ONLINE SEARCH-AUGMENTED REINFORCEMENT POST-TRAINING FOR ALL-ATOM PROTEIN DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Improving pretrained all-atom protein generators toward downstream biophysical objectives remains challenging but is underexplored. Reinforcement-based post-training can directly adapt the generator, but trajectory-level optimization over hundreds of denoising steps is expensive. Inference-time search can instead improve candidate quality through additional sampling and selection, but leaves the generator itself unchanged and must be repeated whenever new candidates are generated. We introduce Search-Augmented Reinforcement Post-Training (SARP), which combines trajectory-free endpoint-based reinforcement with temporary supervision on high-reward endpoints discovered by reward-guided search. Across small-molecule binder and enzyme-design tasks, SARP improves design quality under standard sampling of the partially-latent protein all-atom generator. In small-molecule binder design, SARP decreases binding energy by up to 14.6 REU and RF3 min-iPAE by up to 2.2 Å relative to the pretrained model. Across five enzyme targets, SARP produces 42.2 unique successes per 100 designs, compared with 11.2 for the pretrained model and 35.6 for the RL-only method DGPO. These gains persist under independent AlphaFold3 cross-evaluation, suggesting that the improvements transfer across structure predictors rather than arising solely from optimization of RF3-specific signals. Gradient analysis further shows that search supervision increasingly conflicts with reinforcement over training and that the backbone and latent flows experience different optimization pressures, suggesting that adaptive control of supervision and regularization may be beneficial. Overall, SARP provides a practical post-training framework for protein generative models that enables fast standard-sampling generation.

1 INTRODUCTION

Designing de novo proteins that satisfy functional and structural constraints remains a central challenge in protein engineering (Yang et al., 2026). Recent all-atom generative models, including RFDiffusion3 (Butcher et al., 2025), BoltzGen (Stark et al., 2025), and Proteína-Complexa (Didi et al., 2026), can generate realistic protein structures and molecular complexes, yet designs with favorable binding, confident refolding, or precise active-site geometry remain rare under standard sampling. As a result, practical pipelines still rely heavily on generate-and-filter workflows that score large candidate pools using structural, confidence-based, or energy-based metrics (Watson et al., 2023; Butcher et al., 2025; Stark et al., 2025). Improving the yield of high-quality samples directly from the generator is therefore important, but methods for post-training pretrained protein design models toward downstream biophysical objectives remain comparatively underexplored.

Reinforcement learning (RL) provides a natural mechanism for shifting generative models toward task-specific objectives and has been widely applied to protein sequence generation for fitness and functional optimization (Subramanian et al., 2024; Wang et al., 2025; Xue et al., 2025a). In contrast, online post-training of 3D protein structural diffusion and flow models remains relatively underexplored, with PocketX (Fan et al., 2025b) as one recent example. Group-relative methods such as GRPO are attractive for conditional protein design because they compare samples under the same condition without requiring a separate value model (Shao et al., 2024). Recent work extends group-relative optimization to diffusion and flow generators (Liu et al., 2026; Xue et al., 2025b) by treating denoising as a sequential decision process (McAllister et al., 2026), but trajectory-based optimization like GRPO is costly for long-horizon all-atom generation (e.g. Proteína-Complexa uses a 400

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

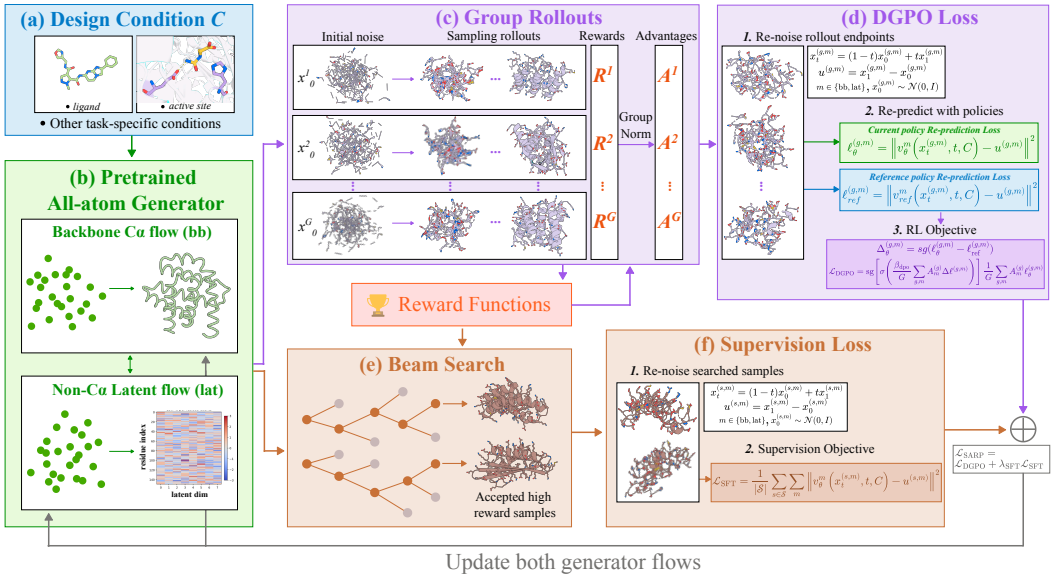

Figure 1: Overview of Search-Augmented Reinforcement Post-Training (SARP). On-policy rollouts are scored to form group-relative DGPO updates and reward-guided search provides additional high-reward endpoints for temporary supervised flow matching. Both objectives update the coupled backbone and latent flows.

denoising step by default) and may provide weak learning signals when on-policy groups contain few high-quality samples.

Inference-time search provides a complementary way to obtain high-reward designs. For example, Proteína-Complexa uses beam search to expand stochastic denoising continuations, score completed candidates with downstream metrics, and retain the most promising branches (Didi et al., 2026). This can substantially improve the quality of the final candidate set compared with standard independent sampling, especially on hard targets. However, search acts only at inference time and leaves the underlying generator unchanged. The structures it discovers therefore remain one-off outputs rather than improvements to the model itself, so similar benefits require repeating the same search procedure and paying the associated computational cost for future generations.

These limitations suggest an endpoint-centric way to combine reinforcement and search. Completed structures provide a common interface for both objectives: on-policy rollout endpoints support trajectory-free group-relative reinforcement, avoiding backpropagation through the original long sampling trajectory, and high-reward structures discovered by search can serve as additional direct flow-matching targets. This allows improvements found through search to be distilled into the generator rather than remaining one-off inference-time gains. Reusing reward-selected samples as supervision is closely related to established policy-improvement and self-training approaches (Silver et al., 2017; Anthony et al., 2017; Gulcehre et al., 2023; Dong et al., 2023).

Our contributions. We introduce ***Search-Augmented Reinforcement Post-Training (SARP)***, an endpoint-centric approach for post-training long-horizon all-atom protein generators (Figure 1). More specifically, we include (1) Trajectory-free reinforcement for coupled all-atom flows. We formulate endpoint-based DGPO (Luo et al., 2025) for a Proteína-Complexa–style generator with coupled explicit C_α and non-C_α latent flows, avoiding full-trajectory storage and per-step importance ratios over the 400-step sampler. (2) Search-augmented endpoint supervision with controlled source and schedule. We jointly optimize DGPO with temporary supervision on reward-guided search endpoints, isolating the mechanism against two controls: a dose-matched variant that distills the best on-policy endpoints, and DGPO with a doubled rollout group. (3) Optimization analysis of this protein post-training. We show that search supervision and on-policy reinforcement pull in increasingly different directions as training proceeds, and that the two coupled flows receive asymmetric optimization pressure. These dynamics explain why the strength and duration of supervision matter, and motivate adaptive rather than fixed schedules for future studies.

Across two protein-design settings, SARP substantially improves design quality under standard sampling for the partially latent protein design model. In small-molecule binder design, SARP improves

binding energy by up to 14.6 REU and RF3 min-iPAE by up to 2.2 Å. For five hard single-target enzyme design optimizations, SARP yields 42.2 unique successful clusters per 100 designs, compared with 11.2 for the pretrained model and 35.6 for DGPO. Gradient analysis further shows that reinforcement and endpoint supervision play distinct optimization roles across the coupled backbone and latent flows, motivating more adaptive control of post-training.

Related Work. We briefly review the most closely related work here and provide a more detailed discussion in Appendix A. Generative protein design has progressed from backbone generation (Watson et al., 2023; Ahern et al., 2026) to all-atom modeling of protein structures and molecular interactions (Butcher et al., 2025; Stark et al., 2025; Didi et al., 2026). Post-training for protein generation has primarily focused on discrete sequence and inverse-folding models (Subramanian et al., 2024; Wang et al., 2025; Xue et al., 2025a). Recent work has extended post-training to diffusion and flow generators through group-relative optimization (Xue et al., 2025b; Liu et al., 2026; Luo et al., 2025) and reward-weighted flow matching that reuses high-reward on-policy endpoints (Fan et al., 2025a). In protein structure generation, PocketX provides one recent example of ligand-conditioned preference optimization (Fan et al., 2025b). In parallel, inference-time search can improve generation quality through additional sampling and selection (Didi et al., 2026), and related work in game playing and language-model alignment has explored distilling search- or reward-selected samples back into the policy (Silver et al., 2017; Anthony et al., 2017; Gulcehre et al., 2023; Dong et al., 2023; Sessa et al., 2025).

2 PRELIMINARIES

Flow matching. Flow Matching (FM) learns a time-dependent velocity field that transports samples from a simple prior to the data distribution (Lipman et al., 2022; Liu et al., 2022). Given a clean sample $\mathbf{x}_1 \sim q(\mathbf{x}_1)$ and Gaussian noise $\mathbf{x}_0 \sim \mathcal{N}(0, I)$, we use the linear path $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$ for $t \in [0,1]$, with target velocity $\mathbf{u} = \mathbf{x}_1 - \mathbf{x}_0$. The velocity model is trained by $\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_1}\left[\|\mathbf{v}_\theta(\mathbf{x}_t,t) - \mathbf{u}\|^2\right]$. Generation starts from $\mathbf{x}_0$ and integrates the learned velocity field from $t=0$ to $t=1$. Latent flow matching applies the same construction to an autoencoder latent $\mathbf{z}_1 = E(\mathbf{x}_1)$, which is decoded back to the data space after generation (Dao et al., 2023).

Coupled explicit–latent protein generation. We apply the post-training framework to the latest Proteína-Complexa-style base model (Didi et al., 2026), which represents a protein of length L as $\mathbf{x} = (\mathbf{x}^{\mathrm{bb}}, \mathbf{x}^{\mathrm{lat}})$, where $\mathbf{x}^{\mathrm{bb}} \in \mathbb{R}^{L\times 3}$ contains explicit C_α coordinates and $\mathbf{x}^{\mathrm{lat}} \in \mathbb{R}^{L\times d_z}$ contains per-residue latents encoding residue identity and the remaining non-C_α atomic geometry. A frozen decoder maps the coupled representation to a full all-atom protein. For modality $m \in \{\mathrm{bb}, \mathrm{lat}\}$, $\mathbf{x}_0^m$ denotes Gaussian noise, $\mathbf{x}_1^m$ the clean endpoint, $\mathbf{x}_t^m = (1-t)\mathbf{x}_0^m + t\mathbf{x}_1^m$ the intermediate state, and $\mathbf{u}^m = \mathbf{x}_1^m - \mathbf{x}_0^m$ the target velocity. A shared network jointly predicts $\mathbf{v}_\theta^{\mathrm{bb}}$ and $\mathbf{v}_\theta^{\mathrm{lat}}$ from the coupled state, time t, and task condition C. Additional details are provided in Appendix C.

Group-relative optimization. Given a condition C, the policy generates a group of rollouts $\{\mathbf{x}_1^{(g)}\}_{g=1}^G \sim \pi_\theta(\cdot \mid C)$, where g indexes rollout-group members. Each sample receives reward $R^{(g)} = R(\mathbf{x}_1^{(g)}, C)$. Group-relative methods normalize rewards within the group (Shao et al., 2024): $A^{(g)} = \frac{R^{(g)} - \mu_R}{\sigma_R + \epsilon}, \qquad \mu_R = \frac{1}{G}\sum_{j=1}^G R^{(j)}$, where σ_R is the within-group reward standard deviation and ϵ is a small numerical-stability constant. This provides a condition-specific preference signal and avoids direct comparison across design targets with different reward scales.

3 METHOD

We introduce **Search-Augmented Reinforcement Post-Training** (SARP), an endpoint-centric framework for post-training the coupled backbone and latent flows for all-atom protein design. Given a design condition $\mathcal{C}$, SARP samples an on-policy rollout group, forms group-relative advantages from task rewards, and applies Direct Group Preference Optimization (DGPO) (Luo et al., 2025) to freshly re-noised rollout endpoints relative to a moving reference policy. Selected high-reward endpoints are additionally used as direct flow-matching targets. SARP constructs these high-reward targets through reward-guided beam search, and we study a matched rollout-derived variant that instead reuses high-reward endpoints from the on-policy group. Search-derived supervision is

active only during an initial phase, after which optimization continues with on-policy DGPO alone. Figure 1 summarizes the framework, with details in Appendix G.

3.1 Endpoint-Based Reinforcement Learning

We adapt DGPO (Luo et al., 2025) to the coupled explicit–latent protein representation by treating each final generated structure as an endpoint preference target. The endpoint is freshly re-noised to an intermediate flow-matching state, and the current policy is compared with a reference policy according to their denoising losses on that state. Consequently, the update does not require the original rollout trajectory, per-step importance ratios, or backpropagation through the sampling procedure.

3.1.1 Stochastic Group Rollouts

At each optimization step, we sample a design condition $\mathcal{C}$ and generate a group of G structures: $\left\{\mathbf{x}_1^{(g)}\right\}_{g=1}^{G} \sim \pi_\theta(\cdot \mid \mathcal{C})$. All group members share the same condition but are initialized from independent noise and sampled with the stochastic Proteína-Complexa sampler, which yields better post-training performance than deterministic ODE rollouts in our experiments.

3.1.2 Reward and Group-Advantage Construction

Each structure is evaluated using task-specific rewards and shared structural safeguards. Full definitions and thresholds are given in Section 4 and Appendix E.

Let $r_k^{(g)}$ denote reward component k for rollout g. We first standardize each component within the ordinary rollout group:

$$\widehat{z}_k^{(g)} = \frac{r_k^{(g)} - \mu_k}{\sigma_k + \epsilon}, \qquad \mu_k = \frac{1}{G}\sum_{j=1}^{G} r_k^{(j)}. \tag{1}$$

The normalized rewards are combined and the scale-normalized structural-validity penalty $p^{(g)}$ is subtracted. We then standardize the combined score within the rollout group, clip extreme values for numerical stability, and re-center the clipped advantages to preserve the zero-mean group-relative weighting:

$$\widetilde{S}^{(g)} = \sum_k w_k \widehat{z}_k^{(g)} - \frac{p^{(g)}}{\sigma_p + \epsilon}. \qquad A^{(g)} = \mathrm{recenter}\left[\mathrm{clip}\left(\frac{\widetilde{S}^{(g)} - \mu_{\widetilde{S}}}{\sigma_{\widetilde{S}} + \epsilon}, -A_{\max}, A_{\max}\right)\right]. \tag{2}$$

The first normalization balances heterogeneous reward scales and the second produces a local preference signal within the rollout group, following DanceGRPO (Xue et al., 2025b). Although the implementation supports distinct modality-specific advantages, our rewards are defined on the jointly decoded all-atom structure, so we use $A_{\mathrm{bb}}^{(g)} = A_{\mathrm{lat}}^{(g)} = A^{(g)}$ in all reported experiments.

3.1.3 DGPO for Coupled Protein Flows

For rollout g and modality $m \in \{\mathrm{bb}, \mathrm{lat}\}$, let $\mathbf{x}_1^{(g,m)}$ denote the generated endpoint. We sample $t \sim \mathcal{U}[0.1, 1.0]$ and construct a fresh prior state $\mathbf{x}_0^{(m)}$ in each modality space. For a fixed time and modality, the same prior state is shared across rollout-group members:

$$\mathbf{x}_t^{(g,m)} = (1-t)\mathbf{x}_0^{(m)} + t\mathbf{x}_1^{(g,m)}, \qquad \mathbf{u}^{(g,m)} = \mathbf{x}_1^{(g,m)} - \mathbf{x}_0^{(m)}. \tag{3}$$

We denote the complete coupled state by $\mathbf{x}_t^{(g)} = (\mathbf{x}_t^{(g,\mathrm{bb})}, \mathbf{x}_t^{(g,\mathrm{lat})})$. Sharing the fresh corruption across group members reduces variation caused by re-noising and provides a more controlled comparison among rollout endpoints.

