correspondence metadata or fewer than three matched motif atoms makes both RMSDs unavailable. Unavailable quantities fail the applicable gates rather than becoming zeros. These are named catalytic-atom measurements, including specified side-chain atoms, not Cα-only backbone RMSDs. This atom-level emphasis follows theozyme scaffolding, although the exact alignment convention below differs from the original AME evaluation (Ahern et al., 2026).

Motif-fit and ligand-fit RMSD. Let p_j and q_j, $j = 1, \ldots, K$, be corresponding motif atoms in the selected refold and task reference. The motif-fit score, called self in the evaluation output, is

$$\mathrm{RMSD}_{\text{motif-fit}} = \min_{R \in \mathrm{SO}(3),\, b \in \mathbb{R}^3} \left[\frac{1}{K} \sum_{j=1}^{K} \| R p_j + b - q_j \|_2^2 \right]^{1/2} . \tag{24}$$

A proper-rotation Kabsch fit (Kabsch, 1976) uses only those motif atoms, with uniform atom weights and no reflection. This measures the internal geometry of the catalytic atom group after removing its overall rigid displacement. The ligand is absent from the fit.

For the ligand-fit score, obtain (R_L, b_L) by aligning the predicted ligand atoms to the reference ligand atoms. Where valid ligand graph automorphisms are available, the atom permutation is selected by minimizing ligand RMSD alone, without using the motif score. Otherwise the evaluator uses its identity-permutation fallback. A valid ligand atom correspondence is required. Apply the resulting transform to the motif and score it without another alignment

$$\mathrm{RMSD}_{\text{ligand-fit}} = \left[\frac{1}{K} \sum_{j=1}^{K} \| R_L p_j + b_L - q_j \|_2^2 \right]^{1/2} . \tag{25}$$

Both columns therefore measure motif atoms. They differ in which atoms determine the alignment. Ligand-fit RMSD retains errors in motif placement relative to the ligand as well as motif deformation. It is not the ligand's own RMSD. For an identical motif correspondence, motif-fit RMSD cannot exceed ligand-fit RMSD, because the former minimizes over all proper rigid transforms. The scores therefore complement each other without measuring independent error components. An intact motif displaced from its ligand can have small motif-fit RMSD and large ligand-fit RMSD.

Protein–ligand interface confidence. Let M be the predictor's matrix of chain-pair ipTM scores. For the declared design-protein group A and ligand group B, the reported interface score is

$$\mathrm{ipTM}_{A,B} = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} \max\{M_{ab}, M_{ba}\}. \tag{26}$$

Each enzyme task has one protein chain and one ligand chain, so this reduces to one symmetrized pair score. No coordinate alignment is involved. The global scalar ipTM is not used, because it may select a different interface. In this evaluation representation, a CCD ligand contributes one predictor token even though its atomic coordinates contain multiple atoms. The score is therefore a supporting interface-confidence measure, not a direct assay of catalytic geometry or activity, and its calibration is not assumed to match protein–protein ipTM.

Enzyme gates require interface ipTM ≥ 0.35, interface PAE ≤ 18 Å, pLDDT ≥ 70 on the 0–100 scale, motif-fit RMSD ≤ 2.0 Å, and ligand-fit RMSD ≤ 3.0 Å. Interface PAE averages both full cross-partner matrix blocks. Gate thresholds are distinct from the normalization anchors used in ranking, and gate passes are evaluated jointly for each selected structure. A generated candidate with an unavailable required metric fails the corresponding gate. A target with no evaluated candidate is instead recorded as unevaluated and excluded from the gate denominator. The pLDDT threshold of 70 on the 0–100 scale is equivalent to 0.70 on the predictor's 0–1 scale.

Binder and antibody gates. Binder and antibody gates require interface ipTM ≥ 0.50 and interface ipAE ≤ 15.0 Å. pLDDT is not a gate. Antibody interface scores compare the designed H+L group with the antigen and exclude the heavy–light interface. Gates are applied to each selected design using unrounded scores.

Antibody CDR redesign. We compare antibody–antigen interface ipTM and ipAE. Co-DIRECT additionally reports mean pLDDT over redesigned CDR residues (90.84), whereas the baseline outputs provide design-role pLDDT over H+L and complex-wide pLDDT. These are different spatial averages and are not ranked together. The H+L statistic is dominated by fixed framework residues and does not substitute for confidence in the redesigned CDRs. Antibody-design studies also use structural self-consistency and interface assessment (Bennett et al., 2026). We additionally measure framework-aligned CDR-H3 consistency (Appendix B.7), while junction validity is not assessed by that RMSD. These are computational assessments of a design, not direct measurements of affinity.

B.7 Generated–refold structural consistency

Structural consistency of the selected designs. A sequence-only or sequence-and-ligand refold may have favorable geometry without retaining its co-generated backbone. Structural self-consistency tests this separate property, as in prior protein-design evaluations (Watson et al., 2023;

Bennett et al., 2026). We compare the generated backbone of each already selected candidate with its already selected ESMFold2-Fast sample. No candidate or refold is reselected using consistency, and no additional generation, refolding, or ProteinMPNN redesign is performed.

Coordinates, correspondence, and scope. Generated coordinates are taken from PDB or mmCIF exports after decoding, and refold coordinates from the selected ESMFold2-Fast structure. Both are in angstroms. Design chains are identified by their assigned roles and matched in sequence order without searching over residue permutations. Enzyme scRMSD compares the generated and refolded Cα coordinates of the designed 180-residue chain. Ligand atoms, separate motif tokens, and other context coordinates are excluded. Restoring catalytic amino-acid identities changes the submitted sequence but does not replace the generated design-chain Cα coordinates. For antibodies, the supplied framework is restored before export and provides the alignment reference, while the designed CDR-H3 coordinates remain generated.

Alignment and score. Let g_i and r_i be corresponding generated and refolded Cα coordinates, with alignment set A and scoring set S. A proper Kabsch fit (Kabsch, 1976) solves

$$(R^*, b^*) = \arg \min_{R \in \mathrm{SO}(3),\, b \in \mathbb{R}^3} \sum_{i \in A} \|R g_i + b - r_i\|_2^2,$$
$$\mathrm{RMSD}_{A \to S} = \sqrt{\frac{1}{|S|} \sum_{i \in S} \|R^* g_i + b^* - r_i\|_2^2}. \tag{27}$$

The fit permits rotation and translation, with determinant $+1$ and no reflection or scaling. For binder/enzyme scRMSD, $A = S$ contains the mapped design-chain Cα atoms. For antibodies, heavy and light chains map as B$\to$H and C$\to$L using their chain roles. Framework atoms define one rigid transform across the mapped antibody chains. The score uses the designed CDR-H3 span. CDR-H3 is not fitted separately. Each evaluated span contains 13 scored Cα atoms. This is a CDR-H3 endpoint, not an all-CDR or whole-antibody score.

Coverage and missing evaluations. Alignment and scoring coverage are computed separately as the number of matched Cα atoms divided by the expected count. Both equal one for the valid comparisons. Seven method–target cases have no selected design. These are binder BoltzGen and Chroma on `supp_1avg`, and RFdiffusion2 on enzyme tasks `M0050_1dbt`, `M0058_1cju`, `M0093_1dqa`, `M0375_4ts9`, and `M0732_1xs1`. This leaves 29 targets for each of the two binder methods and 36 for RFdiffusion2 scRMSD. Other scRMSD cohorts are complete. The number of valid comparisons differs from that for catalytic-atom RMSDs because Cα correspondence does not require successful catalytic-atom mapping.

B.8 Metric availability

The benchmark contains 30 binder, 41 enzyme, and 32 antibody tasks. Table means use all valid values for each method and metric, whereas differences use valid observations for both methods on the same target and measured region. Binder BoltzGen and Chroma each have one unevaluated target. For enzymes, frozen and full adaptation each have one failed motif mapping. RFdiffusion2 has five unevaluated tasks and eight additional mapping failures. Its ligand ipTM is available on 36 targets and both RMSDs on 28. Co-DIRECT has all three enzyme endpoints on 41 targets. No metric is assigned a zero for a missing observation. Missing required metrics fail the applicable gate. Cases without an evaluated design are reported separately in the coverage counts.