For each modality, the masked online and reference denoising losses are

$$\ell_\theta^{(g,m)} = \left\| \mathbf{v}_\theta^m\left(\mathbf{x}_t^{(g)}, t, \mathcal{C}\right) - \mathbf{u}^{(g,m)} \right\|_{\mathrm{mask},m}^2, \qquad \ell_{\mathrm{ref}}^{(g,m)} = \left\| \mathbf{v}_{\theta_{\mathrm{ref}}}^m\left(\mathbf{x}_t^{(g)}, t, \mathcal{C}\right) - \mathbf{u}^{(g,m)} \right\|_{\mathrm{mask},m}^2. \tag{4}$$

The two modalities use different masks: the backbone loss is evaluated over generated, non-padded residues, whereas the latent loss is restricted to positions with valid latent-encoder representations. This excludes ligand and other non-protein atoms, as well as any protein positions without a valid latent encoding. We then aggregate the detached, advantage-weighted online–reference loss gaps over all rollout-group members and both modalities to construct a single bounded group-level modulation weight:

$$w_{\mathrm{DGPO}} = \sigma\left(\frac{\beta_{\mathrm{dpo}}}{G}\sum_{g=1}^{G}\sum_{m\in\{\mathrm{bb},\mathrm{lat}\}} A_m^{(g)}\,\mathrm{sg}\left[\ell_\theta^{(g,m)} - \ell_{\mathrm{ref}}^{(g,m)}\right]\right). \tag{5}$$

Here, $\sigma(\cdot)$ is the sigmoid function and $\mathrm{sg}(\cdot)$ denotes stop-gradient. The sigmoid produces one scalar weight over all samples and both modalities; no explicit clipping is applied to the denoising gaps.

The DGPO objective is

$$\mathcal{L}_{\mathrm{DGPO}} = \mathrm{sg}\left(w_{\mathrm{DGPO}}\right)\frac{1}{G}\sum_{g=1}^{G}\sum_{m\in\{\mathrm{bb},\mathrm{lat}\}} A_m^{(g)}\ell_\theta^{(g,m)}. \tag{6}$$

Intuitively, the objective favors lower denoising loss on positive-advantage endpoints and higher loss on negative-advantage endpoints. Repeating this comparison across freshly re-noised flow states shifts the policy toward endpoints that receive higher reward under the same design condition. The objective is averaged over multiple sampled flow times, with the remaining implementation details provided in Appendix G.3.

3.1.4 Reference-Policy Regularization with EMA

In addition to the DGPO preference update, we regularize the online velocity field toward a slowly moving reference policy:

$$\mathcal{L}_{\mathrm{ref}} = \frac{1}{G}\sum_{g=1}^{G}\sum_{m\in\{\mathrm{bb},\mathrm{lat}\}}\left\|v_\theta^m\left(\mathbf{x}_t^{(g)},t,C\right) - \mathrm{sg}\left[v_{\theta_{\mathrm{ref}}}^m\left(\mathbf{x}_t^{(g)},t,C\right)\right]\right\|_{\mathrm{mask},m}^2. \tag{7}$$

This reference-field consistency loss discourages abrupt changes to the denoising dynamics while allowing the preference objective to improve rewarded endpoints. It plays an anchoring role analogous to reference-policy regularization in reinforcement and preference optimization, but it is not an explicitly evaluated distributional KL divergence.

The reference policy is initialized from the pretrained generator. After an initial warm-up period, it is re-anchored to the current online policy and subsequently updated as an exponential moving average (EMA) of the online parameters. The resulting slowly lagging reference provides a stable local anchor while tracking recent policy improvement. More details are provided in Appendix G.4.

3.2 Reward-Selected Endpoint Supervision

In addition to relative reinforcement on rollout endpoints, SARP directly supervises selected high-reward endpoints obtained through reward-guided beam search. These searched endpoints are excluded from the DGPO rollout group because search-induced selection makes them off-policy with respect to the ordinary rollout distribution.

For the search-derived targets used in the main SARP formulation, a searched endpoint $x_1^{(s)}$ is accepted for supervision only if its search score is at least as high as the best score in the corresponding ordinary rollout group:

$$S_{\mathrm{search}}\left(x_1^{(s)}\right) \geq \max_{g\in\{1,\ldots,G\}} S_{\mathrm{search}}\left(x_1^{(g)}\right). \tag{8}$$

where S_{search} represents the search score, which is more explained in Appendix G.5. Thus, search supervision is applied only to candidates that match or exceed the strongest ordinary rollout under the same search score. If no searched endpoint satisfies this criterion, the supervision term is omitted. For SARP-Rollout, we instead construct the supervision set directly from the on-policy rollout

group by selecting the high-reward endpoint(s) under the same task-specific score. These selected endpoints remain part of the DGPO rollout group and are additionally reused as direct flow-matching targets, without introducing any extra search or generation. More details about how we construct a fair comparison can be found in Appendix G.6.

Let $\mathcal{S}$ denote the set of accepted endpoints. Each accepted endpoint is detached from the search procedure, freshly re-noised at a flow time $t \sim \mathcal{U}[0.1, 1.0]$, and treated as a fixed flow-matching target:

$$\mathbf{x}_t^{(s,m)} = (1-t)\mathbf{x}_0^{(s,m)} + t\mathbf{x}_1^{(s,m)}, \qquad \mathbf{u}^{(s,m)} = \mathbf{x}_1^{(s,m)} - \mathbf{x}_0^{(s,m)}. \tag{9}$$

The resulting supervision loss on a firing data-parallel rank r is

$$\mathcal{L}_{\mathrm{SFT}}^{(r)} = \frac{1}{|\mathcal{S}_r|} \sum_{s \in \mathcal{S}_r} \sum_{m \in \{\mathrm{bb,lat}\}} \left\| \mathbf{v}_\theta^m \left(\mathbf{x}_t^{(s)}, t, \mathcal{C} \right) - \mathbf{u}^{(s,m)} \right\|_{\mathrm{mask},m}^2, \tag{10}$$

where $\mathcal{S}_r$ denotes the search endpoints accepted on rank r and $\mathbf{x}_t^{(s)} = (\mathbf{x}_t^{(s,\mathrm{bb})}, \mathbf{x}_t^{(s,\mathrm{lat})})$. Under data-parallel training over N_{rank} ranks, the optimized objective is

$$\mathcal{L}_{\mathrm{SFT}} = \frac{1}{N_{rank}} \sum_{r=1}^{N_{rank}} \mathbf{1}\{|\mathcal{S}_r| > 0\} \mathcal{L}_{\mathrm{SFT}}^{(r)}, \tag{11}$$

so ranks without an accepted endpoint contribute no SFT gradient. Search supervision is active during a subset of post-training and is subsequently disabled, after which optimization continues with on-policy DGPO and reference regularization alone. Its duration is a configurable component of SARP rather than a fixed part of the optimization objective. Our ablations in Appendix O show that its effect depends on search strength and task dynamics.

3.3 Joint SARP Objective

The complete objective at optimization step n is

$$\mathcal{L}_{\mathrm{SARP}} = \mathbb{E}_t \left[\mathcal{L}_{\mathrm{DGPO}} + \beta_{\mathrm{ref}} \mathcal{L}_{\mathrm{ref}} \right] + \lambda_{\mathrm{SFT}}(n) \mathcal{L}_{\mathrm{SFT}}. \tag{12}$$

where $\lambda_{\mathrm{SFT}}(n)$ controls the contribution of direct endpoint supervision and is set to zero after the supervised phase. The DGPO term provides group-relative optimization over on-policy rollouts, whereas $\mathcal{L}_{\mathrm{SFT}}$ directly fits selected high-reward endpoints.

4 Experiment

We evaluate SARP on two complementary all-atom protein-design settings: small-molecule binder design and target-specific enzyme design. These experiments test whether search-augmented post-training improves standard-sampling quality across both broad binding objectives and precise catalytic-geometry constraints. We then analyze the interaction between search supervision, reinforcement, and the coupled backbone and latent flows. More visualizations are in Appendix B.

4.1 Optimization of Small-Molecule Binder Design

Task setup. We post-train the base model on a subset of Plinder (Durairaj et al., 2024) that Proteína-Complexa (Didi et al., 2026) is also trained on. Generations are conditioned on the experimentally observed 3D conformation and coordinates of a small molecule, where the generated protein length is set to match that of the corresponding native binder. The training subset contains 2,170 complexes from 2,155 PDB entries and 1,564 unique ligands. For evaluation, we use the small-molecule binder benchmark by RFDiffusion3 (Butcher et al., 2025) and Proteína-Complexa, comprising FAD, IAI, OQO, and SAM. More details and clarifications are provided in Appendix D.

Experimental setup. We mainly optimize two complementary measures of binder quality: protein–ligand binding energy and RosettaFold3 (RF3) protein–ligand min-iPAE (Corley et al., 2025). Binding energy is computed using the PyRosetta separated-interface energy (Chaudhury et al., 2010) after contact gating and local relaxation, rewarding physically favorable protein–ligand interactions. RF3

Table 1: **Mean performance across four ligands under the Binding Energy + RosettaFold3 min-iPAE reward.** Reported metrics are macro-averaged over FAD, IAI, OQO, and SAM on the top 30% of ranked designs. RL-based methods are compared under a maximum training budget of 200 optimization steps. SARP uses beam width $W = 2$ in this table. Best results are shown in **bold**, and second-best results are underlined.

Method	RL	Search Aug.	$\Delta G \downarrow$ REU	min-iPAE $\downarrow$ Å	scRMSD $\downarrow$ Å	ligRMSD $\downarrow$ Å	pLDDT $\uparrow$	Div. $\downarrow$ meanTM
Base model	–	–	1.86	5.82	3.61	9.87	69.35	0.351
GRPO	✓	–	-3.48	5.94	3.77	9.93	69.33	**0.348**
DGPO	✓	–	-4.27	4.52	1.87	9.62	75.72	0.433
Online Search-SFT	–	✓	-2.70	4.65	1.29	12.66	**79.29**	0.577
SARP (full supervision)	✓	✓	-11.15	**3.67**	**1.22**	6.99	79.10	0.410
SARP (turn-off @ 90)	✓	✓	**-12.69**	4.28	1.31	7.14	78.55	0.421
SARP (turn-off @ 135)	✓	✓	-12.49	4.57	1.36	**6.89**	77.67	0.415

min-iPAE is computed by refolding the generated sequence with the ligand, which provides a scalable interface-confidence signal for online post-training. As fully described in Appendix E, structure penalties and diversity reward are also applied. We use $w_{\text{bind}} = 1$, $w_{\text{RF3}} = 1$, $w_{\text{div}} = 0.1$, and $\gamma_{div} = 0.5$. We compare several post-training variants to isolate the effects of endpoint-based reinforcement, direct endpoint supervision, and supervision source. GRPO directly optimizes sampled generation trajectories and DGPO applies trajectory-free endpoint optimization using stochastic rollouts. Online Search-SFT distills high-reward endpoints obtained through search back to the model with pure supervised learning without RL. SARP combines DGPO with search-derived endpoint supervision, either throughout full training or only during an initial warm-start stage. We additionally evaluate SARP-Rollout (Appendix G.6, Appendix P), a matched variant that replaces search-derived targets with high-reward endpoints selected directly from the on-policy rollout group.

Results. Table 1 reports macro-averaged performance over the four evaluation ligands and Figure 2 shows SARP improves rapidly and achieves higher ceilings than pure DGPO. For each method and ligand, we generate 200 designs and evaluate the top 30% ranked by a predefined combination of ΔG and RF3 min-iPAE, reflecting the practical design setting in which only the highest-ranked candidates from a fixed generation budget are advanced for downstream evaluation. Per-ligand results are provided in Appendix H. Among RL-only methods, DGPO improves binding energy and interface confidence over the pretrained model, while increasing its rollout group to $G = 64$ mainly benefits binding energy and does not reproduce SARP's overall gains (Appendix K). SARP further improves design quality, reaching avg. $\Delta G = -12.69$ REU and full-supervision SARP achieves avg. min-iPAE $= 3.67$ Å, scRMSD $=$ 1.22 Å, and pLDDT $= 79.10$, and retains substantially greater diversity than Search-SFT and comparable or better diversity than DGPO. SARP-Rollout attains similar binder quality but lower structural diversity (Appendix P), a trend that becomes more pronounced in the enzyme setting in Section 4.2. These improvements persist under AF3 cross-evaluation (Appendix I). Additional ablations show that diversity loss depends on search strength and supervision duration, with stronger persistent search causing greater concentration, where the structural-diversity reward helps counteract this effect (Appendices O and M).

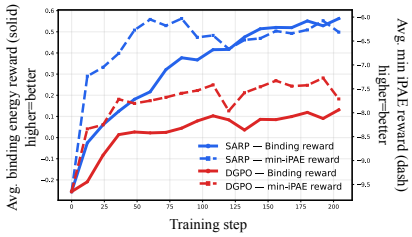

Figure 2: Training dynamics of the binding-energy and RF3 min-iPAE rewards during post-training.

Reward hacking. Separated-interface energy can favor non-interacting proteins placed far from the ligand by assigning near-zero energy. We prevent this shortcut with a contact gate that penalizes insufficient protein–ligand proximity and contacts, restricting the binding-energy reward to interacting complexes (Figure 5; Appendix E.2).

4.2 Optimization of Enzyme Design

Task setup. We further evaluate SARP on target-specific enzyme-design optimization following the atomic-motif setting of RFDiffusion2 (Ahern et al., 2026) and Proteína-Complexa (Didi et al., 2026). Each task specifies a ligand together with the identities and target atomic coordinates of

catalytic residues. The ligand and catalytic motif are fixed as conditioning inputs and the base generator generates the remaining protein structure and sequence. We selected five hard targets with low pass rate as shown in Proteína-Complexa (Didi et al., 2026): M0732(5 res islands), M0157(6 res islands), M0552(5 res islands), M0904(6 res islands), and M0375 (7 res islands), which differ in ligand geometry and catalytic configuration. Unlike the small-molecule binder experiment, each target is optimized independently. All models are evaluated using 100 standard generations without search at inference time.

Enzyme-specific rewards. We design rewards to directly reflect Proteína-Complexa (Didi et al., 2026) success criteria for enzyme design, which measure recovery of the prescribed catalytic configuration while maintaining a physically valid active site. They combine catalytic-residue identity match, catalytic-motif insertion RMSD, and backbone RMSD following Proteína-Complexa. We also add general geometry penalties and diversity reward. Full reward definitions are provided in Appendix E. We use $w_{\text{cat_id}} = 1$, $w_{\text{cat_rmsd}} = 2$, $w_{\text{bb}} = 1$, $w_{\text{div}} = 0.2$, and $\gamma_{div} = 0.3$.

Experimental setup. The overall training and evaluation setup follows the small-molecule binder experiment. Following Proteína-Complexa, a design is successful if all catalytic residue identities are recovered, the RF3-refolded backbone has scRMSD ≤ 2.0 Å, the catalytic functional groups have all-atom scRMSD ≤ 1.5 Å, and no designed protein atom lies within 1.5 Å of the ligand. We use $n_{\text{branches}} = 4$ and beam width 8 for all enzyme targets except M0375, which uses $n_{\text{branches}} = 2$ and beam width 2 to encourage diversity for this hard target. Training-time diversity determines the duration of endpoint supervision, which is disabled once diversity begins to decline. We monitor the corresponding reward metrics and diversity on training rollouts, stopping training once the task metrics plateau and before substantial diversity loss. The final 100-design evaluation set is not used for checkpoint selection.