B.9 Target pairing and statistical analysis

We treat each target as the independent unit for design comparisons. Each target contributes one selected candidate/refold pair per method, with separate observations for each metric and measured region. Comparisons pair Full with the comparator on the same target and measured region, using matching evaluation protocols, selection rules, and units. Only pairs with two valid numeric observations enter difference estimates. Missing observations are excluded rather than assigned zero. Their counts and causes are given in Appendix B. Multiple metrics, candidates, and refolds do not increase the number of independent targets. Reference-state diagnostics instead group related reference structures by source protein (the parent), as specified in Appendix D.5.

Difference and interval definitions. For target i and comparator c, the figures and primary paired tables use the same subtraction order

$$\Delta_{i,c} = y_{i,\text{Full}} - y_{i,c}. \tag{28}$$

Positive differences favor Full for ipTM. Negative differences favor Full for ligand-fit RMSD. We report the mean over valid pairs and improved/tied/worse counts using the metric's preferred direction. Numerical ties satisfy $|\Delta| \leq 10^{-8}$. This is an arithmetic tolerance, not a practical-effect threshold. For each comparison, we resample paired targets with replacement 10,000 times and report the 2.5th and 97.5th percentiles of the resampled means (Efron, 1992). Calculations use unrounded observations and fixed random seeds. Candidates and refolds from the same target do not count as independent targets.

Intervals are reported separately for each comparison without adjustment for multiple comparisons. They describe variation across evaluated targets conditional on the fitted models. They do not quantify training-seed variability, and biological relatedness between targets can affect the independence assumption.

C Additional Results for the Figure 2 Diagnostic

The tables below extend Figure 2 with uncertainty estimates, deterioration thresholds, regional contrasts, and changes across sampling times.

C.1 Benefits and costs relative to the original error

Table 7: Effect size relative to baseline loss. Percentages are ratios of parent-weighted means. Total change is $E[B + H]$, the sum of benefit and cost rather than their net difference.

Task	$E\ell_{\text{full}}$	$E\ell_{\text{design}}$	Total/full	Total/design
Binder	36.250	17.458	0.1748%	0.3629%
AME	75.189	40.748	0.1361%	0.2512%
Antibody	218.386	13.002	0.0067%	0.1127%

Benefits and costs are comparable in magnitude even though their combined size is small relative to the original prediction error. Total change over the design region is 0.3629%, 0.2512%, and 0.1127% for binders, enzymes, and antibodies. The corresponding 95% intervals are $[0.2556, 0.4937]\%$, $[0.1552, 0.4792]\%$, and $[0.0914, 0.1349]\%$. Thus a small net effect can conceal offsetting improvements and deteriorations.

C.2 Within-state and across-state cancellation

To locate cancellation, let E denote the same weighted aggregation and define

$$\begin{aligned} T &= E[B_s + H_s] - |E[G_s]| = W + A, \\ W &= E[B_s + H_s - |G_s|], \qquad A = E|G_s| - |E[G_s]|. \end{aligned} \tag{29}$$

$W = 2E\min(B_s, H_s)$ counts both sides cancelled within a state and draw. A includes cancellation across draws, indices, references, and parents. The post-hoc within-state shares W/T are 63.1% $[55.7, 73.1]$, 67.6% $[62.5, 72.6]$, and 17.0% $[14.9, 19.4]$. Summing modalities before splitting signs reduces W but leaves within-state cancellation across residues in every task (Appendix C).

This exploratory decomposition was developed after the initial measurement. $W/2$ is the amount cancelled on either side within a state, whereas A combines cancellation across draws, times, references, and parents. Both terms use the same aggregation weights.

The across-state term A is identical under both splits because G_s is unchanged. Its values are 0.02262, 0.03066, and 0.01121 for binder, AME, and antibody. Residue summation removes within-residue opposition between modalities but leaves opposition across residues. For binders, the interval for W/T after summing channels includes 50%, so neither within-state nor across-state cancellation clearly dominates. Neither split measures the recoverable gain of a learned policy.

Table 8: Cancellation by location. The primary split is over modality–residue terms. The alternative sums modalities at each residue before splitting signs. Values in brackets are 95% parent-bootstrap intervals.

Task	Split	W	W/T (%)
Binder	Block–residue	0.03861 $[0.03476, 0.04259]$	63.1 $[55.7, 73.1]$
	Residue-summed	0.02882 $[0.02565, 0.03213]$	56.0 $[48.1, 67.2]$
AME	Block–residue	0.06401 $[0.05863, 0.06952]$	67.6 $[62.5, 72.6]$
	Residue-summed	0.05072 $[0.04640, 0.05508]$	62.3 $[56.8, 67.8]$
Antibody	Block–residue	0.00229 $[0.00216, 0.00243]$	17.0 $[14.9, 19.4]$
	Residue-summed	0.00134 $[0.00123, 0.00145]$	10.7 $[9.1, 12.6]$

Both signs occur in 100%, 100%, and 99.03% of state–draw observations for binders, enzymes, and antibodies. Across four corruption draws, all-positive or all-negative patterns account for 31.1%, 31.0%, and 48.2% of matched residue–channel pairs. These descriptive frequencies include arbitrarily small changes. The magnitude thresholds below quantify how often deterioration exceeds a specified error scale.

C.3 Residue-level deterioration frequency and sensitivity

Definition and aggregation. The residue-frequency analysis reuses the stored per-residue losses from the same frozen-model measurements. It introduces no new model evaluations. For a designed residue i, first sum the two component changes to obtain the total error increase $a_i(s) = -\sum_m \Delta_{m,i}(s)$. Let D_s be the designed-residue set and define the mean original error per designed residue as $\bar{\ell}_s = |D_s|^{-1} \sum_{i \in D_s, m} \ell_{m,i}(v; s)$. The fraction exceeding a threshold τ is

$$p_s(\tau) = \frac{1}{|D_s|} \sum_{i \in D_s} \mathbf{1}\{a_i(s) > \tau \bar{\ell}_s\}, \qquad \tau \in \{0, 10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}\}. \tag{30}$$

The zero threshold counts any strictly positive increase. Nonzero thresholds use the mean original error over all designed residues in that evaluation, not each residue's own baseline and not a separate baseline for each region. Zero or undefined baselines leave the corresponding rates undefined rather than zero. For a regional rate, restrict the count and residue denominator to that region while retaining the same whole-design scale $\bar{\ell}_s$. Empty regions have undefined rates. Paired regional comparisons require both regions.

Rates are averaged over four noise draws, eight evaluated times, references, and source parents, with equal final parent weights. The uncertainty uses 10,000 shared parent-bootstrap replicates (seed 20260921). Residues and repeated evaluations are not treated as independent samples. Intervals are nominal and unadjusted across the six thresholds. These thresholds quantify sensitivity to effect size and do not represent biological-importance cutoffs. The full cohorts contain 18,779/26,915/15,370 designed residue positions across reference structures for binder/enzyme/antibody, each evaluated 32 times. Positions are counted within each reference. Overlapping references may include the same biological site. The reported frequencies are per-evaluation fractions, not the fraction of residues ever or always harmed.

Relation to signed loss accounting and regional interpretation. Benefit and cost in Equation 20 split signs before summing components, whereas the counting analysis sums components first. A residue may therefore contribute to cost while its total error improves. The two summaries measure complementary aspects of the same prediction changes.

At the zero threshold, region-minus-other deterioration frequencies differ by $+0.586$ percentage points for binder interfaces ($[-0.963, 2.223]$), $+0.093$ for enzyme ligand proximity ($[-1.371, 1.537]$), and -2.276 for antigen contacts in designed CDR-H3 ($[-2.992, -1.573]$). The binder and enzyme intervals include zero, while antibody contacts have a lower rate. Costs therefore occur at important regions without a consistent excess deterioration frequency. At the 5% threshold, the antibody rate rounds to zero but is 0.00364% before rounding ($[0.00070, 0.00826]$).

Table 9: Fraction of designed residues whose total prediction error increases beyond the indicated percentage of the same evaluation's mean original error per designed residue. Entries are parent-weighted percentages with nominal 95% parent-bootstrap intervals. The zero and 1% thresholds supply the frequency rows in Figure 2.

Threshold	Binder	Enzyme	Antibody
0%	44.99 [43.50, 46.51]	47.86 [46.86, 48.90]	56.60 [55.11, 58.06]
0.01%	43.20 [41.80, 44.58]	46.38 [45.24, 47.47]	52.88 [51.38, 54.35]
0.1%	30.99 [29.34, 32.57]	35.85 [34.25, 37.33]	30.23 [29.04, 31.45]
0.5%	9.73 [8.43, 11.02]	14.87 [13.50, 16.20]	3.34 [2.94, 3.78]
1%	4.39 [3.61, 5.19]	7.79 [6.91, 8.67]	0.39 [0.32, 0.48]
5%	0.40 [0.27, 0.55]	0.87 [0.69, 1.06]	0.00 [0.00, 0.01]

C.4 Effects at molecular contacts

Tables 10 and 11 compare each functional region with the other designed residues on the same references. Region definitions and sample counts appear in Appendix B.5. Costs are present within binder interfaces, ligand-proximal enzyme residues, and antigen contacts in designed CDR-H3. These measurements distinguish the presence of costs from enrichment of costs in a particular region.

Table 10: Regional effects relative to each region's original error, using matched reference subsets (percent). Each column uses the region's own baseline. Table 11 gives joint-bootstrap contrasts and uncertainty.

Task	Region	r_B (%)	r_H (%)	r_G (%)
Binder	Interface	$+0.151278$	$+0.153920$	-0.002642
	Other designed	$+0.194838$	$+0.178013$	$+0.016825$
Enzyme	Ligand-proximal proxy	$+0.173860$	$+0.182946$	-0.009086
	Other designed	$+0.101136$	$+0.114860$	-0.013724
Antibody	Antigen contact in H3	$+0.061145$	$+0.046708$	$+0.014436$
	Other designed	$+0.060525$	$+0.055277$	$+0.005249$

Table 11: Regional differences on matched reference subsets subtract other designed residues from the region of interest and are expressed in percentage points. Intervals are nominal 95% shared parent-bootstrap intervals. Positive Δr_G means greater relative improvement. Positive Δr_H means greater relative harm.

Task	Quantity	Contrast (pp)	95% interval
Binder	Δr_B	-0.043560	$[-0.096552, -0.008132]$
	Δr_H	-0.024093	$[-0.075699, +0.006778]$
	Δr_G	-0.019467	$[-0.044057, +0.001454]$
	Δr_{B+H}	-0.067653	$[-0.167236, -0.004088]$
Enzyme	Δr_B	$+0.072723$	$[+0.023766, +0.114881]$
	Δr_H	$+0.068086$	$[+0.004173, +0.117506]$
	Δr_G	$+0.004638$	$[-0.010753, +0.025889]$
	Δr_{B+H}	$+0.140809$	$[+0.028744, +0.231316]$
Antibody	Δr_B	$+0.000619$	$[-0.003137, +0.005554]$
	Δr_H	-0.008569	$[-0.012814, -0.003568]$
	Δr_G	$+0.009188$	$[+0.006477, +0.012559]$
	Δr_{B+H}	-0.007949	$[-0.015233, +0.001527]$

Regional findings and sensitivity. Table 11 gives the complete primary contrasts. At binder interfaces, relative benefit is lower, whereas the net and harm intervals include zero. The net-contrast estimates remain negative across 4, 5, and 6 Å, but the intervals at the primary 5 Å and 6 Å thresholds include zero. Enzyme ligand-proximal residues show increased relative benefit and harm with an unresolved net contrast. In the recognized cofactor/nucleotide subset (25 paired references from 24

parents), the harm contrast is $+0.003299$ pp ($[-0.039846, +0.059733]$). Antibody contacts within designed H3 have relative net improvement of 0.014436%, compared with 0.005249% outside contacts. The positive net contrast and intervals excluding zero persist across the 4, 5, and 6 Å thresholds.

Table 12: Exploratory contact-cutoff sensitivity of the regional net contrast. † denotes the unchanged primary cutoff. The matched reference subset is formed separately at each cutoff and may therefore change. Intervals are nominal 95% intervals.

Task	Cutoff (Å)	Δr_G (pp)	95% interval
Binder	4	-0.019661	$[-0.041460, -0.000600]$
Binder	5†	-0.019467	$[-0.044057, +0.001454]$
Binder	6	-0.018337	$[-0.043936, +0.002008]$
Enzyme	4	$+0.010045$	$[-0.003702, +0.035173]$
Enzyme	6†	$+0.004638$	$[-0.010753, +0.025889]$
Enzyme	8	$+0.002714$	$[-0.014609, +0.022365]$
Antibody	4	$+0.008333$	$[+0.005690, +0.011715]$
Antibody	5†	$+0.009188$	$[+0.006477, +0.012559]$
Antibody	6	$+0.009772$	$[+0.006971, +0.013290]$

Exploratory block summaries. We additionally summarize regional effects separately for the backbone and residue-latent blocks over the full cohort. These summaries use every reference in the cohort. They therefore complement the paired regional comparisons above, which use a restricted subset, but do not estimate the same quantity. Each block uses its own regional baseline. Its relative contrast cannot be added to another block's contrast to recover the combined relative effect. For example, binder interface-minus-other net contrasts are -0.034359 pp ($[-0.074529, -0.004363]$) for the backbone and $+0.014499$ pp ($[+0.006578, +0.023210]$) for residue latents. Opposite signs in these exploratory summaries suggest that a region-level average can conceal differences between computational blocks, with different behavior in the two channels. Table 13 reports all task–block combinations with nominal, unadjusted intervals.

Table 13: Exploratory block-specific net contrasts on the full cohorts (92/111/430 parents), rather than the matched reference subset in Table 10. Each block uses its own regional baseline. Relative contrasts cannot be summed across blocks. Intervals are nominal 95% intervals.

Task	Block	Δr_G (pp)	95% interval
Binder	Backbone	-0.034359	$[-0.074529, -0.004363]$
	Residue latent	$+0.014499$	$[+0.006578, +0.023210]$
Enzyme	Backbone	$+0.007996$	$[-0.012231, +0.034663]$
	Residue latent	$+0.012742$	$[-0.002446, +0.027496]$
Antibody	Backbone	$+0.010312$	$[+0.007192, +0.014132]$
	Residue latent	-0.000887	$[-0.004366, +0.002465]$

C.5 Changes across sampling times

Table 14 resolves the net effect at the eight prespecified calls. The sign changes across times in all three tasks. These per-time estimates show why the aggregate should be interpreted over the specified schedule rather than as a uniform improvement or deterioration at every step.

Table 14: Net effects over all designed residues at the eight prespecified sampling calls, in the original loss units. Intervals are marginal 95% parent-bootstrap intervals. No simultaneous coverage or selected-time claim is implied.

Task	Call	Net effect G	95% interval
Binder	1	-0.025842	$[-0.060675, +0.002635]$
	58	$+0.002498$	$[-0.017439, +0.021590]$
	115	$+0.003825$	$[-0.001590, +0.009083]$
	172	$+0.005908$	$[+0.002570, +0.009397]$
	228	$+0.007225$	$[+0.004371, +0.010295]$
	285	$+0.009301$	$[+0.006380, +0.012369]$
	342	$+0.013987$	$[+0.009571, +0.018352]$
	399	$+0.000119$	$[-0.000025, +0.000261]$
Enzyme	1	-0.083450	$[-0.135114, -0.040098]$
	58	$+0.045081$	$[+0.026308, +0.062637]$
	115	$+0.013417$	$[+0.006448, +0.020553]$
	172	$+0.004171$	$[-0.000230, +0.009541]$
	228	-0.006418	$[-0.010643, -0.001659]$
	285	-0.022133	$[-0.027085, -0.016886]$
	342	-0.011588	$[-0.016310, -0.006764]$
	399	-0.000610	$[-0.000849, -0.000365]$
Antibody	1	$+0.048563$	$[+0.041435, +0.055972]$
	58	-0.016422	$[-0.019427, -0.013630]$
	115	-0.009648	$[-0.011074, -0.008254]$
	172	-0.005233	$[-0.006105, -0.004379]$
	228	-0.003436	$[-0.004134, -0.002736]$
	285	-0.002818	$[-0.003478, -0.002159]$
	342	-0.001706	$[-0.002316, -0.001103]$
	399	-0.000062	$[-0.000088, -0.000037]$

D Detailed Definitions of Model Variants

The controls distinguish three questions. Component retraining tests which parts of the complete method contribute to the fitted model. Inference interventions test how a fixed trained model uses its responses. Objective controls test whether additional supervision helps and whether its placement matters. Tables 15 and 16 specify what changes in each case.