Results. Table 2 summarizes performance across the five enzyme targets. DGPO consistently improves pass rate and geometric recovery over the pretrained model. Simply doubling its rollout group to $G = 64$ does not reproduce the SARP gains, yielding 27.8 unique passing clusters on average compared with 35.6 for the default $G = 32$ DGPO and 42.2 for SARP (Appendix K). SARP achieves strong overall performance, reaching pass rates of 24–89%, backbone RMSD of approximately 0.6 Å, and an average catalytic-site RMSD of 1.38 Å. A matched endpoint-source ablation shows that rollout-derived supervision attains similar local geometric quality, whereas search-derived supervision preserves substantially greater structural diversity (Appendix P). Search-SFT shows a similar quality–diversity trade-off, achieving strong geometric recovery but often concentrating successful generations into only a few structural basins. Additional training-dynamics analyses show that search frequently improves over native rollouts, but this advantage narrows over training and wider search supervision accelerates diversity loss (Appendix O). This is consistent with the inference-time search ablation in Appendix J. Under standard sampling without inference-time search, SARP retains 18–54 unique passing clusters across the five targets. AF3 cross-evaluation further preserves the main improvements in pass rate, catalytic geometry, and unique passing designs (Appendix I).

4.3 Gradient Analysis Reveals Distinct Optimization Roles of Search Supervision and Reinforcement

To understand how search-derived endpoint supervision interacts with DGPO and how post-training acts on the coupled backbone and latent flows, we instrument a small-molecule binder SARP run with four search branches and beam width $W = 8$. Full gradient definitions and numerical results are provided in Appendix N.

Search-SFT increasingly departs from the DGPO update direction. With $\lambda_{\text{SFT}} = 0.05$, the weighted Search-SFT gradient norm is on average $0.74\times$ that of DGPO. Its cosine with the DGPO gradient decreases from -0.004 over steps 0–19 to -0.119 over steps 61–80, and the fraction of the DGPO-directed component canceled by Search-SFT increases from approximately 0.6% to 12%. This growing disagreement offers a plausible mechanism for why the duration of search supervision matters, analogous to recent observations that auxiliary alignment objectives in diffusion training can become increasingly misaligned with the primary denoising objective and benefit from early termination (Wang et al., 2026).

The two coupled flows experience different optimization pressures. After normalizing for output dimensionality, the DGPO preference gradient is relatively stronger on the non-C_α latent flow, with

Table 2: **Target-specific enzyme-design optimization.** Results use 100 generations per model. Pass is the fraction satisfying all AME criteria; Unique pass is the number of Foldseek clusters at TM$\geq$ 0.5 among passing designs. For SARP, Train #step is reported as $a(b)$, where a is the total number of post-training steps and b is the number of steps with auxiliary endpoint supervision enabled. Best results within each target are shown in **bold**, and second-best results are underlined.

Target	Method	Train #step	Cat. match ↑	Cat. RMSD ↓ Å	BB RMSD ↓ Å	Lig. clash ↓ %	Pass ↑ %	Unique pass ↑
M0732	Base model	–	0.998	2.95	1.49	0.008%	20%	17
	GRPO	104	0.994	2.24	0.90	0.009%	18%	11
	DGPO	100	0.990	1.53	0.97	0.021%	62%	42
	Online Search-SFT	30	0.984	2.07	1.06	**0.005%**	17%	1
	SARP	60(20)	**1.000**	**1.11**	**0.58**	0.006%	**89%**	**44**
M0157	Base model	–	0.999	2.74	1.77	**0.001%**	18%	17
	GRPO	122	0.999	1.80	0.71	0.004%	30%	26
	DGPO	170	**1.000**	1.66	0.63	**0.001%**	57%	33
	Online Search-SFT	20	0.997	2.36	0.63	0.003%	20%	17
	SARP	90(30)	**1.000**	**1.34**	**0.58**	0.003%	**74%**	**43**
M0552	Base model	–	0.998	3.09	1.52	0.004%	11%	7
	GRPO	74	0.998	2.30	0.81	**0.003%**	7%	6
	DGPO	120	0.998	1.62	0.74	0.015%	60%	40
	Online Search-SFT	40	0.997	2.15	0.68	0.011%	13%	1
	SARP	90(30)	**1.000**	**1.36**	**0.57**	**0.003%**	**72%**	**54**
M0904	Base model	–	**1.000**	2.39	1.28	**0.000%**	21%	12
	GRPO	126	0.999	1.94	0.90	0.021%	24%	21
	DGPO	120	0.999	1.61	0.99	0.005%	75%	47
	Online Search-SFT	40	**1.000**	2.04	0.78	0.003%	40%	3
	SARP	140(20)	**1.000**	**1.20**	**0.60**	**0.000%**	**88%**	**52**
M0375	Base model	–	0.999	3.18	1.49	0.011%	5%	3
	GRPO	94	**1.000**	2.62	0.91	0.005%	2%	2
	DGPO	110	**1.000**	2.26	1.15	0.006%	18%	16
	Online Search-SFT	50	**1.000**	2.63	0.81	**0.004%**	6%	1
	SARP	80(20)	**1.000**	**1.91**	**0.63**	0.008%	**24%**	**18**

$R_{\text{policy}}^{\text{lat/bb}} \approx 1.50$, whereas moving-reference regularization is relatively stronger on the explicit backbone flow, with $R_{\text{ref}}^{\text{lat/bb}} \approx 0.72$. However, this gradient asymmetry does not translate directly into proportional sequence-versus-backbone displacement. Along the corresponding $W = 8$ lineage, sequence-associated changes become more pronounced over training. Alanine frequency increases from 4.5% to 11.9%, sequence entropy shifts by -4.35 pretrained-model standard deviations, and hydrophobic-residue fraction increases by 3.09 standard deviations. The backbone simultaneously reorganizes toward more local packing: long-range contacts decrease by 1.00 standard deviations and the fraction of C_α pairs at 8–10 Å increases by 1.55, while radius of gyration is essentially unchanged.

Together, these analyses show that gradient allocation and output-space adaptation are distinct, with endpoint supervision and the coupled model parameterization shaping how optimization is expressed in sequence and structure. This motivates future methods to control latent and backbone adaptation, endpoint supervision, and distributional drift separately.

5 Discussion

Our results suggest that direct supervision of high-reward endpoints is an effective complement to endpoint-based reinforcement, and that supervision source and duration shape the quality–diversity trade-off. In the matched endpoint-source ablation, rollout-derived supervision often matches search-derived SARP in local geometric quality but produces fewer structurally distinct successful designs. This is consistent with rollout self-distillation concentrating updates on modes already represented in the current policy, whereas search explores additional continuations and selects improved endpoints before distillation.

Gradient analysis provides a complementary optimization view. Under strong beam search, Search-SFT and DGPO become increasingly misaligned over training, while DGPO and moving-reference regularization exert different pressures on the non-Cα latent and explicit backbone flows. These asymmetries motivate adaptive control of supervision strength, duration, and modality-specific regularization. Our study relies on computational reward proxies, which may not fully predict experimental performance. Nevertheless, SARP converts expensive search into temporary training-time supervision, amortizing high-reward design strategies into fast standard sampling.

AI USE STATEMENT

Generative AI tools were used during the preparation of this work to assist with research ideation and execution, to draft portions of the manuscript, and to improve the clarity and presentation of the writing. Uses included discussion of experimental and analysis strategies, assistance with code and research workflows, and drafting or polishing manuscript text. All AI-generated outputs were reviewed, verified, and revised by the authors as appropriate.

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.

Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. *Advances in neural information processing systems*, 30, 2017.

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.

Sidhartha Chaudhury, Sergey Lyskov, and Jeffrey J Gray. Pyrosetta: a script-based interface for implementing molecular modeling algorithms using rosetta. *Bioinformatics*, 26(5):689–691, 2010.

Nathaniel Corley, Simon Mathis, Rohith Krishna, Magnus S Bauer, Tuscan R Thompson, Woody Ahern, Maxwell W Kazman, Rafael I Brent, Kieran Didi, Andrew Kubaney, et al. Accelerating biomolecular modeling with atomworks and rf3. *BioRxiv*, 2025.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael M Bronstein, et al. Scaling atomistic protein binder design with generative pretraining and test-time compute. *arXiv preprint arXiv:2603.27950*, 2026.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

Janani Durairaj, Yusuf Adeshina, Zhonglin Cao, Xuejin Zhang, Vladas Oleinikovas, Thomas Duignan, Zachary McClure, Xavier Robin, Gabriel Studer, Daniel Kovtun, et al. Plinder: The protein-ligand interactions dataset and evaluation resource. *BioRxiv*, pp. 2024–07, 2024.

Jiajun Fan, Shuaike Shen, Chaoran Cheng, Yuxin Chen, Chumeng Liang, and Ge Liu. Online reward-weighted fine-tuning of flow matching with wasserstein regularization. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Yuliang Fan, Zaikai He, Bin Li, Bin He, Mingshu Zhang, Jian Zhang, and Haicang Zhang. Pocketx: Preference alignment for protein pockets design through group relative policy optimization. *bioRxiv*, pp. 2025–12, 2025b.

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. *arXiv preprint arXiv:2507.09466*, 2025.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Rohith Krishna, Jue Wang, Woody Ahern, Pascal Sturmfels, Preetham Venkatesh, Indrek Kalvet, Gyu Rie Lee, Felix S Morey-Burrows, Ivan Anishchenko, Ian R Humphreys, et al. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 384(6693):eadl2528, 2024.

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

Sidney Lyayuga Lisanza, Jacob Merle Gershon, Samuel WK Tipps, Jeremiah Nelson Sims, Lucas Arnoldt, Samuel J Hendel, Miriam K Simma, Ge Liu, Muna Yase, Hongwei Wu, et al. Multistate and functional protein design using rosettafold sequence space diffusion. *Nature biotechnology*, 43(8):1288–1298, 2025.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in neural information processing systems*, 38:40783–40818, 2026.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Yihong Luo, Tianyang Hu, and Jing Tang. Reinforcing diffusion models by direct group preference optimization. *arXiv preprint arXiv:2510.08425*, 2025.

David McAllister, Miika Aittala, Tero Karras, Janne Hellsten, Angjoo Kanazawa, Timo Aila, and Samuli Laine. Finite difference flow optimization for rl post-training of text-to-image models. *arXiv preprint arXiv:2603.12893*, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot-Desenonges, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shahriari, Sarah Perrin, Abram Friesen, Geoffrey Cideron, et al. Bond: Aligning llms with best-of-n distillation. In *International Conference on Learning Representations*, volume 2025, pp. 59040–59060, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Hannes Stark, 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*, pp. 2025–11, 2025.

Jithendaraa Subramanian, Shivakanth Sujit, Niloy Irtisam, Umong Sain, Riashat Islam, Derek Nowrouzezahrai, and Samira Ebrahimi Kahou. Reinforcement learning for sequence design leveraging protein language models. *arXiv preprint arXiv:2407.03154*, 2024.

Ziqiao Wang, Wangbo Zhao, Yuhao Zhou, Zekai Li, Zhiyuan Liang, Mingjia Shi, Xuanlei Zhao, Pengfei Zhou, Kaipeng Zhang, Zhangyang Wang, et al. Repa works until it doesn't: Early-stopped, holistic alignment supercharges diffusion training. *Advances in Neural Information Processing Systems*, 38:136854–136887, 2026.

Ziwen Wang, Jiajun Fan, Ruihan Guo, Thao Nguyen, Heng Ji, and Ge Liu. Proteinzero: Self-improving protein generation via online reinforcement learning. *arXiv preprint arXiv:2506.07459*, 2025.

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.

Fanglei Xue, Andrew Kubaney, Zhichun Guo, Joseph K Min, Ge Liu, Yi Yang, and David Baker. Improving protein sequence design through designability preference optimization. *arXiv preprint arXiv:2506.00297*, 2025a.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025b.

Wei Yang, Shunzhi Wang, Gyu Rie Lee, Jason Z Zhang, Alexis Courbet, David Juergens, Xinru Wang, Thomas Schlichthaerle, Mohamad Abedi, Robert Ragotte, et al. The past, present and future of de novo protein design. *Nature*, 652(8112):1139–1152, 2026.

Chengxin Zhang, Morgan Shine, Anna Marie Pyle, and Yang Zhang. Us-align: universal structure alignments of proteins, nucleic acids, and macromolecular complexes. *Nature methods*, 19(9): 1109–1115, 2022.

APPENDIX

A Extended Related Work **15**

B Qualitative Examples **16**

C Background on the Base Protein Generator **16**

D Details of the small molecule binders training dataset **18**

E Detailed Reward Implementation **18**

E.1 General Structural Rewards 19

E.2 Small-Molecule Binder Rewards 20

E.3 Enzyme-Design Rewards 20

E.4 Reward Aggregation and Group-Relative Advantage 21

F Derivation of the Contact-Gate Thresholds **22**

G Additional Method Details **22**

G.1 Complete SARP Training Procedure 22

G.2 Endpoint Re-noising Details 23

G.3 Modality-Specific Loss Normalization 24

G.4 Reference-Policy Schedule 24

G.5 Reward-Guided Beam Search 24

G.6 Matching the Effective Endpoint-Supervision Dose 25

G.7 Reward-Normalization Edge Cases 26

G.8 Default Training Configuration 26

G.9 Training horizon and checkpoint selection 26

H Full Results and Additional Analysis for Small-Molecule Binder Design **27**

I Cross-Evaluator Validation with AlphaFold3 **28**

I.1 Small-molecule binder cross-evaluation 28

I.2 Enzyme-design cross-evaluation 28

J Comparison with Inference-Time Search **29**

K Effect of DGPO Rollout Group Size **30**

L Preference-Learning Dynamics in DGPO **32**

M Effect of the Structural-Diversity Reward **34**

N Gradient Decomposition of Joint Post-Training **34**

N.1 Gradient-Decomposition Probe 35
N.2 Search-SFT and DGPO Become Increasingly Misaligned 35
N.3 Policy and Reference Regularization Prefer Different Flows 35
N.4 Reward and Structural-Validity Terms Co-Drive the Policy Update 36
N.5 Output-Level Changes Along the Instrumented $W = 8$ Lineage 36

O Ablation on Beam-Search Scale and Supervision Duration **37**

P Ablation on Search-Derived vs. Rollout-Derived Endpoint Supervision **40**

Q Trajectory-Based GRPO Implementation **41**

A Extended Related Work

Generative Protein Structure and Interaction Design. Generative protein-design methods have progressed from backbone generation to joint sequence–structure co-generation, and more recently to all-atom modeling of biomolecular interactions. RFDiffusion (Watson et al., 2023) introduced diffusion-based protein backbone generation and demonstrated conditional design for motif scaffolding, symmetric assemblies, and protein binders. RFDiffusion2 (Ahern et al., 2026) extended this framework to atom-level enzyme active-site scaffolding by conditioning generation on catalytic-residue identities and target atomic geometries. Subsequent work moved beyond backbone-only generation to jointly model protein sequence and structure. ProteinGenerator (Lisanza et al., 2025) performs joint sequence–structure generation using a RoseTTAFold-based diffusion model. Discrete Flow Models (DFMs) (Campbell et al., 2024) introduce a multimodal flow framework that jointly generates continuous structures and discrete sequences through coupled continuous and discrete flows. More recent models further extend generation to complete biomolecular complexes. RFDiffusion3 (Butcher et al., 2025) performs all-atom generation of biomolecular interactions, BoltzGen (Stark et al., 2025) develops a unified all-atom framework for binder design across diverse target classes, and Proteína-Complexa (Didi et al., 2026) jointly models explicit C_α coordinates and non-C_α residue latents. These systems are commonly deployed in generate–score–filter pipelines, where sampled candidates are evaluated using structural confidence, geometric constraints, or energy-based metrics.

Reinforcement and Preference Optimization for Protein Design. Reinforcement learning and preference optimization have been used to adapt protein generators toward different objectives. Early work primarily considered sequence-level optimization. Subramanian et al. (Subramanian et al., 2024) learn a mutation policy for protein sequence design, using protein language models or periodically updated proxy models to provide rewards for biological plausibility. ProteinZero (Wang et al., 2025) instead applies online reinforcement learning to an inverse-folding model, combining structure-based feedback from ESMFold, a learned $\Delta\Delta G$ predictor, reference-policy regularization, and embedding-level diversity regularization to improve designability, stability, and sequence diversity. Designability preference optimization (Xue et al., 2025a) constructs preference pairs from AlphaFold-predicted structural quality and fine-tunes LigandMPNN using DPO and residue-level DPO, thereby aligning sequence generation more directly with the likelihood of folding into the intended backbone. More recently, PocketX (Fan et al., 2025b) extends online reinforcement learning beyond sequence-only generation by applying GRPO to ligand-conditioned pocket structure–sequence co-design, optimizing properties such as Vina binding score and evolutionary plausibility.

Group-Relative Optimization for Diffusion and Flow Models. Group Relative Policy Optimization compares multiple samples generated under the same condition and normalizes rewards within the group, avoiding a separately trained value model (Shao et al., 2024). Recent work has extended group-relative optimization from autoregressive models to diffusion and flow-based generators. DanceGRPO (Xue et al., 2025b) treats the denoising process as a sequence of policy decisions for visual generation. Flow-GRPO (Liu et al., 2026) applies online reinforcement learning to flow-matching models through stochastic sampling and trajectory-level policy optimization. These methods retain information along the generation trajectory and compute policy updates from denoising decisions. DGPO (Luo et al., 2025) instead performs direct group preference optimization from completed samples by re-noising final outputs and comparing their denoising losses under online and reference policies.

Search and Test-Time Compute for Generative Design. Search and test-time computation improve generative design by allocating additional computation to promising candidates. Protein-design pipelines often combine stochastic generation with structure prediction, energy evaluation, filtering, sequence redesign, or iterative refinement (Watson et al., 2023; Butcher et al., 2025; Stark et al., 2025). Proteína-Complexa applies beam search within the generative process by branching intermediate flow states, completing multiple stochastic continuations, scoring the resulting structures, and retaining high-performing branches (Didi et al., 2026). More broadly, search-generated outputs can be reused as supervision through self-training or policy distillation, in which samples produced using a stronger or more computationally expensive procedure are transferred into a cheaper model.

Search-Augmented Policy Improvement. A related line of work combines reinforcement learning with supervised updates on high-quality samples discovered through search or policy sampling.

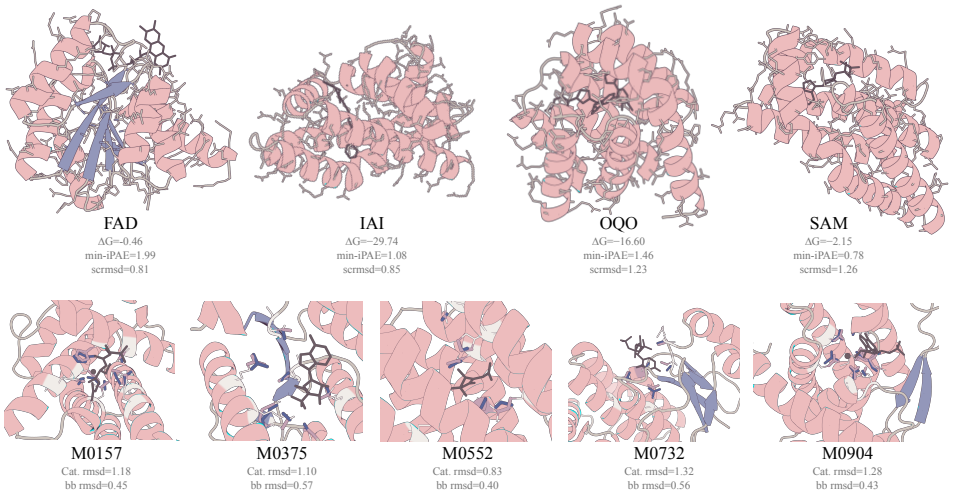

Figure 3: Representative SARP-generated designs under standard sampling. **Top:** small-molecule binder designs for FAD, IAI, OQO, and SAM. **Bottom:** enzyme-design examples for M0157, M0375, M0552, M0732, and M0904.

In game playing, AlphaGo Zero and Expert Iteration use tree search to construct improved action targets that are subsequently distilled into the policy network, separating expensive search from the policy that generalizes its discoveries (Silver et al., 2017; Anthony et al., 2017). Similar ideas have appeared in language-model post-training. ReST generates candidate outputs from the current policy and reuses reward-filtered samples for further training (Gulcehre et al., 2023). RAFT and Best-of-N distillation explicitly fine-tune models toward high-reward generations selected from larger candidate sets (Dong et al., 2023; Sessa et al., 2025). These approaches illustrate a broader principle in which additional computation is used to identify better-than-typical samples and the resulting examples are subsequently incorporated into the model, reducing the need to reproduce the same search procedure at inference time.

B Qualitative Examples

Figure 3 shows some designs generated by SARP under standard sampling for both small-molecule binder and enzyme-design tasks.

C Background on the Base Protein Generator

Proteína-Complexa (Didi et al., 2026) is a fully atomistic conditional protein generator built on the partially latent flow-matching architecture of La-Proteína (Geffner et al., 2025). It jointly generates a protein sequence and its all-atom structure while conditioning on a molecular design target, such as a protein surface, small molecule, or prescribed catalytic motif. The original framework combines a pretrained generative prior with inference-time optimization methods, including best-of-N sampling, beam search, Feynman–Kac steering, and Monte Carlo tree search. In this work, we use a pretrained base model that follows the Proteína-Complexa architecture (Didi et al., 2026) as the base policy and study how its sampling distribution can instead be improved through reinforcement post-training.

Partially latent all-atom representation. For a protein of length L, Proteína-Complexa separates the representation into explicit C_α coordinates and continuous residue-level latents:

$$\mathbf{x} = \left(\mathbf{x}^{\mathrm{bb}}, \mathbf{x}^{\mathrm{lat}}\right), \qquad \mathbf{x}^{\mathrm{bb}} \in \mathbb{R}^{L\times 3}, \qquad \mathbf{x}^{\mathrm{lat}} \in \mathbb{R}^{L\times d_z}. \tag{13}$$

Here, $\mathbf{x}^{\mathrm{bb}}$ contains the explicit C_α coordinates and each latent vector encodes the amino-acid identity and the remaining non-C_α atomic geometry of its residue, including backbone N, C, and O atoms and residue-specific sidechains. The representation is obtained from a variational autoencoder with encoder

$$E : \left(\mathbf{x}^{\mathrm{bb}}, \mathbf{x}^{\neg\mathrm{bb}}, \mathbf{s}\right) \mapsto \mathbf{x}^{\mathrm{lat}}, \tag{14}$$

and decoder

$$D : \left(\mathbf{x}^{\mathrm{bb}}, \mathbf{x}^{\mathrm{lat}}\right) \mapsto \left(\hat{\mathbf{x}}^{\neg\mathrm{bb}}, \hat{\mathbf{s}}\right), \tag{15}$$

where $\mathbf{s}$ denotes the amino-acid sequence and $\mathbf{x}^{\neg\mathrm{bb}}$ denotes the non-C_α coordinates in an Atom37 representation. This converts the mixed discrete–continuous all-atom generation problem into a fixed-size continuous representation per residue.

Coupled flow matching. After training the autoencoder, Proteína-Complexa models the joint distribution of explicit C_α coordinates and residue latents using flow matching. For modality $m \in \{\mathrm{bb}, \mathrm{lat}\}$, a clean endpoint $\mathbf{x}_1^m$ is paired with a Gaussian prior sample $\mathbf{x}_0^m$ and interpolated as

$$\mathbf{x}_{t_m}^m = (1 - t_m)\mathbf{x}_0^m + t_m \mathbf{x}_1^m, \qquad \mathbf{u}^m = \mathbf{x}_1^m - \mathbf{x}_0^m. \tag{16}$$

The backbone and latent modalities may use separate interpolation times and schedules. A shared transformer denoiser receives the complete coupled state and jointly predicts the two velocity fields,

$$\left(v_\theta^{\mathrm{bb}}, v_\theta^{\mathrm{lat}}\right) = v_\theta\left(\mathbf{x}_{t_{\mathrm{bb}}}^{\mathrm{bb}}, \mathbf{x}_{t_{\mathrm{lat}}}^{\mathrm{lat}}, t_{\mathrm{bb}}, t_{\mathrm{lat}}, C\right), \tag{17}$$

where C denotes the target and other design conditions. The corresponding training objective is

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\left[\left\|v_\theta^{\mathrm{bb}} - \mathbf{u}^{\mathrm{bb}}\right\|^2 + \left\|v_\theta^{\mathrm{lat}} - \mathbf{u}^{\mathrm{lat}}\right\|^2\right]. \tag{18}$$

Although the two output modalities have distinct targets and schedules, they are predicted from the same joint state. Consequently, both flows jointly determine the decoded sequence and all-atom geometry.

Coupled stochastic sampling. Generation begins from independent Gaussian prior states for the explicit C_α coordinates and the non-C_α residue latents. Although the two modalities are evolved jointly by a shared denoiser, we use the same modality-specific time schedules as Proteína-Complexa for the backbone and latent variables, with the C_α coordinates denoised more rapidly than the residue latents. For modality $m \in \{\mathrm{bb}, \mathrm{lat}\}$, the predicted flow velocity v_θ^m is converted into the score of the corresponding intermediate distribution, denoted by s_θ^m, and sampling follows the SDE

$$d\mathbf{x}_t^m = \left[v_\theta^m\left(\mathbf{x}_t, t, C\right) + g_m(t) s_\theta^m\left(\mathbf{x}_t, t, C\right)\right] dt + \sqrt{2 g_m(t) \lambda_{\mathrm{noise}}^m}\, d\mathbf{W}_t^m, \tag{19}$$

where $g_m(t)$ controls the score-correction and diffusion strength and $\lambda_{\mathrm{noise}}^m < 1$ scales the diffusion variance, yielding low-temperature stochastic sampling. Following the official Proteína-Complexa implementation, stochastic noise is disabled near the clean endpoint and sampling switches to a deterministic low-temperature tail in which the recovered score is rescaled before being converted back to a velocity field. This retains stochastic exploration over most of the trajectory while avoiding additional random perturbations immediately before decoding. Once both modalities reach their terminal states, the generated pair $\left(\hat{\mathbf{x}}^{\mathrm{bb}}, \hat{\mathbf{x}}^{\mathrm{lat}}\right)$ is passed through the frozen autoencoder decoder, which reconstructs the non-C_α Atom37 coordinates and predicts the amino-acid identity of each residue, generating a jointly generated protein sequence and all-atom structure.

Search over coupled sampling trajectories. The original Proteína-Complexa framework applies inference-time search directly to the partially latent sampling process. A beam element is the complete coupled state

$$\mathbf{x}_t = \left(\mathbf{x}_t^{\mathrm{bb}}, \mathbf{x}_t^{\mathrm{lat}}\right), \tag{20}$$

rather than a backbone-only structure. At each search stage, every retained state is copied into multiple branches, and each branch is advanced through several stochastic denoising steps using independently sampled noise. Each intermediate branch is then rolled out to the terminal backbone–latent state, decoded into a clean all-atom structure–sequence pair, and evaluated using a downstream reward. The highest-scoring intermediate states are retained as the beam for the next stage. In the original framework, this procedure is used only at inference time and must be repeated independently for every design target.

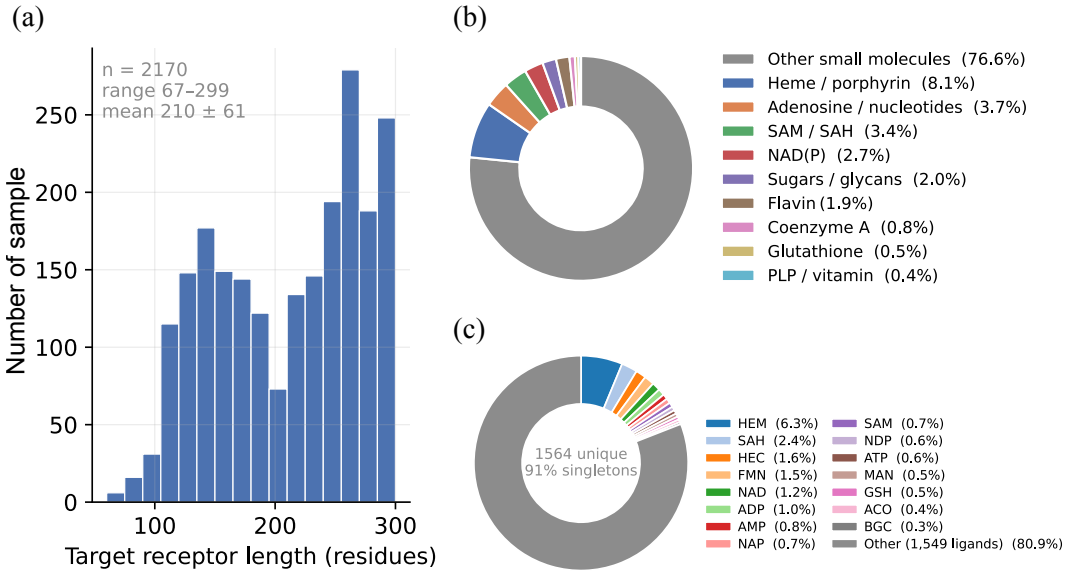

Figure 4: **Composition of the Plinder subset used for small-molecule binder post-training. (a)** Distribution of target receptor lengths across 2,170 systems, covering small-to-medium single-domain proteins with lengths ranging from 67 to 299 residues. **(b)** Ligand functional classes. **(c)** Frequencies of the top-15 ligand codes, with all remaining ligands grouped as "Other," illustrating the long-tailed ligand diversity of the dataset: 1,564 unique ligands, 91% singletons, and no single ligand dominating the training set.

D Details of the small molecule binders training dataset

We provide an overview of the composition of the Plinder subset used for post-training in Figure 4. An example input condition provided to the base generator for rollout generation is:

```
input structure: 7mfx__1__1.A__1.E.cif
ligand name: ZFJ
protein length: 133
```

Evaluation-ligand overlap and benchmark scope. We follow the established RFdiffusionAA (Krishna et al., 2024)/RFdiffusion3 (Butcher et al., 2025) small-molecule benchmark comprising FAD, SAM, IAI, OQO. Importantly, this benchmark was not designed as a strict unseen-ligand split: RFdiffusionAA explicitly evaluates FAD and SAM despite their frequent occurrence in its training set, while IAI and OQO were chosen as ligands more distant from the training distribution; RFdiffusion3 subsequently uses the same four-ligand benchmark. Accordingly, ligand-identity overlap for FAD and SAM is consistent with the established evaluation protocol rather than unique to our post-training setup.

In our setting, repeated ligand identity also does not imply duplication of the same conditioning example. Conditioning uses the experimentally observed ligand coordinates from each protein–ligand complex, so occurrences of the same chemical component can correspond to different bound conformations, protein environments, and interaction geometries. We therefore interpret this benchmark as evaluating post-training effectiveness across a mixture of in-distribution and more distribution-shifted ligand conditions, rather than as a strict zero-shot test of unseen ligand identities.

E Detailed Reward Implementation

We organize the reward functions into four groups: general structural safeguards shared across tasks, diversity reward, small-molecule binder rewards, and enzyme-design rewards. Task-specific rewards

are standardized within each rollout group before combination and structural-validity penalties are applied separately to demote geometrically invalid samples.

E.1 General Structural Rewards

General structural-validity penalties. To prevent the policy from improving task rewards through structurally degenerate solutions, we apply one-sided penalties for backbone discontinuity, invalid decoded sidechains, backbone clashes, and protein–ligand clashes. For rollout sample g, let f_g^{bb} denote the chain-break-free backbone fraction, f_g^{sc} the valid-sidechain fraction, $f_g^{\text{bb-clash}}$ the backbone-clash fraction, and $f_g^{\text{lig-clash}}$ the ligand-clash fraction.

The chain-break-free fraction measures the proportion of consecutive Cα pairs whose distances remain within the valid backbone range. The valid-sidechain fraction measures the proportion of residues whose decoded sidechain atoms satisfy the required chemical and geometric validity checks. The backbone-clash and ligand-clash fractions measure the proportions of generated atoms involved in protein–protein and protein–ligand steric overlaps, respectively.