D.1 Ordinary adaptation and inference budget

Ordinary full adaptation trains the trunk with the original flow-matching objective and retains ordinary self-conditioning. It excludes the additional contact, response, and router losses. This baseline compares the complete method with standard adaptation of the same generator family. The 799-call control reuses that adaptation checkpoint and increases ordinary sampling from 400 to 799 steps. It has no router or anticipation query. Its purpose is to test whether additional inference computation alone reproduces the gains of Co-DIRECT, whose 400 steps use 800 generator calls. The comparison matches call counts approximately rather than wall time, memory, or training cost.

D.2 Retrained component controls

The no-anticipation control retains the trainable trunk, router, weighted self-conditioning, and contact and router objectives. It removes the probe call and response objective and makes one prediction per sampling step.

The no-router control removes the learned router, its contact-classification and leadership-regression objectives, residue-wise follower assignments, and router weighting of self-conditioning. The generator retains ordinary clean-prediction or zero self-conditioning. It is trained with $\mathcal{L}_{\mathrm{FM}}+2\mathcal{L}_{\mathrm{interface}}+\mathcal{L}_{\mathrm{ant}}$. Thus, the generator's contact-region supervision remains present even though the separate router objectives are removed. A symmetric extra-pass supplies the second prediction without assigning either channel leadership. Geometry snap is disabled in both training and generation.

Table 15: Variant settings shared across the three families. Every fitted variant retains flow matching. The autoencoder is fixed. Contact/response/router entries give added-loss coefficients. "Ordinary" means clean-prediction/zero self-conditioning without router weights, not removal of self-conditioning. Inference controls reuse a fitted checkpoint and add no training updates.

Variant	Trained parameters	Contact / response / router	Self-conditioning	Checkpoint use
Full	Trunk + router	2 / 1 / 1	Router-weighted	Family initialization; shared training procedure
Full adaptation	Trunk	0 / 0 / 0	Ordinary	Family initialization; flow-only adaptation
No router	Trunk	2 / 1 / 0	Ordinary	Retrained; symmetric extra-pass, $\alpha = 1$
No anticipation	Trunk + router	2 / 0 / 1	Router-weighted	Retrained; one prediction per step
Frozen trunk	Router	2 / 1 / 1	Router-weighted	Initial trunk held fixed
No response loss	Trunk + router	2 / 0 / 1	Router-weighted	Retrained; probe retained at inference
Original-state auxiliary	Trunk + router	2 / 1 at Call 1 / 1	Router-weighted	Retrained; probe retained at inference
Fixed direction, trained	Trunk	2 / 1 / 0	Ordinary	Retrained; fixed roles
Fixed/shuffled roles	None additional	Full training objective	Router-weighted	Full checkpoint; inference only
Adaptation, 799 calls	None additional	Adaptation objective	Ordinary	Adaptation checkpoint; inference only

Table 16: Interpretation of Figure 4 controls. A new training run and a changed inference policy answer different questions.

Control	Change from Full	Question addressed
No router	No learned router or roles; both channels use Call 2; ordinary self-conditioning	Directed versus symmetric anticipation with contact supervision retained
No anticipation	Trunk and router trained; no anticipation	Value of the anticipation query and its response objective
Frozen trunk + our method	Initial trunk fixed; router trained	Value of adapting the generator within the combined method
Adaptation, 799 calls	Ordinary adaptation checkpoint; 799 base calls	Extra generator calls as an alternative use of inference computation
Fixed/shuffled followers	Full training run; changed inference roles; router enabled	Learned roles versus the tested policies at this checkpoint
No response loss	Separate training run; anticipation used at generation	Value of supervising the deployed response
Original-state auxiliary loss	Separate training run; auxiliary supervision at the original input	Response-input supervision beyond the additional original-state objective
Fixed direction, retrained	Trunk trained; ordinary self-conditioning retained	Remove learned roles and weights, not the self-conditioning mechanism

For this control, both channels share an Euler probe at the designed, occupied residues. With A from Appendix A.3,

$$s^{\text{probe}}_{t,m,i} = s_{t,m,i} + A_i \delta \, \text{sg}(v^{(0)}_{m,i}), \qquad m \in \{x, z\}. \tag{31}$$

Training uses $\delta = (1 - \bar{t})/400$, where $\bar{t}$ is the sampled training time, and stops gradients through the first velocity. Generation uses the live first velocity and the mean actual increment of the two modality schedules. Both inputs remain unchanged outside A, and molecular context is unchanged. If the first sampling step has no preceding increment, $\delta = 0$ but the second call still runs. Each of 400 sampling steps therefore uses two generator evaluations. Both calls use ordinary, unweighted self-conditioning h.

The Euler displacement $\delta v^{(0)}$ is not clipped. The endpoint-displacement clipping used for Full's latent lookahead is not applied to this Euler probe. Neither the second velocity nor the residual $v^{(1)} - v^{(0)}$ is clipped before replacement, and no geometry snap is applied.

The scalar response has the form $v^{(0)} + \alpha(v^{(1)} - v^{(0)})$, with $\alpha = 1$. Both channels therefore use the complete second velocity at every valid designed residue. Using the intervention mask A defined in Appendix A.3, the output is

$$v^{\text{out}}_{m,i} = \begin{cases} v^{(1)}_{m,i}, & A_i = 1, \\ v^{(0)}_{m,i}, & A_i = 0, \end{cases} \qquad m \in \{x, z\}. \tag{32}$$

Without follower assignments, response supervision uses all residues in A for each channel,

$$\mathcal{L}^{\text{no router}}_{\text{ant}} = \sum_{m \in \{x,z\}} \frac{\sum_i A_i \| v^{(1)}_{m,i} - u_{m,i} \|_2^2}{\max(1, \sum_i A_i)}. \tag{33}$$

The target $u_m = (s_{1,m} - s_{t,m})/(1 - t_m)$ is evaluated at the original state and supplies supervision only. The squared norm sums the three coordinate or eight latent feature errors before averaging over eligible residues. We sum the two channel means without averaging over feature dimensions. If there are no eligible residues, the corresponding term is zero. The response coefficient remains 1, the contact coefficient remains 2, and router objectives are absent.

Full also replaces the selected follower velocity completely, but retains the first velocity for its leader. The no-router control retains the same contact and anticipation loss coefficients and full response replacement while removing role selection and changing the probe. All reported no-router measurements use this setting, with 400 sampling steps and two generator calls per step.

The frozen-trunk control retains the routed anticipation mechanism and trains the router while holding the initial generator parameters fixed.

D.3 Fixed and shuffled roles at inference

These controls reuse the Full checkpoint and retain learned self-conditioning weights. One assigns structure leadership at every designed residue, making sequence the follower. Another assigns sequence leadership everywhere, making structure the follower. The shuffled control permutes roles among residues within distance strata. These comparisons test the learned residue assignments against a single global direction and against reassignment among residues at comparable partner distances. The initial role determines the probe and response selection; the second router evaluation still updates conditioning weights.