We define the structural-validity penalty for rollout g as

$$\begin{aligned} p^{(g)} = \lambda_{\text{geom}} \Big[& \big(\tau_{\text{bb}} - f_g^{\text{bb}}\big)_+ + \big(\tau_{\text{sc}} - f_g^{\text{sc}}\big)_+ \\ & + \big(f_g^{\text{bb-clash}} - \tau_{\text{bb-clash}}\big)_+ + \big(f_g^{\text{lig-clash}} - \tau_{\text{lig-clash}}\big)_+ \Big], \end{aligned} \tag{21}$$

where $(x)_+ = \max(0, x)$. We use

$$\tau_{\text{bb}} = 0.90, \qquad \tau_{\text{sc}} = 0.80, \qquad \tau_{\text{bb-clash}} = 0.05, \qquad \tau_{\text{lig-clash}} = 0, \qquad \lambda_{\text{geom}} = 1.0, \tag{22}$$

The same thresholds are used for both small-molecule binder and enzyme-design post-training.

In other words, structures with less than 90% backbone continuity, less than 80% valid sidechains, more than 5% backbone clashes, and any ligand clashes are penalized. Samples satisfying all four criteria receive no structural-validity penalty.

Structural-diversity reward. Group-relative optimization can progressively concentrate probability mass on structurally similar high-reward solutions. To discourage within-group redundancy, we define an additional group-level diversity reward from pairwise structural similarity among the G policy rollouts. Unlike the task-specific reward, this term depends on the full rollout group and is therefore computed only after all G structures have been generated. For generated structures $\{\mathcal{S}_1, \ldots, \mathcal{S}_G\}$, we extract the designed protein chain and compute the symmetric pairwise TM-score matrix

$$T_{ij} = \text{TM}(\mathcal{S}_i, \mathcal{S}_j) \in [0, 1], \qquad T_{ij} = T_{ji}, \qquad T_{ii} = 1. \tag{23}$$

We compute TM-score using `US-align` (Zhang et al., 2022), with length normalization averaged over the two structures and larger T_{ij} indicates greater structural similarity. We then convert pairwise similarity into a kernelized similarity score,

$$k_{ij} = \exp\left(-\frac{(1 - T_{ij})^2}{\gamma}\right), \tag{24}$$

and define the diversity reward for rollout i as

$$r_{\text{div},i} = -\frac{1}{G-1} \sum_{j \neq i} k_{ij}, \qquad r_{\text{div},i} \in [-1, 0]. \tag{25}$$

A sample that is highly similar to the rest of the rollout group receives $r_{\text{div},i}$ closer to -1, whereas a structurally distinct sample receives a value closer to 0, so higher values correspond to greater within-group structural novelty. The diversity reward is standardized within the rollout group in the same manner as the task-specific reward heads, with weight w_{div}. Search-generated candidates are excluded from the pairwise similarity matrix, so the diversity reward is computed only over samples generated by the policy.

E.2 SMALL-MOLECULE BINDER REWARDS

Interface binding-energy reward. The binding-energy reward measures whether the generated protein forms a favorable physical interface with the fixed ligand. Before evaluating the energy, we apply a contact gate to ensure that the ligand is physically engaged by the generated protein.

From the protein–ligand heavy-atom coordinates, we compute the minimum intermolecular distance $d_{\min}$, the ligand–pocket centroid distance d_{cen}, and the number of contacting protein residues n_{c} with at least one heavy atom within $5\,\text{Å}$ of the ligand. A structure passes the gate when

$$\text{pass} = \big(d_{\min} < 4.5\,\text{Å}\big) \wedge \big(d_{\text{cen}} < 8\,\text{Å}\big) \wedge \big(n_{\text{c}} \geq 8\big). \tag{26}$$

Structures that fail the gate are not assigned a favorable binding-energy reward and are ranked below physically engaged interfaces. The gate prevents the policy from exploiting separated-interface energy by placing the protein far from the ligand. The thresholds were selected from the empirical distributions of these geometric statistics in reference protein-ligand complexes, where the calibration procedure and supporting analysis are provided in Appendix F.

For structures that pass the contact gate, we compute the PyRosetta separated-interface energy

$$\Delta G = E_{\text{complex}} - E_{\text{separated}} \tag{27}$$

using `InterfaceAnalyzerMover` with the `beta_nov16` score function. Before evaluation, we locally use minimization on the interface for protein residues within $8\,\text{Å}$ of the ligand to reduce measurement noise. Lower ΔG indicates a more favorable predicted interface.

We map the energy to a bounded higher-is-better reward:

$$r_{\text{bind}} = \tanh\left(-\frac{\Delta G - c}{s}\right), \qquad c = 5, \qquad s = 10. \tag{28}$$

This transformation preserves the ordering of interface energies and limits the influence of extreme Rosetta values.

RF3 interface-confidence reward. A generated binder should not only have favorable energy in its generated conformation, but should also refold into a confident realization of the intended protein–ligand interface. We therefore use RF3 protein–ligand interface predicted aligned error as an additional reward signal.

For each rollout sample, we directly use the sequence generated by the base model together with the fixed ligand and refold the complex using RF3 (Corley et al., 2025), with $N_{\text{steps}} = 50$ and $n_{\text{recycle}} = 3$.

Let $\text{PAE}^{\min}_{c^\star j}$ denote the minimum predicted aligned error over atom or residue pairs spanning the designed chain $c^\star$ and partner chain j. We define the symmetrized minimum interface PAE as

$$\text{iPAE}_{\min} = \min_{j \neq c^\star} \frac{1}{2}\left(\text{PAE}^{\min}_{c^\star j} + \text{PAE}^{\min}_{j c^\star}\right). \tag{29}$$

Lower $\text{iPAE}_{\min}$ indicates greater confidence in the relative placement of the generated protein and ligand. We therefore define the higher-is-better reward as

$$r_{\text{RF3}} = -\text{iPAE}_{\min}. \tag{30}$$

Small-molecule binder reward combination. For binder post-training, we combine the binding-energy, RF3 confidence, and diversity rewards after within-group standardization:

$$R_i^{\text{binder}} = w_{\text{bind}}\widehat{r}_{\text{bind},i} + w_{\text{RF3}}\widehat{r}_{\text{RF3},i} + w_{\text{div}}\widehat{r}_{\text{div},i}, \tag{31}$$

where $\widehat{r}$ denotes reward standardization within the rollout group. The structural-validity penalty from Equation 21 is subsequently subtracted before the final advantage normalization.

E.3 ENZYME-DESIGN REWARDS

Catalytic-residue match reward. For enzyme design, the generated protein must recover the prescribed identities of the catalytic residues. Let $\mathcal{T}_{\text{cat}}$ denote the set of catalytic motif positions,

with generated residue identity $\widehat{a}_k$ and prescribed residue identity $a_k^\star$ at catalytic position k. We define

$$r_{\text{cat_id}} = \frac{1}{|\mathcal{T}_{\text{cat}}|} \sum_{k \in \mathcal{T}_{\text{cat}}} \mathbf{1}\left[\widehat{a}_k = a_k^\star\right]. \tag{32}$$

Thus, $r_{\text{cat_id}} \in [0, 1]$ is the fraction of prescribed catalytic residue identities recovered by the generated structure. A value of one indicates an exact catalytic-residue match.

Catalytic sites and backbone RMSD rewards. We directly optimize the raw geometric metrics produced by the enzyme-design evaluation pipeline, without additional nonlinear shaping. Let d_{cat} denote the catalytic functional groups RMSD relative to the prescribed active-site configuration, and let d_{bb} denote the backbone RMSD of the generated catalytic motif relative to its target geometry.

Because lower RMSD values indicate more accurate active-site recovery, we simply reverse their signs:

$$r_{\text{cat_rmsd}} = -d_{\text{cat_rmsd}}, \qquad r_{\text{bb}} = -d_{\text{bb}}. \tag{33}$$

This preserves the exact ranking induced by the underlying RMSD metrics and provides a direct optimization signal for ligand placement and catalytic-site backbone geometry.

Enzyme reward combination. For enzyme-design post-training, we combine catalytic-residue recovery, catalytic-residue RMSD, and backbone RMSD after within-group standardization:

$$R_i^{\text{enzyme}} = w_{\text{cat_id}} \widehat{r}_{\text{cat_id},i} + w_{\text{cat_rmsd}} \widehat{r}_{\text{cat_rmsd},i} + w_{\text{bb}} \widehat{r}_{\text{bb},i} + w_{\text{div}} \widehat{r}_{\text{div},i}. \tag{34}$$

The same general structural-validity penalty used for binder design, including the ligand-clash term, is then subtracted before final group-relative normalization. The diversity reward may additionally be included as a lightly weighted regularizer when specified by the experiment.

E.4 Reward Aggregation and Group-Relative Advantage

Let $r_{k,i}$ denote task reward head k for rollout sample i in a group of size G. Each reward head is standardized within the group:

$$\widehat{r}_{k,i} = \frac{r_{k,i} - \mu_k}{\sigma_k + \epsilon}, \qquad \mu_k = \frac{1}{G} \sum_{j=1}^{G} r_{k,j}, \qquad \sigma_k = \text{std}_j\left(r_{k,j}\right). \tag{35}$$

This normalization places heterogeneous metrics, such as Rosetta energy, interface PAE, catalytic match, and RMSD, on comparable within-group scales.

Let

$$R_i = \sum_k w_k \widehat{r}_{k,i} \tag{36}$$

denote the weighted sum of standardized task rewards. The structural-validity penalty is normalized by its within-group standard deviation without mean-centering:

$$\widetilde{R}_i = R_i - \frac{p_i}{\sigma_p + \epsilon}, \qquad \sigma_p = \text{std}_j(p_j). \tag{37}$$

Not mean-centering the penalty preserves the interpretation that valid structures receive no geometric deduction, while invalid structures are only demoted.

Finally, the penalty-adjusted score is standardized within the rollout group, clipped to obtain the final group-relative advantage, then recentered:

$$A_i = \text{recenter}\left[\text{clip}\left(\frac{\widetilde{R}_i - \mu_{\widetilde{R}}}{\sigma_{\widetilde{R}} + \epsilon}, -5, 5\right)\right], \tag{38}$$

where

$$\mu_{\widetilde{R}} = \frac{1}{G} \sum_{j=1}^{G} \widetilde{R}_j, \qquad \sigma_{\widetilde{R}} = \text{std}_j\left(\widetilde{R}_j\right). \tag{39}$$

This two-stage construction first balances task rewards with different units and then converts their aggregate into a condition-specific relative preference signal. Applying the geometric penalty before the final normalization ensures that structurally invalid samples are consistently demoted relative to valid candidates generated under the same condition.

Table 3: Distribution of native protein–ligand contact statistics over $N = 13{,}528$ training complexes and the fraction of native complexes satisfying each calibrated threshold. The thresholds are chosen near the tails of the native distributions so that genuine binders are largely retained while non-contacting designs are rejected.

Quantity	Unit	Min	p_5	p_{10}	Median	p_{90}	Max	Threshold	Native pass
$d_{\min}$	Å	0.14	1.61	1.80	2.60	2.95	4.45	< 4.5	100.0%
d_{cen}	Å	0.04	0.55	0.75	2.12	3.95	17.19	< 8	99.8%
n_{c}	–	1	6	9	17	25	46	≥ 8	92.9%

F Derivation of the Contact-Gate Thresholds

The interface binding-energy reward is protected by a *contact gate* to ensure that the ligand is physically engaged by the generated protein before the Rosetta energy is evaluated. Without this gate, the optimizer can exploit a degenerate solution in which the generated protein moves away from the ligand, yielding a near-zero interface energy that can outrank poorly scored interacting candidates. Samples that fail the contact gate skip Rosetta reward evaluation and instead receive a fixed low binding reward, $r_{\text{bind}} = -1.2$, which is below the minimum value of the tanh-scaled binding reward, -1.

Concretely, a design passes the gate only if all three conditions hold:

$$\text{pass} = \big(d_{\min} < d^*_{\min}\big) \wedge \big(d_{\text{cen}} < d^*_{\text{cen}}\big) \wedge \big(n_{\text{c}} \geq n^*_{\text{c}}\big), \tag{40}$$

where $d_{\min}$ is the minimum protein–ligand heavy-atom distance, d_{cen} is the distance between the ligand centroid and the centroid of the predicted pocket residues, and n_{c} is the number of protein residues with at least one heavy atom within 5 Å of the ligand.

We set the contact-gate thresholds to

$$(d^*_{\min}, d^*_{\text{cen}}, n^*_{\text{c}}) = (4.5\,\text{Å}, 8\,\text{Å}, 8).$$

These values were calibrated from the empirical distribution of native protein–ligand contacts. Specifically, we computed $d_{\min}$, d_{cen}, and n_{c} for all $N = 13{,}528$ ground-truth complexes in the full Plinder dataset and chose thresholds near the tails of the native distributions. This calibration makes the gate permissive for genuine protein–ligand interfaces while rejecting non-contacting or weakly contacting generated structures. Table 3 summarizes the native distributions and the corresponding pass rates.

The two distance thresholds are intentionally loose. The maximum native $d_{\min}$ is 4.45 Å, so the 4.5 Å cutoff admits essentially all native complexes. Similarly, the 8 Å centroid-distance cutoff is well above the native 90th percentile of 3.95 Å and mainly serves as a sanity check against grossly displaced ligands. The most selective condition is the contact-count threshold. Requiring $n_{\text{c}} \geq 8$ lies near the lower tail of the native distribution, between $p_5 = 6$ and $p_{10} = 9$, and retains 92.9% of native pockets. This criterion penalizes designs that give up binding the ligand but remains permissive enough to avoid discarding most realistic binding-site geometries.

G Additional Method Details

This appendix provides implementation details omitted from Section 3. Specifically, we summarize the complete optimization procedure and report the implementation choices required to reproduce SARP, including repeated endpoint re-noising, modality-specific masking and loss normalization, the reference-policy update schedule, reward-guided beam-search implementation, reward-normalization edge cases, and default training hyperparameters.

G.1 Complete SARP Training Procedure

Algorithm 1 summarizes one optimization step of SARP. Rollout generation and reward-guided beam search are performed without gradient tracking. Only the detached final endpoints are freshly re-noised to construct the DGPO and Search-SFT objectives. Reference-policy updates are applied after each optimizer step following the schedule described in Appendix G.4.

Algorithm 1 SARP algorithm

Require: Pretrained policy π_θ, dataset $\mathcal{D}$, rollout group size G, re-noising samples M, search schedule $\lambda_{\mathrm{SFT}}(n)$

Initialize θ from pretrained model and set $\theta_{\mathrm{ref}} \leftarrow \theta$
for optimizer step $n = 1, \ldots, N$ **do**
  Sample a design condition $\mathcal{C} \sim \mathcal{D}$
  Generate rollout endpoints $\{\mathbf{x}_1^{(g)}\}_{g=1}^{G} \sim \pi_\theta(\cdot \mid \mathcal{C})$ from independent initial noise
  Evaluate task rewards and structural-validity penalties
  Compute group-relative advantages
  Set $\mathcal{L}_{\mathrm{DGPO}} \leftarrow 0$ and $\mathcal{L}_{\mathrm{ref}} \leftarrow 0$
  for $s = 1, \ldots, M$ **do**
    Sample $t_s \sim \mathcal{U}[0.1, 1.0]$
    Re-noise all endpoints using modality-specific corruption shared across group members
    Compute online and reference DSM losses for the backbone and latent flows
    Compute detached modality-specific denoising gaps
    Compute a detached group-level sigmoid weight from the advantage-weighted gaps
    Accumulate the weighted online DSM loss into $\mathcal{L}_{\mathrm{DGPO}}$
    Accumulate velocity-field regularization into $\mathcal{L}_{\mathrm{ref}}$
  end for
  Average $\mathcal{L}_{\mathrm{DGPO}}$ and $\mathcal{L}_{\mathrm{ref}}$ over the M re-noising samples
  if $\lambda_{\mathrm{SFT}}(n) > 0$ **then**
    Run reward-guided beam search under condition $\mathcal{C}$
    Retain searched endpoints satisfying the experiment-specific acceptance criterion
    Re-noise accepted endpoints and compute $\mathcal{L}_{\mathrm{SFT}}$; set it to zero if none are accepted
  else
    Set $\mathcal{L}_{\mathrm{SFT}} \leftarrow 0$
  end if
  $\mathcal{L}_{\mathrm{SARP}} \leftarrow \mathcal{L}_{\mathrm{DGPO}} + \beta_{\mathrm{ref}}\mathcal{L}_{\mathrm{ref}} + \lambda_{\mathrm{SFT}}(n)\mathcal{L}_{\mathrm{SFT}}$
  Update θ using $\nabla_\theta \mathcal{L}_{\mathrm{SARP}}$
  if $n = N_{\mathrm{warm}}$ **then**
    Re-anchor the reference policy: $\theta_{\mathrm{ref}} \leftarrow \theta$
  else if $n > N_{\mathrm{warm}}$ **then**
    Update $\theta_{\mathrm{ref}} \leftarrow \alpha\theta_{\mathrm{ref}} + (1 - \alpha)\theta$
  end if
end for

G.2 Endpoint Re-noising Details

For each rollout group, the DGPO and reference objectives are evaluated on $M = 8$ independently sampled flow times

$$t_s \sim \mathcal{U}[0.1, 1.0), \qquad s = 1, \ldots, M.$$

For every sampled time and modality, fresh prior states are constructed using the same noise sampling, centering, and masking operations as the pretrained Proteína-Complexa flow model. The same corruption realization is shared across all rollout-group members for a fixed flow time and modality, while backbone and latent priors are sampled independently in their respective representation spaces.