D.4 Auxiliary supervision and fixed-direction retraining

The no-response-loss control retains anticipation at inference but omits its supervised objective during a separate training run. The original-state auxiliary control instead applies the extra supervision to the first prediction at the original input, retaining the probe at inference. Together they distinguish the value of an additional supervised objective from the smaller contribution of supervising the response itself. The retrained fixed-direction control uses a global role assignment selected on validation data and trains the generator with that assignment. Generation uses the same assignment and ordinary self-conditioning. Unlike the no-router control, it retains a leader and follower and therefore retains a directional probe. It is also distinct from the inference-only fixed policies, which reuse Full with learned conditioning weights.

D.5 Local response measurements

Figure 4e compares Full, no response loss, and the retrained fixed-direction control on corrupted reference complexes. The cohorts contain 27 binder, 22 enzyme, and 19 antibody source proteins. The enzyme analysis additionally restricts to residues near the ligand within the same 22 source proteins. This regional subset overlaps the all-designed analysis and is not added to it.

For source protein p, let B_p, C_p, A_p be the mean benefit, cost, and shared reference error after averaging its states and references. The reported benefit is $100 \sum_p B_p / \sum_p A_p$, with the analogous expression for cost. The same source proteins and reference denominator are used for all variants. First and output errors satisfy $B_p - C_p = E_p^{\mathrm{first}} - E_p^{\mathrm{out}}$. Reporting these errors distinguishes a useful final response from a large change caused only by an inaccurate first prediction. Residue deterioration frequencies sum both channels before thresholding.

This measurement uses the learned snap response and a shared reference denominator. Figure 2 instead uses the frozen initial checkpoints, a uniform probe, and its own original-error normalization. Their percentages answer different questions and are not a direct before-and-after comparison.

D.6 Training-seed uncertainty

The reported intervals condition on the fitted checkpoint for each variant. They measure variation across targets or source proteins and do not estimate variability across independent training seeds. The documented enzyme training seed is 1401; a complete set of per-run seeds is not included in the available execution records.

E Additional Design and Ablation Results

E.1 Complete benchmark comparisons

Table 17 and Figure A1 report the complete benchmark under the common evaluation procedure in Appendix B. Each method contributes three candidates and five refolds per candidate, with one selected candidate and refold per target.

Frozen Complexa and ordinary full adaptation compare Co-DIRECT with the same generator before and after flow-matching adaptation. RFdiffusion and RFantibody followed by ProteinMPNN provide task-specific inverse-folding pipelines (Watson et al., 2023; Ahern et al., 2026; Butcher et al., 2025; Bennett et al., 2026; Dauparas et al., 2022). The evaluated BoltzGen configuration includes generation and inverse folding (Stärk et al., 2025a). Chroma and DiffAb provide additional generative comparisons (Ingraham et al., 2023; Luo et al., 2022). These groups describe the evaluated pipelines. Comparisons also reflect differences in models, training, and conditioning.

Table 17 reports means over valid observations for all evaluated methods and metrics. Figure A1 shows target-level variation behind the primary comparisons, including additional baselines omitted from the compact main figure. Ordinary-adaptation target differences appear in Figure 4a. Each paired estimate uses targets available for that comparison, so it can differ from subtracting table means. Missing evaluations are reported through coverage and are not assigned zero scores.

Full adaptation's binder ipTM of 0.3092 rises to 0.5471 with Co-DIRECT. The paired difference is +0.2379 ($[0.1545, 0.3154]$), with 26 improvements and 4 declines. Differences from RFdiffusion1 and RFdiffusion3 are +0.0974 ($[0.0101, 0.1836]$) and +0.0742 ($[0.0145, 0.1340]$), showing higher binder ipTM for Co-DIRECT than both RF pipelines. Against BoltzGen, the difference is -0.0036 ($[-0.0955, 0.0886]$), with 15 improvements and 14 declines, leaving this ipTM comparison unresolved. BoltzGen retains better mean ipAE, binder confidence, and scRMSD. Co-DIRECT passes the binder gates on 13/30 targets, compared with adaptation's 2/30 and BoltzGen's 15/29.

Co-DIRECT reduces enzyme ligand-fit RMSD from adaptation's 19.89 to 8.16 Å. On 40 matched targets, the reduction is 11.70 Å ($[9.36, 13.99]$), with 39 improvements and 1 decline. Against RFdiffusion2 with ProteinMPNN, the paired difference is -1.22 ($[-2.20, -0.37]$) Å on 36 targets, with 25 improvements and 11 declines. Mean motif-fit RMSD is 1.81 versus 2.53 Å. The RFdiffusion2 comparison covers 36 of 41 targets; five lack an evaluation. No Co-DIRECT enzyme target passes

Table 17: **Refold quality and generated–refold consistency.** Each target retains its selected candidate/refold from three candidates and five refolds each. Gray identifies Co-DIRECT. Bold marks the best mean, not statistical superiority. n is the valid count. Enzyme $n = a/b$ gives the counts for motif/ligand-fit RMSD and ligand-ipTM/scRMSD, respectively. Binder/antibody ipTM and ipAE assess the design–target interface. Binder pLDDT is design-chain confidence (0–100). scRMSD compares generated/refolded design-chain Cα coordinates. CDR-H3 RMSD uses framework alignment. Lower RMSD means greater consistency, not necessarily better task geometry. Cohorts contain 30/41/32 targets. Appendix B.7 defines these metrics. †RFantibody also searches docking poses.

Binder design

Method	ipTM ↑ (0–1)	ipAE ↓ (Å)	pLDDT ↑ (0–100)	scRMSD ↓ (Å)	n
Complexa frozen	0.241	24.44	38.19	17.148	30
Full adaptation	0.309	22.51	45.92	16.904	30
RFdiffusion1 + ProteinMPNN	0.450	17.51	76.14	10.904	30
RFdiffusion3 + ProteinMPNN	0.473	18.62	68.94	10.272	30
BoltzGen	**0.551**	**13.83**	**82.55**	**2.324**	29
Chroma	0.271	21.85	69.53	16.906	29
Co-DIRECT	0.547	14.04	78.15	4.608	30

Enzyme scaffolding

Method	Motif-fit RMSD ↓ (Å)	Ligand-fit RMSD ↓ (Å)	Ligand ipTM ↑ (0–1)	scRMSD ↓ (Å)	n
Complexa frozen	8.696	17.305	0.447	17.723	40/41
Full adaptation	9.972	19.891	0.414	18.021	40/41
RFdiffusion2 + ProteinMPNN	2.529	9.400	0.540	9.544	36
Co-DIRECT	**1.813**	**8.164**	**0.592**	**7.420**	41

Antibody CDR redesign

Method	ipTM ↑ (0–1)	ipAE ↓ (Å)	CDR-H3 RMSD ↓ (Å)	n
Complexa frozen	0.450	19.71	2.164	32
Full adaptation	0.474	19.10	2.115	32
RFantibody + ProteinMPNN†	0.550	14.38	1.944	32
DiffAb	**0.551**	14.75	**1.914**	32
Co-DIRECT	0.541	**13.32**	1.955	32

every gate: all 41 ligand-fit RMSDs exceed 3 Å, with a median of 7.33 Å and a minimum of 4.62 Å. Lower mean RMSDs therefore do not yet yield designs meeting all joint geometric and confidence criteria.

Mean antibody ipTM rises from adaptation's 0.4738 to 0.5412, with a paired difference of +0.0674 ($[-0.0125, 0.1498]$) and 18 improving versus 14 declining targets. The interval includes zero, as does the primary ipTM contrast with the 799-call control. Secondary ipAE decreases by 5.78 Å ($[3.75, 7.86]$) on 28/32 targets, and gate passes rise from 9/32 to 18/32. The ipAE comparison also favors Full over the 799-call control. RFantibody and DiffAb retain slightly higher mean ipTM, with paired intervals spanning zero. Co-DIRECT has lower mean ipAE than both, with only the RFantibody secondary interval excluding zero. Appendix E.6 reports target-level transitions.