Self-conditioning. DGPO denoising uses self-conditioning with probability 0.5. A detached prediction of the clean endpoint is optionally fed back into the final velocity prediction.

G.3 Modality-Specific Loss Normalization

The masked denoising loss for modality m is normalized by both the number of valid variables and the feature dimension,

$$\ell_\theta^{(g,m)} = \frac{\left\| M_g^{(m)} \odot \left[\mathbf{v}_\theta^m(\mathbf{x}_t^{(g)}, t, \mathcal{C}) - \mathbf{u}^{(g,m)} \right] \right\|_2^2}{\max\left[(\sum_i M_{g,i}^{(m)}) d_m, 1 \right]}. \tag{41}$$

The backbone loss is evaluated over generated, non-padded residues,

$$M_g^{(\mathrm{bb})} = M_g^{(\mathrm{pad})},$$

whereas the latent loss additionally excludes residues unsupported by the latent encoder,

$$M_g^{(\mathrm{lat})} = M_g^{(\mathrm{pad})} \odot M_g^{(\mathrm{enc})}.$$

Both the online–reference denoising gaps and the resulting DGPO sigmoid weight are detached from the computation graph, so gradients flow only through the online denoising losses. Unless otherwise stated, we use $\beta_{\mathrm{dpo}} = 50$.

G.4 Reference-Policy Schedule

The reference policy is initialized from the pretrained base model checkpoint and remains fixed for the first $N_{\mathrm{warm}} = 20$ optimization steps. At the end of the warm-up stage, the reference is re-anchored to the current online policy and is subsequently updated after every optimization step using an exponential moving average,

$$\theta_{\mathrm{ref}} \leftarrow \alpha\theta_{\mathrm{ref}} + (1-\alpha)\theta, \qquad \alpha = 0.99. \tag{42}$$

Reference parameters are never updated by gradient descent and are always evaluated without gradients.

G.5 Reward-Guided Beam Search

Beam search follows the inference procedure of Proteína-Complexa (Didi et al., 2026). Let

$$0 = s_0 < s_1 < \cdots < s_L = 1$$

denote the predefined branch points. At each branch point, every retained beam state is expanded into n_{branch} stochastic continuations, which are integrated to the next branch point. Each partial trajectory is then completed to the final flow time using the standard stochastic sampler and evaluated with a task-specific search score. The highest-scoring partial states are retained for the next search stage.

Branching is performed jointly on the coupled backbone and latent representations. To combine multiple objectives, each raw metric is first linearly rescaled onto a common $[0, 10]$ interval using fixed task-specific bounds and then aggregated with fixed weights,

$$S = \sum_i w_i \, \mathrm{clip}\left(10\,\frac{x_i - l_i}{u_i - l_i},\, 0,\, 10 \right), \tag{43}$$

where x_i is the raw metric, $[l_i, u_i]$ specifies its rescaling range (reversed for lower-is-better metrics), and w_i is its search weight. Unlike DGPO rewards, which are normalized within each rollout group, these score transformations use fixed global ranges.

For small-molecule binder design, the shaped binding reward $r_{bind} \in [-1, 1]$ and RF3 min-iPAE reward $r_{RF3} \in [-30, 0]$ Å are linearly rescaled to $[0, 10]$, giving

$$S_{\text{binder}} = \text{clip}\left(10\frac{r_{bind} - (-1)}{1 - (-1)}, 0, 10\right) + \text{clip}\left(10\frac{30 - min_iPAE}{30}, 0, 10\right). \tag{44}$$

For enzyme design, the search score combines catalytic-residue recovery r_{cat_id}, backbone RMSD r_{bb}, and catalytic-motif all-atom RMSD r_{cat_rmsd}. Using weights $(1, 1, 2)$ and fixed rescaling ranges $[0, 1]$, $[4, 0]$ Å, and $[3, 0]$ Å, respectively, we use

$$S_{\text{enzyme}} = \text{clip}\big(10 * r_{cat_id},\, 0,\, 10\big) + \text{clip}\Big(10\,\frac{4 - d_{bb}}{4},\, 0,\, 10\Big) + 2\,\text{clip}\Big(10\,\frac{3 - d_{cat_rmsd}}{3},\, 0,\, 10\Big). \tag{45}$$

Beam-search trajectories are never differentiated through. Only accepted final endpoints are detached and used as Search-SFT targets, where intermediate beam states are never supervised directly.

G.6 Matching the Effective Endpoint-Supervision Dose

To construct a controlled comparison between search-derived SARP and SARP-rollout, we match the effective amount of endpoint supervision received by the two variants. A naive approach would match the number of supervised endpoints per optimization step. However, this does not match the effective SFT gradient in our implementation because per-sample SFT weights are normalized to sum to one within each distributed rank:

$$\mathcal{L}_{\text{SFT}}^{(r)} = \sum_{i=1}^{N_r} w_i^{(r)} \ell_i^{(r)}, \qquad \sum_{i=1}^{N_r} w_i^{(r)} = 1, \tag{46}$$

for any rank r with at least one accepted endpoint. Consequently, a rank supervising one endpoint and a rank supervising multiple endpoints contribute comparable loss scale, but a rank with no accepted endpoint contributes zero SFT gradient.

With data-parallel training over N_{rank} ranks, the resulting SFT gradient can be written as

$$g_{\text{SFT}} = \frac{1}{N_{rank}} \sum_{r=1}^{N_{rank}} g_r = \frac{n_{\text{fire}}}{N_{rank}}\, \bar{g}_{\text{fire}} = d\, \bar{g}_{\text{fire}}, \tag{47}$$

where n_{fire} is the number of ranks receiving at least one supervision target, $d = n_{\text{fire}}/N_{rank}$ is the firing duty cycle, and $\bar{g}_{\text{fire}}$ denotes the average gradient over firing ranks. Thus, the effective SFT gradient scale is primarily controlled by the fraction of firing rank-steps rather than by the raw number of accepted endpoints on each firing rank.

This distinction matters because search-derived SARP and SARP-Rollout have different natural firing behavior. Search-derived supervision is applied only when beam search produces an endpoint satisfying the rollout-relative acceptance criterion, so its firing duty cycle is generally below one and varies across targets. In contrast, selecting the top-k endpoints from the on-policy rollout group always produces supervision targets unless explicitly throttled. Without correction, SARP-Rollout would therefore receive a systematically larger supervised gradient.

We match the two variants using two quantities measured from the corresponding search-derived SARP run. First, we set the rollout supervision duty cycle to

$$d = \Pr(N_{\text{accept}} > 0)\,, \tag{48}$$

so that SARP-Rollout receives supervised updates at the same expected frequency as search-derived SARP. Second, we set the number of rollout endpoints supervised on a firing event to

$$k \approx \mathbb{E}\left[N_{\text{accept}} \mid N_{\text{accept}} > 0\right], \tag{49}$$

matching the typical number of supervision targets presented when search supervision fires. The duty cycle d controls the effective supervised-gradient magnitude, whereas k controls the breadth of the distilled endpoint set. Both quantities are measured separately for each target from the corresponding SARP trajectory rather than treated as global constants.

For example, on M0552, search-derived SARP fires on 82.9% of rank-steps during the matched supervision window, with a mean of 5.31 accepted endpoints conditional on firing. We therefore

Table 4: Default SARP training configuration.

Hyperparameter	**Value**
Local rollout group size G	32
Distributed training GPUs	8 NVIDIA L40
Learning rate	3×10^{-6}
Rollout sampler	Proteína-Complexa stochastic sampler
Standard rollout steps	400
Re-noising samples per group M	8
Re-noising time range	$[0.1, 1.0]$
DGPO self-conditioning probability	0.5
Advantage clipping range	$[-5, 5]$
DGPO scale β_{dpo}	50
Reference regularization weight β_{ref}	0.05
Fixed-reference warm-up N_{warm}	20 steps
EMA reference coefficient α	0.99

use $d = 0.829$ and $k = 5$ for the corresponding SARP-Rollout control. Across the target-specific enzyme experiments, the measured firing duty cycle varies substantially across targets. For M0157, M0375, M0552, M0732, and M0904, the measured firing duty cycles are 0.833, 0.375, 0.829, 0.787, and 0.750, respectively. The corresponding mean numbers of accepted endpoints conditional on firing are 5.79, 1.54, 5.31, 5.33, and 5.04, leading us to use rollout target counts k=6,2,5,5, and 5, respectively. The associated search beam widths are 8,2,8,8, and 8, with replace-k values 8,2,8,8, and 8. Without duty-cycle matching, an always-firing SARP-Rollout control would receive approximately 1.20×, 2.67×, 1.21×, 1.27×, and 1.33× the effective SFT gradient of the corresponding search-derived SARP runs.

G.7 Reward-Normalization Edge Cases

Only ordinary on-policy rollout samples participate in DGPO reward normalization. Search-generated samples are excluded from the computation of reward means, standard deviations, and group-relative advantages. If the within-group standard deviation of any reward component is below 10^{-6}, its normalized contribution is set to zero, preventing numerically unstable preference signals. Missing or unevaluable reward terms are excluded from the corresponding normalization statistics. If a rollout provides no valid reward signal, it contributes no DGPO or reference-regularization update.

G.8 Default Training Configuration

Table 4 summarizes the default hyperparameters shared across all experiments. Task-specific reward weights, search configurations, acceptance thresholds, search-supervision schedules, and training lengths are reported in the corresponding experimental sections.

All reported evaluations use the standard stochastic Proteína-Complexa sampler without beam search. Beam search is therefore used only during post-training to improve exploration and construct Search-SFT targets, and is not required during inference.

G.9 Training horizon and checkpoint selection

We use different checkpointing protocols for the two experimental settings because they serve different evaluation purposes. For small-molecule binder optimization, RL-based methods are trained under a fixed maximum horizon of 200 optimization steps to provide a common update budget for cross-method comparison. In contrast, each enzyme target is optimized independently and exhibits a different convergence rate. For DGPO and SARP, we select the checkpoint with the highest per-rollout training reward before a substantial diversity collapse, identified by a steep drop in the diversity reward. The duration of endpoint supervision in SARP is determined separately and is shortened once such a diversity decline is observed. Online Search-SFT has no RL reward signal,

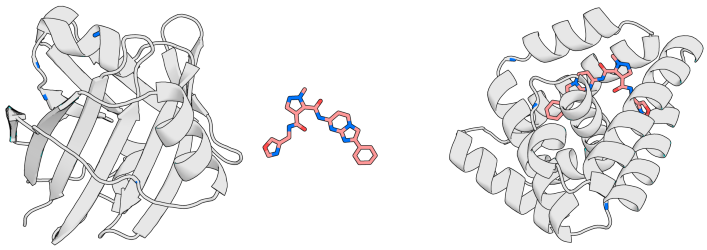

Figure 5: **Prevention of binding-energy reward hacking.** Without contact gating, a generated protein can remain separated from the ligand yet obtain a deceptively favorable separated-interface energy (left). Contact gating prevents this shortcut and promotes physically interacting protein–ligand complexes (right).

so its training is stopped once the supervised training loss has converged. Consequently, the enzyme training lengths vary across targets and methods and should be interpreted as target-specific selected checkpoints rather than compute-matched budgets. The final 100-generation evaluation in Table 2 is performed only after checkpoint selection and is not used to choose the reported checkpoint.

H Full Results and Additional Analysis for Small-Molecule Binder Design

The complete per-ligand results for small-molecule binder design are reported in Table 5, where for SARP beam search we set $n_{branches} = 4$ and beam width = 2. We additionally provide the qualitative analysis of binding-energy reward hacking in Figure 5.

Table 5: **Per-ligand comparison under the Binding Energy + RosettaFold3 min-iPAE reward.** RL-based methods are compared under a maximum training budget of 200 optimization steps. scRMSD denotes backbone Cα RMSD to the RF3 refold, ligRMSD denotes Cα-aligned ligand-pose RMSD, and diversity is measured by mean pairwise USalign TM-score (lower is more diverse).

Method	Lig.	Configuration		Evaluation					
		RL	Search Sup.	ΔG REU ↓	min-iPAE Å ↓	pLDDT ↑	scRMSD Å ↓	ligRMSD Å ↓	Div. meanTM ↓
Base model									
	FAD	–	–	10.40	4.07	73.44	1.87	10.57	0.379
	IAI	–	–	-9.09	9.41	61.53	6.11	11.46	0.327
	OQO	–	–	2.96	6.33	66.33	4.31	9.27	0.329
	SAM	–	–	3.18	3.46	76.08	2.16	8.16	0.368
GRPO									
	FAD	✓	–	4.24	4.00	74.29	2.15	10.51	0.362
	IAI	✓	–	-12.61	9.29	60.92	5.45	10.79	0.332
	OQO	✓	–	-5.90	6.36	65.62	5.79	10.45	0.333
	SAM	✓	–	0.36	4.10	76.49	1.70	7.97	0.366
DGPO									
	FAD	✓	–	1.64	3.42	77.33	1.46	10.72	0.495
	IAI	✓	–	-13.40	7.52	70.71	2.83	9.90	0.370
	OQO	✓	–	-7.02	4.05	76.17	1.76	8.09	0.352
	SAM	✓	–	1.69	3.09	78.69	1.42	9.77	0.516
Online Search-SFT (λ_{DGPO}=0)									
	FAD	–	✓	0.85	4.42	78.06	1.27	15.29	0.569
	IAI	–	✓	-9.45	5.58	79.44	1.19	12.21	0.604
	OQO	–	✓	-4.64	4.27	79.84	1.35	10.46	0.583
	SAM	–	✓	2.45	4.32	79.82	1.36	12.67	0.551

Continued on next page

Table 5: **Per-ligand comparison under the Binding Energy + RosettaFold3 min-iPAE reward** (continued).

Method	Lig.	Configuration		Evaluation					
		RL	Search Sup.	ΔG REU ↓	min-iPAE Å ↓	pLDDT ↑	scRMSD Å ↓	ligRMSD Å ↓	Div. meanTM ↓
SARP (full supervision)									
	FAD	✓	✓	-1.05	3.44	78.74	1.13	8.94	0.385
	IAI	✓	✓	-25.69	4.41	77.27	1.22	6.53	0.432
	OQO	✓	✓	-16.63	2.77	79.92	1.31	5.47	0.414
	SAM	✓	✓	-1.20	4.08	80.47	1.20	7.01	0.407
SARP (turn-off SFT @ step 90)									
	FAD	✓	✓	-1.83	3.88	78.23	1.26	8.36	0.396
	IAI	✓	✓	-26.64	4.51	76.79	1.39	6.67	0.449
	OQO	✓	✓	-18.41	2.43	79.44	1.34	5.53	0.422
	SAM	✓	✓	-3.88	6.28	79.73	1.24	8.02	0.418
SARP (turn-off SFT @ step 135)									
	FAD	✓	✓	-2.36	4.02	76.99	1.32	8.17	0.402
	IAI	✓	✓	-25.96	4.96	75.84	1.38	6.34	0.431
	OQO	✓	✓	-18.29	3.09	78.17	1.44	5.64	0.417
	SAM	✓	✓	-3.34	6.21	79.70	1.30	7.39	0.409

I Cross-Evaluator Validation with AlphaFold3

Our primary evaluations use RosettaFold3 (RF3)-derived confidence and self-consistency metrics, and RF3-based quantities also enter the post-training objectives for several tasks. This creates a potential concern that the observed improvements may partly reflect specialization to the structure predictor used during optimization. To assess whether the gains transfer across evaluators, we re-evaluate both the small-molecule binder and enzyme-design benchmarks using AlphaFold3 (AF3). AF3 is used strictly as an evaluation-only refolding model and never enters generation, search, sample selection, or post-training.

I.1 Small-molecule binder cross-evaluation

For the small-molecule binder benchmark, we retain exactly the same RF3-selected designs reported in Table 1. We then refold the fixed sequence–ligand pairs with AF3 and recompute min-iPAE, pLDDT, backbone self-consistency RMSD, and ligand RMSD. Because the selected designs themselves are unchanged, ΔG and structural diversity remain identical by construction. This protocol therefore isolates the effect of changing the evaluation model while holding both the candidate set and the RF3-based selection procedure fixed.

Table 6 shows that the improvements obtained by SARP persist under AF3 evaluation. The base model obtains an AF3 min-iPAE of 4.45 Å, scRMSD of 3.70 Å, and pLDDT of 76.65, while DGPO improves these quantities to 3.78 Å, 2.03 Å, and 82.86, respectively. The SARP variants further improve AF3-assessed interface confidence and structural self-consistency. In particular, SARP with Search-SFT turned off at step 90 achieves the best average AF3 min-iPAE (2.89 Å), ligRMSD (6.94 Å), and pLDDT (87.86), while SARP-rollout achieves the best scRMSD (1.23 Å), followed by full-supervision SARP at 1.30 Å.

Overall, the relative ordering of the strongest post-trained models is largely preserved under AF3, despite AF3 never appearing in the reward. This provides evidence that the improvements are not restricted to RF3-specific confidence or self-consistency scores.

I.2 Enzyme-design cross-evaluation

We perform an analogous cross-evaluator analysis for the target-specific enzyme-design benchmark. The generated designs, training checkpoints, generation budget, and AME criteria are kept identical to those in Table 2, where only the refolding model is changed from RF3 to AF3. Catalytic-residue identity matching and ligand-clash statistics are properties of the generated design and therefore remain unchanged. Catalytic-group RMSD, backbone RMSD, pass rate, and the set of passing structures are recomputed using AF3 refolds. Because the passing subset can change under the new

Table 6: **Cross-evaluator validation of RF3-selected small-molecule binder designs with AlphaFold3.** For each method and ligand, we evaluate exactly the same top 30% subset reported in Table 1, selected using the predefined criterion combining Rosetta ΔG and RF3 min-iPAE. The selected designs are refolded with AF3, from which min-iPAE, scRMSD, ligRMSD, and pLDDT are recomputed. ΔG and structural diversity are unchanged by construction. AF3 is used only for evaluation and never enters generation, search, selection, or post-training. Metrics are macro-averaged over FAD, IAI, OQO, and SAM. Best results are shown in **bold**, and second-best results are underlined.

Method	RL	Search Aug.	$\Delta G \downarrow$ REU	AF3 min-iPAE $\downarrow$ Å	AF3 scRMSD $\downarrow$ Å	AF3 ligRMSD $\downarrow$ Å	AF3 pLDDT $\uparrow$	Div. $\downarrow$ meanTM
Base model	–	–	1.86	4.45	3.70	9.89	76.65	0.351
GRPO	✓	–	-3.48	4.39	3.52	9.85	76.71	**0.348**
DGPO	✓	–	-4.27	3.78	2.03	10.14	82.86	0.433
Online Search-SFT	–	✓	-2.70	3.51	1.45	12.75	86.75	0.577
SARP-rollout (full supervision)	✓	–	-11.20	3.10	**1.23**	7.11	87.59	0.426
SARP (full supervision)	✓	✓	-11.15	3.08	1.30	7.25	87.49	0.410
SARP (turn-off @ 90)	✓	✓	**-12.69**	**2.89**	1.34	**6.94**	**87.86**	0.421
SARP (turn-off @ 135)	✓	✓	-12.49	2.91	1.49	7.07	87.38	0.415

refolding model, the Foldseek clustering used to compute the number of unique passing designs is also recomputed.

Table 7 shows that the main trends also transfer to AF3 in the enzyme-design setting. SARP achieves AF3 pass rates of 82%, 64%, 68%, 86%, and 25% on M0732, M0157, M0552, M0904, and M0375, respectively, compared with 10%, 18%, 5%, 10%, and 2% for the base model. SARP achieves the best catalytic-group and backbone RMSD on all five targets.

The improvement is particularly clear in the number of structurally distinct successful designs. SARP yields 39, 34, 47, 54, and 19 unique passing Foldseek clusters across the five targets, compared with 10, 16, 4, 6, and 2 for the base model. Relative to DGPO, SARP also produces more unique passing structures on four of the five targets: M0732 (39 vs. 38), M0157 (34 vs. 29), M0552 (47 vs. 38), and M0904 (54 vs. 43). On M0375, DGPO instead yields more unique passing structures (26 vs. 19), showing that the advantage has some target-specific variation.

Online Search-SFT alone performs substantially worse than SARP under AF3 evaluation, both in overall success and in the diversity of successful designs. On M0732, Online Search-SFT reaches an 18% pass rate with only 1 unique passing cluster, whereas SARP reaches 82% pass with 39 unique clusters. Similarly, on M0552, Online Search-SFT obtains 12% pass with only 1 unique cluster, compared with 68% pass and 47 unique clusters for SARP. Across the five targets, Online Search-SFT consistently produces only a small number of distinct successful structures, suggesting that search-derived endpoint supervision alone is insufficient to reproduce the broader success and diversity achieved by combining search supervision with on-policy DGPO.

M0375 shows a different trade-off from the other enzyme targets. Under AF3 evaluation, DGPO achieves the highest pass rate (26%) and the largest number of unique passing structures (26), while SARP reaches a similar 25% pass rate with 19 unique passing clusters and lower catalytic-site and backbone RMSD. Online Search-SFT performs substantially worse, with a 5% pass rate and only 1 unique passing cluster. Thus, on this target, DGPO yields more structurally distinct successful designs under AF3 evaluation, while SARP provides better geometric recovery at a comparable pass rate.

Taken together, the AF3 cross-evaluation preserves the main qualitative conclusions of the RF3-based benchmarks across both small-molecule binding and enzyme design. Since AF3 is never used during post-training, search, or sample selection, the observed improvements provide evidence that the learned gains transfer across structure predictors rather than arising solely from direct optimization of RF3-specific evaluation signals.

J Comparison with Inference-Time Search

Since SARP uses beam search during post-training, an important question is whether its benefit simply matches applying beam search at inference time or can instead be superior to pure search.

Table 7: **Cross-evaluator validation of target-specific enzyme-design optimization evaluated with AlphaFold3.** The generated designs, training checkpoints, generation budget, and AME criteria are identical to those in Table 2; only the refolding model is changed from RF3 to AF3. Cat. match and Lig. clash are unchanged, whereas Cat. RMSD, BB RMSD, Pass, and Unique pass are recomputed using AF3 refolds. Because the passing subset can change under AF3, Foldseek clustering is recomputed on the AF3-passing designs. AF3 is used only for evaluation and never enters generation, search, or post-training. Results use 100 generations per model. Unique pass denotes the number of Foldseek clusters at TM≥ 0.5 among passing designs. For SARP, Train #step is reported as $a(b)$, where a is the total number of post-training steps and b is the number of steps with auxiliary endpoint supervision enabled. Best results within each target are shown in **bold**, and second-best results are underlined.

Target	Method	Train #step	Cat. match ↑	Cat. RMSD ↓ Å	BB RMSD ↓ Å	Lig. clash ↓ %	Pass ↑ %	Unique pass ↑
M0732	Base model	–	0.998	3.32	1.98	0.008%	10%	10
	GRPO	104	0.994	2.90	1.89	0.009%	18%	15
	DGPO	100	0.990	1.78	1.20	0.021%	52%	38
	Online Search-SFT	30	0.984	2.16	1.08	**0.005%**	18%	1
	SARP	60(20)	**1.000**	**1.24**	**0.73**	0.006%	**82%**	**39**
M0157	Base model	–	0.999	2.57	1.43	**0.001%**	18%	16
	GRPO	122	0.999	2.03	0.83	0.004%	25%	23
	DGPO	170	**1.000**	1.75	0.64	**0.001%**	46%	29
	Online Search-SFT	20	0.997	2.47	0.65	0.003%	13%	8
	SARP	90(30)	**1.000**	**1.39**	**0.61**	0.003%	**64%**	**34**
M0552	Base model	–	0.998	2.98	1.42	0.004%	5%	4
	GRPO	74	0.998	2.89	1.23	**0.003%**	5%	5
	DGPO	120	0.998	1.73	0.93	0.015%	50%	38
	Online Search-SFT	40	0.997	2.82	2.06	0.011%	12%	1
	SARP	90(30)	**1.000**	**1.42**	**0.58**	**0.003%**	**68%**	**47**
M0904	Base model	–	**1.000**	2.62	1.29	**0.000%**	10%	6
	GRPO	126	0.999	2.39	1.68	0.021%	26%	20
	DGPO	120	0.999	1.60	0.94	0.005%	75%	43
	Online Search-SFT	40	**1.000**	2.06	0.85	0.003%	31%	2
	SARP	140(20)	**1.000**	**1.27**	**0.70**	**0.000%**	**86%**	**54**
M0375	Base model	–	0.999	3.05	1.15	0.011%	2%	2
	GRPO	94	**1.000**	2.98	0.98	0.005%	2%	2
	DGPO	110	**1.000**	2.13	1.03	0.006%	**26%**	**26**
	Online Search-SFT	50	**1.000**	2.83	1.02	**0.004%**	5%	1
	SARP	80(20)	**1.000**	**1.97**	**0.69**	0.008%	25%	19

We therefore compare the pretrained base model under standard sampling, the same checkpoint with inference-time beam search, and the SARP checkpoint under standard sampling without search on 100 designs per target.

As shown in Table 8, inference-time beam search substantially improves the pretrained model, confirming that search is effective for identifying higher-quality enzyme designs. SARP achieves comparable or better overall design quality under standard sampling, outperforming the best inference-time search in pass rate on M0732 and M0904 and consistently improves backbone recovery across all five targets. More importantly, SARP produces the largest number of distinct successful structural clusters on every target, indicating that post-training internalizes the benefits of search while preserving a substantially broader set of successful solutions rather than concentrating on a few high-reward structural basins. Moreover, these gains are achieved using standard sampling alone at inference, eliminating the need for beam search and reducing inference cost.

These results suggest that the benefits of search can be effectively incorporated into the model through post-training. Once trained, SARP especially recovers 1.2–3.3× more distinct successful clusters on every target with standard sampling alone.

K Effect of DGPO Rollout Group Size

We are also interested in investigating whether SARP's gains could be reproduced simply by increasing the number of on-policy samples available to DGPO. To examine this, we additionally train DGPO with rollout group size $G = 64$, compared with the default $G = 32$ setting.

Table 8: **Comparison with inference-time search on target-specific enzyme design.** All arms use a matched budget of ≈100 designs per target. The pretrained model and SARP sample directly. Inference Search applies beam search to the *same* pretrained checkpoint with branching factor 4 and beam width W, run for $\lfloor 100/W \rfloor$ independent trees. Because each stage evaluates $4W$ candidates and keeps W, the three search arms cost the same compute as one another, and $4\times$ the sampling and $16\times$ the oracle calls per design relative to no search. Pass is the fraction satisfying all AME criteria; Unique pass is the number of Foldseek clusters at TM≥ 0.5 among passing designs. Best values within each target are shown in **bold**.

Target	Method	Cat. match ↑	Cat. RMSD ↓ Å	BB RMSD ↓ Å	Lig. clash ↓	Pass ↑ %	Unique pass ↑ #
M0732	Base model	0.998	2.95	1.49	0.008%	20%	17
	Search (W=2)	**1.000**	1.51	0.72	**0.000%**	58%	23
	Search (W=4)	**1.000**	1.37	0.92	**0.000%**	87%	20
	Search (W=8)	**1.000**	1.26	0.84	0.003%	77%	11
	SARP	**1.000**	**1.11**	**0.58**	0.006%	**89%**	**44**
M0157	Base model	0.999	2.74	1.77	0.001%	18%	17
	Search (W=2)	**1.000**	1.59	0.90	**0.000%**	61%	36
	Search (W=4)	**1.000**	1.31	0.78	0.003%	85%	25
	Search (W=8)	**1.000**	**1.28**	0.68	0.001%	**90%**	18
	SARP	**1.000**	1.34	**0.58**	0.003%	74%	**43**
M0552	Base model	0.998	3.09	1.52	0.004%	11%	7
	Search (W=2)	**1.000**	1.89	0.82	0.005%	36%	21
	Search (W=4)	**1.000**	1.62	0.67	0.001%	61%	14
	Search (W=8)	**1.000**	1.39	0.63	**0.000%**	**80%**	13
	SARP	**1.000**	**1.36**	**0.57**	0.003%	72%	**54**
M0904	Base model	**1.000**	2.39	1.28	**0.000%**	21%	12
	Search (W=2)	**1.000**	1.43	0.77	0.003%	68%	16
	Search (W=4)	**1.000**	1.32	0.72	0.014%	65%	11
	Search (W=8)	**1.000**	1.31	0.86	**0.000%**	86%	10
	SARP	**1.000**	**1.20**	**0.60**	**0.000%**	**88%**	**52**
M0375	Base model	0.999	3.18	1.49	0.011%	5%	3
	Search (W=2)	0.997	2.06	0.72	0.003%	16%	9
	Search (W=4)	**1.000**	1.98	1.02	**0.000%**	25%	8
	Search (W=8)	**1.000**	**1.72**	0.65	0.001%	**30%**	6
	SARP	**1.000**	1.91	**0.63**	0.008%	24%	**18**

Table 9 reports the results across the five enzyme-design targets. The checkpoint-selection criteria are the same as what we describe in Section 4.2 Experimental Setup part. Increasing the DGPO group size does not consistently improve performance: $G = 64$ improves some targets and metrics, but yields an average pass rate of 45.6% and 27.8 unique passing clusters, compared with 54.4% and 35.6 for $G = 32$. SARP achieves 69.4% average pass rate and 42.2 unique passing clusters, indicating that its gains are not reproduced by simply enlarging the ordinary DGPO rollout group.

We observe a similar trend for small-molecule binder design. Table 10 compares the two DGPO rollout group sizes with SARP under the same evaluation protocol as Table 1. Increasing the group size from $G = 32$ to $G = 64$ substantially improves the binding-energy objective, from $\Delta G = -4.27$ to -9.64 REU, but does not yield consistent improvements across the remaining metrics. In particular, SARP still achieves stronger binding energy (-11.15 REU), lower scRMSD and ligand RMSD, higher pLDDT, and greater structural diversity. Together with the enzyme-design results, these results suggest that the benefit of SARP cannot be explained solely by exposing DGPO to a larger number of on-policy rollouts.

Table 9: **Effect of DGPO rollout group size on target-specific enzyme-design optimization.** Results use 100 standard generations per model. Pass is the fraction satisfying all AME criteria; Unique pass is the number of Foldseek clusters at TM$\geq$ 0.5 among passing designs. For SARP, training steps are reported as $a(b)$, where a is the total number of post-training steps and b is the number of steps with auxiliary endpoint supervision enabled. Best results within each target are shown in **bold**, and second-best results are underlined.