Refolding receives sequence and ligand inputs rather than generated coordinates. Mean Cα scRMSD falls from adaptation's 16.90 to 4.61 Å for binders and from 18.02 to 7.42 Å for enzymes, improving 30/30 and 41/41 targets. Antibody framework-aligned CDR-H3 RMSD falls from 2.12 to 1.96 Å, with 16/32 improvements. BoltzGen retains lower binder scRMSD (2.32 Å). RFdiffusion2 has mean enzyme scRMSD 9.54 Å, and its paired contrast favors Co-DIRECT. Antibody-specialist structural contrasts span zero. Tables 21–23 report intervals and counts.

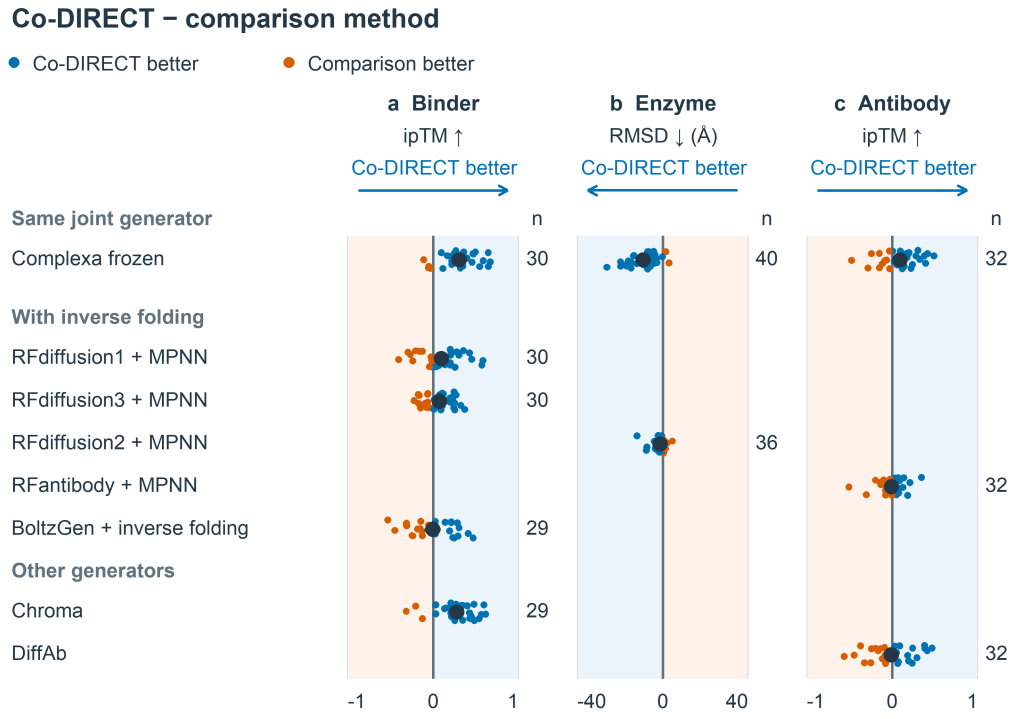


Figure A1: **Target-level variation in benchmark comparisons.** Differences are Co-DIRECT minus comparator. Positive interface ipTM for binders and antibodies and negative ligand-fit RMSD for enzymes favor Co-DIRECT, as indicated by the arrows. Each point represents one matched target. Blue favors Co-DIRECT, vermilion favors the comparator, and gray marks a tie. Dark summaries show paired means and marginal 95% target-bootstrap intervals, with the paired count n beside each comparison. Blank cells denote unevaluated combinations. Inverse-folding methods use ProteinMPNN except BoltzGen, which uses its own inverse-folding stage. Method groups describe comparison roles. Ordinary adaptation is shown in Figure 4a. Additional endpoints and metric-specific coverage appear in Table 17.

E.2 Complete paired effects and component comparisons

Tables 18–20 give the primary paired effects for external methods and all main ablations. Every difference is Co-DIRECT minus the comparator. Positive interface ipTM and negative ligand-fit RMSD differences favor Co-DIRECT. Counts report improved, tied, and worse targets, using the procedures in Appendix B.9.

Removing anticipation incurs mean losses of 0.0931 binder ipTM, 3.362 Å enzyme RMSD, and 0.0551 antibody ipTM. No-router losses are 0.1001 binder ipTM, 4.020 Å enzyme RMSD, and 0.0427 antibody ipTM. Both sets of intervals favor Full across all three families. Frozen-trunk losses are 0.0737 binder ipTM, 2.590 Å enzyme RMSD, and 0.0527 antibody ipTM; the enzyme and antibody intervals favor Full, while the binder interval includes zero. These comparisons support combining anticipation and routing, with task-dependent evidence for further generator adaptation. The no-router control changes role selection, probe construction, and self-conditioning together.

Increasing ordinary adaptation from 400 to 799 sampling calls changes binder ipTM from 0.3092 to 0.3180, enzyme ligand-fit RMSD from 19.89 to 19.40 Å, and antibody ipTM from 0.4738 to 0.4780. Full retains paired mean advantages of 0.2291 ipTM, 11.20 Å, and 0.0632 ipTM. Binder and enzyme intervals favor Full; the antibody ipTM interval includes zero. Its scRMSD advantages over the 799-call control are 11.019 Å for binders and 7.099 Å for enzymes, with intervals excluding zero. The antibody structural-consistency interval spans zero. Ordinary adaptation and its 799-call control

Table 18: Differences in interface ipTM on paired binder targets. Positive values favor Co-DIRECT. Intervals are marginal 95% target-bootstrap intervals. Counts are improved/tied/worse.

Comparator	n	Mean Δ	95% interval	Counts
Complexa frozen	30	0.3057	$[0.2311, 0.3812]$	26/0/4
Full adaptation	30	0.2379	$[0.1545, 0.3154]$	26/0/4
RFdiffusion1 + MPNN	30	0.0974	$[0.0101, 0.1836]$	20/0/10
RFdiffusion3 + MPNN	30	0.0742	$[0.0145, 0.1340]$	19/0/11
BoltzGen	29	-0.0036	$[-0.0955, 0.0886]$	15/0/14
Chroma	29	0.2763	$[0.1909, 0.3559]$	26/0/3
No router	30	0.1001	$[0.0801, 0.1199]$	28/0/2
No anticipation	30	0.0931	$[0.0669, 0.1186]$	29/0/1
Frozen trunk	30	0.0737	$[-0.0196, 0.1684]$	16/0/14
Adaptation, 799 calls	30	0.2291	$[0.1433, 0.3066]$	28/0/2
Fixed direction, retrained	30	0.0445	$[-0.0353, 0.1263]$	14/0/16
Always sequence follower	30	0.0144	$[-0.0368, 0.0595]$	21/0/9
Always backbone follower	30	0.0173	$[-0.0205, 0.0552]$	18/0/12
Shuffled roles	30	0.0483	$[-0.0322, 0.1248]$	20/0/10
No response loss	30	0.0543	$[-0.0356, 0.1423]$	19/0/11
Original-state auxiliary loss	30	0.0055	$[-0.0389, 0.0476]$	15/0/15

Table 19: Differences in ligand-fit RMSD on paired enzyme targets (Å). Negative values favor Co-DIRECT. Intervals are marginal 95% target-bootstrap intervals. Counts are improved/tied/worse.

Comparator	n	Mean Δ	95% interval	Counts
Complexa frozen	40	-9.11	$[-11.05, -7.25]$	38/0/2
Full adaptation	40	-11.70	$[-13.99, -9.36]$	39/0/1
RFdiffusion2 + MPNN	36	-1.22	$[-2.20, -0.37]$	25/0/11
No router	41	-4.02	$[-5.20, -2.83]$	40/0/1
No anticipation	41	-3.36	$[-3.74, -2.96]$	41/0/0
Frozen trunk	41	-2.59	$[-4.13, -0.96]$	30/0/11
Adaptation, 799 calls	40	-11.20	$[-13.55, -8.84]$	39/0/1
Fixed direction, retrained	41	-1.64	$[-3.12, 0.09]$	28/0/13
Always sequence follower	41	-0.33	$[-1.00, 0.34]$	25/0/16
Always backbone follower	41	-0.38	$[-0.94, 0.20]$	25/0/16
Shuffled roles	41	-1.73	$[-3.38, 0.09]$	25/0/16
No response loss	41	-1.99	$[-3.28, -0.70]$	27/0/14
Original-state auxiliary loss	41	-0.15	$[-0.88, 0.56]$	20/0/21

Table 20: Differences in interface ipTM on paired antibody targets. Positive values favor Co-DIRECT. Intervals are marginal 95% target-bootstrap intervals. Counts are improved/tied/worse.

Comparator	n	Mean Δ	95% interval	Counts
Complexa frozen	32	0.0909	$[0.0138, 0.1619]$	22/0/10
Full adaptation	32	0.0674	$[-0.0125, 0.1498]$	18/0/14
RFantibody + MPNN	32	-0.0085	$[-0.0633, 0.0417]$	16/0/16
DiffAb	32	-0.0097	$[-0.0967, 0.0769]$	14/0/18
No router	32	0.0427	$[0.0128, 0.0718]$	28/0/4
No anticipation	32	0.0551	$[0.0375, 0.0735]$	28/0/4
Frozen trunk	32	0.0527	$[0.0036, 0.0976]$	22/0/10
Adaptation, 799 calls	32	0.0632	$[-0.0176, 0.1464]$	17/0/15
Fixed direction, retrained	32	0.0352	$[-0.0163, 0.0829]$	23/0/9
Always sequence follower	32	0.0213	$[-0.0309, 0.0695]$	22/0/10
Always backbone follower	32	0.0236	$[-0.0322, 0.0773]$	20/0/12
Shuffled roles	32	0.0379	$[-0.0162, 0.0886]$	21/0/11
No response loss	32	0.0419	$[-0.0172, 0.0988]$	22/0/10
Original-state auxiliary loss	32	0.0092	$[-0.0477, 0.0606]$	19/0/13

have distinct mean scRMSDs: 16.904 versus 15.627 Å for binders and 18.021 versus 14.519 Å for enzymes.

E.3 Direction policies and auxiliary supervision

The following comparisons supplement Figure 4. Component retraining changes the fitted model. Inference-only policies reuse Full's checkpoint. Appendix D defines the interventions, and Tables 18–20 report all primary paired estimates.

Fixed and shuffled inference roles. At the Full checkpoint, structure leadership incurs mean losses of 0.0144 binder ipTM, 0.328 Å enzyme RMSD, and 0.0213 antibody ipTM. Sequence leadership incurs 0.0173 binder ipTM, 0.384 Å enzyme RMSD, and 0.0236 antibody ipTM, and shuffling roles incurs 0.0483 binder ipTM, 1.732 Å enzyme RMSD, and 0.0379 antibody ipTM. Every corresponding primary-endpoint interval includes zero. Conditioning remains enabled, but changed roles alter probes and subsequent trajectories. Mean trends alone do not establish a primary design-quality advantage for learned residue assignments.

Auxiliary supervision and its location. Without follower-response supervision, mean losses are 0.0543 binder ipTM, 1.987 Å enzyme RMSD, and 0.0419 antibody ipTM. Only the enzyme primary-endpoint interval excludes zero. Moving the auxiliary objective to the first prediction gives losses of 0.0055 binder ipTM, 0.148 Å enzyme RMSD, and 0.0092 antibody ipTM. Both controls retain Probe at generation, and all Full versus original-state auxiliary primary intervals include zero. These endpoints leave the value of response-specific loss placement unresolved, while the reference-state measurements examine local prediction effects.

E.4 Generated and refolded backbone agreement

Tables 21–23 report paired structural-consistency differences for the same selected designs. Negative values favor Co-DIRECT. Mean binder and enzyme scRMSDs are 4.608 and 7.420 Å, with medians 4.181 and 5.045 Å. Co-DIRECT improves 30/30 binder, 41/41 enzyme, and 16/32 antibody targets over ordinary adaptation. The antibody improvement in mean error reflects the magnitudes of changes despite equal counts of improvements and declines. BoltzGen retains lower binder scRMSD. The enzyme RFdiffusion2 contrast favors Co-DIRECT, while antibody-specialist intervals span zero. Numerical ties indicate equal reported scalar errors within tolerance and do not establish identical coordinates.

Table 21: Binder design-chain scRMSD differences between Co-DIRECT and comparators on paired targets in Å. Negative differences favor Co-DIRECT. Counts are improved/tied/worse. Intervals are nominal 95% target-bootstrap intervals.

Comparator	n	Mean Δ	95% interval	Counts
Complexa frozen	30	-12.540	$[-15.151, -10.372]$	30/0/0
Full adaptation	30	-12.297	$[-14.622, -10.254]$	30/0/0
RFdiffusion1 + MPNN	30	-6.297	$[-9.417, -3.034]$	22/0/8
RFdiffusion3 + MPNN	30	-5.664	$[-7.874, -3.354]$	25/0/5
BoltzGen	29	2.398	$[0.203, 4.290]$	4/0/25
Chroma	29	-12.184	$[-14.399, -9.773]$	28/0/1
No router	30	-3.532	$[-4.091, -2.950]$	30/0/0
No anticipation	30	-3.112	$[-3.993, -2.187]$	28/0/2
Frozen trunk	30	-2.537	$[-4.193, -0.972]$	21/0/9
Adaptation, 799 calls	30	-11.019	$[-13.227, -8.702]$	28/0/2
Fixed direction, retrained	30	-2.033	$[-3.852, -0.291]$	17/0/13
Always sequence follower	30	-0.980	$[-2.790, 0.823]$	17/0/13
Always backbone follower	30	-1.109	$[-3.162, 0.868]$	17/0/13
Shuffled roles	30	-2.212	$[-4.185, -0.206]$	17/0/13
No response loss	30	-2.572	$[-4.633, -0.525]$	19/0/11
Original-state auxiliary loss	30	-0.113	$[-0.680, 0.434]$	15/0/15

Table 22: Enzyme design-chain scRMSD differences between Co-DIRECT and comparators on paired targets in Å. Negative differences favor Co-DIRECT. Counts are improved/tied/worse. Intervals are nominal 95% target-bootstrap intervals.

Comparator	n	Mean Δ	95% interval	Counts
Complexa frozen	41	-10.303	$[-11.764, -8.848]$	41/0/0
Full adaptation	41	-10.601	$[-11.973, -9.191]$	41/0/0
RFdiffusion2 + MPNN	36	-2.117	$[-3.947, -0.389]$	19/0/17
No router	41	-4.940	$[-5.675, -4.221]$	40/0/1
No anticipation	41	-4.290	$[-4.986, -3.561]$	39/0/2
Frozen trunk	41	-3.460	$[-5.491, -1.354]$	27/0/14
Adaptation, 799 calls	41	-7.099	$[-9.226, -5.008]$	34/0/7
Fixed direction, retrained	41	-2.550	$[-4.469, -0.589]$	28/0/13
Always sequence follower	41	-1.739	$[-3.821, 0.339]$	26/0/15
Always backbone follower	41	-1.821	$[-4.028, 0.295]$	23/0/18
Shuffled roles	41	-2.662	$[-4.730, -0.568]$	26/0/15
No response loss	41	-3.032	$[-4.776, -1.269]$	29/0/12
Original-state auxiliary loss	41	-0.163	$[-1.012, 0.683]$	20/0/21

Table 23: Antibody framework-aligned CDR-H3 RMSD differences between Co-DIRECT and comparators on paired targets in Å. Negative differences favor Co-DIRECT. Counts are improved/tied/worse. Intervals are nominal 95% target-bootstrap intervals.

Comparator	n	Mean Δ	95% interval	Counts
Complexa frozen	32	-0.209	$[-0.391, -0.027]$	21/0/11
Full adaptation	32	-0.160	$[-0.326, -0.006]$	16/0/16
RFantibody + MPNN	32	0.011	$[-0.106, 0.122]$	16/0/16
DiffAb	32	0.041	$[-0.143, 0.222]$	15/0/17
No router	32	-0.009	$[-0.072, 0.055]$	15/0/17
No anticipation	32	-0.109	$[-0.196, -0.030]$	17/0/15
Frozen trunk	32	-0.082	$[-0.148, -0.021]$	21/0/11
Adaptation, 799 calls	32	-0.152	$[-0.320, 0.001]$	20/0/12
Fixed direction, retrained	32	-0.016	$[-0.084, 0.043]$	14/0/18
Always sequence follower	32	0.021	$[-0.033, 0.077]$	15/0/17
Always backbone follower	32	-0.029	$[-0.081, 0.019]$	16/0/16
Shuffled roles	32	-0.042	$[-0.131, 0.030]$	17/0/15
No response loss	32	-0.058	$[-0.130, 0.011]$	18/0/14
Original-state auxiliary loss	32	0.001	$[-0.089, 0.080]$	17/0/15

E.5 Threshold pass rates and coverage

Table 24 reports targets passing all family-specific thresholds. Missing required metrics fail the applicable gate, while cases without an evaluated design are counted separately. Threshold passes summarize computational endpoints and do not constitute experimental binding or catalysis.

All 41 Co-DIRECT enzyme targets fail the ligand-fit RMSD threshold. RFdiffusion2 has both fit RMSDs on 36 evaluated targets, each with ligand-fit RMSD above 3 Å; the other five targets were not evaluated. For the frozen, ordinary-adaptation, and 799-call controls, motif correspondence fails on the same target, `M0630_1j79`. Their unavailable fit RMSDs fail the required geometry gates under the protocol. This is a correspondence failure rather than a measured threshold exceedance. These gate outcomes do not require an inference about the unreported interface PAE or pLDDT.

E.6 Complementary antibody endpoints

Interface ipTM remains the primary antibody endpoint in Figure 3. The following exploratory analysis examines ipAE and changes in whether targets pass the evaluation thresholds on the same 32 targets to characterize complementary endpoints and target-level variation beyond mean ipTM. Each method's scores come from its already selected candidate and refold. The analysis does not reselect

Table 24: Targets passing all evaluation thresholds and targets not evaluated in the full benchmark.

Family	Method	Pass / evaluated	Unevaluated
Binder	Complexa frozen	0/30	0
Binder	Full adaptation	2/30	0
Binder	RFdiffusion1 + MPNN	9/30	0
Binder	RFdiffusion3 + MPNN	7/30	0
Binder	BoltzGen	15/29	1
Binder	Chroma	2/29	1
Binder	Co-DIRECT	13/30	0
Enzyme	Complexa frozen	0/41	0
Enzyme	Full adaptation	0/41	0
Enzyme	RFdiffusion2 + MPNN	0/36	5
Enzyme	Co-DIRECT	0/41	0
Antibody	Complexa frozen	7/32	0
Antibody	Full adaptation	9/32	0
Antibody	RFantibody + MPNN	16/32	0
Antibody	DiffAb	16/32	0
Antibody	Co-DIRECT	18/32	0

designs to favor ipAE. Matching uses the same run, target, metric scope, protocol, and selection-rule checks as Appendix B.9.

Table 25: Antibody ipAE differences, Co-DIRECT minus comparator, in Å. Negative values favor Co-DIRECT. Counts are improved/tied/worse out of 32 paired targets. Intervals are marginal 95% target-bootstrap intervals.

Comparator	Mean Δ	95% interval	Counts
Complexa frozen	-6.39	$[-8.19, -4.55]$	28/0/4
Full adaptation	-5.78	$[-7.86, -3.75]$	28/0/4
Adaptation, 799 calls	-5.53	$[-7.74, -3.53]$	28/0/4
RFantibody + MPNN	-1.06	$[-1.91, -0.22]$	17/0/15
DiffAb	-1.43	$[-3.40, 0.52]$	16/0/16

Negative ipAE differences favor Co-DIRECT. Intervals use the paired target-bootstrap procedure in Appendix B.9 and are reported as secondary comparisons without multiplicity adjustment.

Against full adaptation, 18 targets improve in both ipTM and ipAE, 10 improve only in ipAE, 0 improve only in ipTM, and 4 decline in both. Against the 799-call control, these counts are 17, 11, 0, and 4. There are no numerical ties. The secondary endpoints retain the same selected designs and do not change the primary endpoint. Their more consistent ipAE gains complement the uncertain primary ipTM contrast.

Table 26: Paired antibody gate transitions. Retained indicates that both methods pass. Gained means only Co-DIRECT passes, while lost means only the comparator passes. Neither indicates that both fail. Each row sums to 32. These are recorded computational gates, not experimentally validated successes.

Comparator	Retained	Gained	Lost	Neither
Complexa frozen	5	13	2	12
Full adaptation	7	11	2	12
Adaptation, 799 calls	7	11	2	12
RFantibody + MPNN	14	4	2	12
DiffAb	12	6	4	10

Co-DIRECT increases antibody gate coverage from 28.1% for full adaptation to 56.25%, a 28.125 percentage-point increase, with 11 gained and 2 lost targets. Relative to RFantibody, 14 targets pass for both, 4 pass only for Co-DIRECT, and 2 pass only for RFantibody. Relative to DiffAb, these counts are 12, 6, and 4. Co-DIRECT passes 18 targets, compared with 16 for each specialist, with

differences in their covered sets. Gate coverage and mean interface scores describe complementary aspects of the benchmark.

Co-DIRECT has mean pLDDT 90.84 over redesigned CDR residues. Baseline confidence values average over H+L or the whole complex and therefore measure different regions (Table 27). These broader averages include fixed framework residues and are reported separately.

Table 27: Antibody baseline pLDDT by measured region (0–100). H+L and complex-wide averages are not CDR-only confidence and are not compared to Co-DIRECT's CDR pLDDT of 90.84.

Method	H+L pLDDT	Complex pLDDT
Complexa frozen	89.15	86.80
Full adaptation	89.29	86.65
RFantibody	88.93	85.12
DiffAb	89.57	85.62

E.7 LOCAL RESPONSE BENEFITS, COSTS, AND OUTPUT ERROR

Table 28 extends the enzyme ligand-region analysis in Figure 4e to all designed residues in each family. All percentages use the shared reference denominator defined in Appendix D.5. Full produces larger benefits, smaller costs, and lower output error than either control in each scope.

Table 28: Local prediction benefits and costs. Benefit, cost, and first/output error are percentages of a shared mean reference error on the same source proteins. "Near" denotes residues near the ligand. All other rows use all designed residues.

Scope	Method	Benefit	Cost	First	Output
Binder	Full	30.49	10.10	96.66	76.27
Binder	No response loss	16.08	17.92	96.10	97.95
Binder	Fixed, retrained	21.11	15.59	94.65	89.13
Enzyme	Full	34.36	10.53	96.72	72.89
Enzyme	No response loss	15.93	17.34	93.75	95.17
Enzyme	Fixed, retrained	19.54	15.78	92.70	88.94
Enzyme, near	Full	36.63	7.96	97.72	69.05
Enzyme, near	No response loss	16.35	17.66	93.68	94.99
Enzyme, near	Fixed, retrained	20.50	15.10	95.17	89.77
Antibody	Full	35.77	11.67	97.82	73.72
Antibody	No response loss	16.92	17.56	95.69	96.33
Antibody	Fixed, retrained	19.97	15.64	93.90	89.56

Near enzyme ligands, Full has benefit 36.63% and cost 7.96%, compared with 16.35% and 17.66% without response supervision. Output error falls from 94.99% to 69.05% despite a higher first-prediction error for Full. The mean fraction of designed residues with increased error falls from 32.46% to 19.54%. The retrained fixed-direction control has benefit 20.50%, cost 15.10%, and output error 89.77%. These comparisons show improvement in the deployed response rather than only a larger change from the first prediction.

The antibody diagnostic gives the same qualitative pattern on 19 source proteins. Full has benefit 35.77%, cost 11.67%, and output error 73.72%, compared with output error 96.33% without response supervision. Deterioration frequency decreases from 31.10% to 18.85%. These reference-state results complement the 32-target generation benchmark, where mean ipTM improves over ordinary adaptation with heterogeneous target-level effects. Source proteins, residues, corruption draws, candidates, and refolds are kept distinct in aggregation; the diagnostic summaries are reported as point estimates.