Target	Method	Training steps	Cat. match ↑	Cat. RMSD ↓ Å	BB RMSD ↓ Å	Lig. clash ↓ %	Pass ↑ %	Unique pass ↑
M0732	DGPO ($G = 32$)	100	0.990	1.53	0.97	0.021%	62%	42
	DGPO ($G = 64$)	60	0.998	1.24	0.61	0.031%	64%	34
	SARP	60(20)	**1.000**	**1.11**	**0.58**	**0.006%**	**89%**	**44**
M0157	DGPO ($G = 32$)	170	**1.000**	1.66	0.63	**0.001%**	57%	33
	DGPO ($G = 64$)	90	**1.000**	1.50	0.68	**0.001%**	49%	42
	SARP	90(30)	**1.000**	**1.34**	**0.58**	0.003%	**74%**	**43**
M0552	DGPO ($G = 32$)	120	0.998	1.62	0.74	0.015%	60%	40
	DGPO ($G = 64$)	60	**1.000**	1.54	0.58	**0.001%**	46%	40
	SARP	90(30)	**1.000**	**1.36**	**0.57**	0.003%	**72%**	**54**
M0904	DGPO ($G = 32$)	120	0.999	1.61	0.99	0.005%	75%	47
	DGPO ($G = 64$)	60	**1.000**	1.44	0.66	**0.000%**	57%	13
	SARP	140(20)	**1.000**	**1.20**	**0.60**	**0.000%**	**88%**	**52**
M0375	DGPO ($G = 32$)	110	**1.000**	2.26	1.15	0.006%	18%	16
	DGPO ($G = 64$)	50	**1.000**	1.98	0.68	**0.000%**	12%	10
	SARP	80(20)	**1.000**	**1.91**	**0.63**	0.008%	**24%**	**18**

Table 10: **Effect of DGPO rollout group size on small-molecule binder design.** For each method and ligand, 200 designs are generated and ranked using the same predefined criterion combining ΔG and RF3 min-iPAE. Metrics are macro-averaged over FAD, IAI, OQO, and SAM on the top 30% of ranked designs. Best results are shown in **bold**, and second-best results are underlined.

Method	ΔG ↓ REU	min-iPAE ↓ Å	scRMSD ↓ Å	ligRMSD ↓ Å	pLDDT ↑	Div. ↓ meanTM
DGPO ($G = 32$)	-4.27	4.52	1.87	9.62	75.72	0.433
DGPO ($G = 64$, recentred)	-9.64	5.36	1.92	9.62	72.14	0.471
SARP (full supervision)	**-11.15**	**3.67**	**1.22**	**6.99**	**79.10**	**0.410**

L Preference-Learning Dynamics in DGPO

DGPO replaces explicit trajectory likelihood ratios with online-reference denoising-loss differences, so we cannot inspect a conventional policy ratio directly. Instead, we verify whether the update moves the policy in the intended preference direction using two group-level diagnostics.

As also described in Section 3.1.3, for each rollout sample g, we define the denoising gap

$$\Delta^{(g)} = \mathrm{DSM}_{\theta}^{(g)} - \mathrm{DSM}_{\theta_{\mathrm{ref}}}^{(g)},$$

where both losses are evaluated on the same re-noised final sample and target velocity. A negative $\Delta^{(g)}$ means that the online policy fits sample g better than the reference, indicating increased support for that sample. Therefore, a correctly functioning DGPO update should make $\Delta^{(g)}$ more negative for high-advantage samples and less negative, or positive, for low-advantage samples.

We measure this behavior in two ways. First, we compute the within-group correlation $\mathrm{corr}(A^{(g)}, \Delta^{(g)})$. A negative value indicates that higher-reward samples are preferentially fit better than the reference. Second, we split samples by the sign of their advantage and compute the mean denoising gap in each group. Specifically, δ_{pos} averages $\Delta^{(g)}$ over positive-advantage samples, $A^{(g)} > 0$ and δ_{neg} averages $\Delta^{(g)}$ over negative-advantage samples, $A^{(g)} < 0$. Since a more negative gap indicates that the online policy fits a sample better than the frozen reference, a successful preference update should make δ_{pos} more negative than δ_{neg}. We therefore report the preference margin $\delta_{\mathrm{neg}} - \delta_{\mathrm{pos}}$, where larger positive values indicate stronger separation between the policy's fit to high-reward and low-reward samples.

Figure 6 shows that both diagnostics move in the expected direction during training: the advantage–gap correlation becomes negative, and the preference margin becomes positive. This indicates that

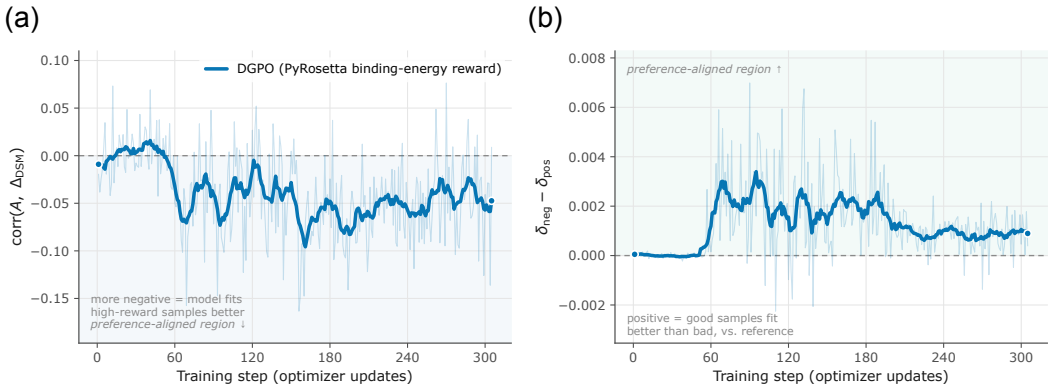

Figure 6: **Internal diagnostics of DGPO preference learning.** **(a)** Correlation between group-relative advantage $A^{(g)}$ and denoising gap $\Delta^{(g)} = \mathrm{DSM}_{\theta}^{(g)} - \mathrm{DSM}_{\theta_{\mathrm{ref}}}^{(g)}$ during training. More negative values indicate stronger preferential fitting of high-advantage samples. **(b)** Preference margin $\delta_{\mathrm{neg}} - \delta_{\mathrm{pos}}$, where positive values indicate that high-advantage samples are fit better than low-advantage samples relative to the reference. Faint lines show per-step values, bold lines show 11-step moving averages, and dashed lines mark zero.

the DGPO update learns the intended reward ordering rather than inheriting it from initialization. The effect is modest in magnitude, as expected under group-normalized advantages, but its direction is consistent with the desired preference gradient.

DGPO group-weight behavior. We additionally inspect the group-level sigmoid modulation inherited from DGPO. For a rollout group of size G, the DGPO group weight is

$$w_{\mathrm{DGPO}} = \sigma(z),$$

with sigmoid logit

$$z = \frac{\beta_{\mathrm{dpo}}}{G} \sum_{g=1}^{G} \sum_{m \in \{\mathrm{bb},\mathrm{lat}\}} A_m^{(g)} \left(\ell_{\theta}^{(g,m)} - \ell_{\mathrm{ref}}^{(g,m)} \right).$$

In our experiments, the backbone and latent flows use the same group-relative advantage, $A_{\mathrm{bb}}^{(g)} = A_{\mathrm{lat}}^{(g)} = A^{(g)}$. Defining the aggregated online–reference denoising-loss gap for rollout g as

$$\Delta\ell^{(g)} = \sum_{m \in \{\mathrm{bb},\mathrm{lat}\}} \left(\ell_{\theta}^{(g,m)} - \ell_{\mathrm{ref}}^{(g,m)} \right),$$

the logit becomes

$$z = \frac{\beta_{\mathrm{dpo}}}{G} \sum_{g=1}^{G} A^{(g)} \Delta\ell^{(g)}.$$

Because the advantages are group-centered, $\frac{1}{G}\sum_g A^{(g)} = 0$, this quantity is exactly the population covariance between the group advantages and denoising-loss gaps:

$$z = \beta_{\mathrm{dpo}} \, \mathrm{Cov}(A, \Delta\ell) = \beta_{\mathrm{dpo}} \, \rho_{A,\Delta\ell} \, \sigma_A \, \sigma_{\Delta\ell},$$

where $\rho_{A,\Delta\ell}$ is the within-group correlation between advantage and denoising-loss gap, and σ_A and $\sigma_{\Delta\ell}$ are their within-group standard deviations.

At $\beta_{\mathrm{dpo}} = 50$, σ_A is near unity by construction, while the measured denoising-gap scale is small, with $\sigma_{\Delta\ell} \approx 0.02$. Thus, the logit magnitude is largely governed by the correlation $\rho_{A,\Delta\ell}$ and remains modest in practice. Over the full run, the group weight remains at

$$w_{\mathrm{DGPO}} = 0.472 \pm 0.048,$$

with 92% of values in $[0.40, 0.60]$ and none below 0.24 or above 0.67. Thus, in this operating regime, the sigmoid remains near its central region and acts primarily as a modest rescaling of the DGPO term rather than as a hard gate.

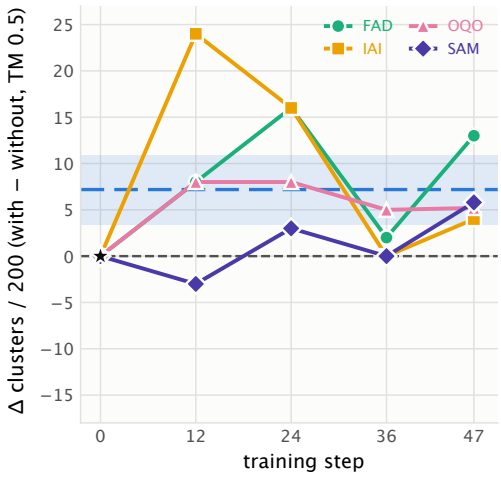

Figure 7: **Effect of the structural-diversity reward on Foldseek cluster count.** For each ligand and checkpoint, we report the difference $\Delta = \text{clusters}_{w=0.1} - \text{clusters}_{w=0.0}$ at a Foldseek clustering threshold of TM $= 0.5$. Positive values indicate that the diversity-reward model produces more distinct structural clusters. Step 0 is the shared pre-RL checkpoint and therefore has $\Delta = 0$ by construction. The dashed line and shaded band show the mean difference across all ligand–checkpoint pairs and its 95% confidence interval.

M Effect of the Structural-Diversity Reward

We compare two SARP runs trained with identical small molecule binder design settings except for the structural-diversity reward weight, using $w_{\text{div}} = 0.1$ for the reward run and $w_{\text{div}} = 0$ for the control. For each of the four benchmark ligands (FAD, IAI, OQO, and SAM) and four training checkpoints, we generate 200 designs per model and cluster the resulting structures using `foldseek easy-cluster` at TM $= 0.5$. Diversity is measured by the number of structural clusters, and we analyze the paired cluster-count difference between the two runs.

As shown in Figure 7, the diversity reward increases the number of structural clusters in 13 of the 16 ligand–checkpoint comparisons, with two ties and only one comparison favoring the control. On average, it produces 7.2 additional clusters per 200 designs. The diversity-reward model generally retains more structural clusters than the control throughout training, although the gap does not consistently increase over optimization. This suggests that the fixed diversity weight provides a stable regularizing effect, but may not fully adapt to changes in the policy distribution during training. Future work could explore adaptive or stage-dependent diversity weighting, for example strengthening the reward when structural collapse is detected and reducing it when it begins to compete with task-specific optimization.

N Gradient Decomposition of Joint Post-Training

We analyze how search supervision, reward optimization, and moving-reference regularization shape the two output flows of the dual-representation model: an explicit Cα backbone flow and a non-Cα latent flow. The analyzed small-molecule binder run uses beam-search augmentation with $\lambda_{\text{SFT}} = 0.05$, $n_{\text{branches}} = 4$, and beam width $W = 8$. The policy reward combines binding energy, RF3 interface min-iPAE, diversity, and a structural-validity penalty.

We decompose the training update along three complementary axes: (1) the relative magnitude and direction of the Search-SFT and DGPO parameter gradients; (2) the relative effects of the DGPO policy and moving-reference terms on the Cα backbone and non-Cα latent flows; and (3) the contribution of individual reward terms to the two flows. We then relate these gradient-level measurements to observable changes in sequence composition and Cα geometry along the same $W = 8$ lineage.

N.1 Gradient-Decomposition Probe

At the first diffusion timestep of the first inner epoch of each training step, we snapshot the accumulated parameter gradients between successive backward calls. Differences between snapshots isolate the parameter-gradient vectors associated with Search-SFT and DGPO:

$$g_{\mathrm{SFT}} = \nabla_\theta \left(\lambda_{\mathrm{SFT}} \mathcal{L}_{\mathrm{SFT}}\right), \qquad g_{\mathrm{DGPO}} = \nabla_\theta \mathcal{L}_{\mathrm{DGPO}}. \tag{50}$$

We additionally compute gradients with respect to the two output velocity fields: the explicit Cα backbone flow (`bb_ca`) and the non-Cα latent flow (`local_latents`). Because the two outputs contain different numbers of scalar dimensions, their raw L2 norms are not directly comparable. For each flow m, we therefore report the per-degree-of-freedom RMS gradient magnitude

$$\widetilde{G}_m = \frac{\|g_m\|_2}{\sqrt{D_m}} = \sqrt{\frac{1}{D_m} \sum_{j=1}^{D_m} g_{m,j}^2}, \tag{51}$$

where D_m is the number of valid scalar outputs in that flow. We then report the scale-corrected latent-to-backbone ratio

$$R_{\text{non-C}\alpha/\mathrm{bb}} = \frac{\widetilde{G}_{\text{non-C}\alpha}}{\widetilde{G}_{\mathrm{backbone}}}. \tag{52}$$

Values above one indicate a stronger average gradient per non-Cα latent dimension, whereas values below one indicate a stronger average gradient per Cα backbone coordinate.

N.2 Search-SFT and DGPO Become Increasingly Misaligned

Search supervision contributes a substantial parameter-space update throughout the probed run. With $\lambda_{\mathrm{SFT}} = 0.05$, the weighted Search-SFT gradient norm is on average $0.74\times$ the DGPO gradient norm. However, the two update directions become progressively less aligned over training: the mean gradient cosine decreases from -0.004 over steps 0–19 to -0.119 over steps 61–80.

To quantify the magnitude of this disagreement relative to the DGPO update, we project the Search-SFT gradient onto the DGPO direction. The fraction of the DGPO-directed component opposed by Search-SFT increases from approximately 0.6% over steps 0–19 to 12% over steps 61–80. Thus, although the two gradients remain largely near-orthogonal, Search-SFT increasingly contains a component opposing the on-policy reinforcement direction as training progresses.

These measurements suggest that the role of search supervision changes over the course of post-training: early Search-SFT updates are nearly orthogonal to DGPO, whereas later updates increasingly depart from the reinforcement direction. This offers a plausible optimization-level mechanism for the empirical sensitivity to the duration of search supervision under strong beam search.

N.3 Policy and Reference Regularization Prefer Different Flows

The output-level gradient decomposition reveals a relative asymmetry between the DGPO policy objective and moving-reference regularization. The policy gradient has a mean per-DOF ratio

$$R^{\mathrm{policy}}_{\text{non-C}\alpha/\mathrm{bb}} \approx 1.50, \tag{53}$$

and is non-Cα-latent dominant on 60 of the 81 densely probed training steps. Thus, after correcting for output dimensionality, the average policy-gradient magnitude per non-Cα latent degree of freedom is approximately $1.5\times$ that of a Cα backbone coordinate.

In contrast, the moving-reference component has

$$R^{\mathrm{ref}}_{\text{non-C}\alpha/\mathrm{bb}} \approx 0.72, \tag{54}$$

indicating relatively stronger per-DOF gradients on the explicit Cα backbone output.

The two objectives therefore distribute optimization pressure differently across the coupled representation. This distinction is relative rather than exclusive: both objectives produce nonzero gradients on both flows. Moreover, these ratios characterize where optimization pressure is applied and should not be interpreted as directly predicting proportional sequence-versus-backbone displacement in generated samples.

Table 11: **Decomposition of the DGPO policy gradient by objective term and output flow.** Backbone and non-Cα latent entries are L2 norms of the corresponding term-specific contribution vectors. Per-DOF ratios above one indicate a non-Cα latent bias.

Objective term	Cα backbone	Non-Cα latent	Per-DOF latent/bb	Approx. share
Binding energy ($w = 1$)	1.29e-4	2.81e-4	1.53	30%
RF3 min-iPAE ($w = 1$)	1.34e-4	2.82e-4	1.49	30%
Structural-validity penalty	1.29e-4	3.04e-4	1.66	31%
Diversity ($w = 0.1$)	1.3e-5	2.8e-5	1.50	3%

N.4 Reward and Structural-Validity Terms Co-Drive the Policy Update

We next decompose the DGPO policy gradient into contributions from the binding-energy reward, RF3 interface min-iPAE reward, structural-validity penalty, and diversity reward. Table 11 reports the L2 norm of each term-specific contribution vector for the Cα backbone and non-Cα latent outputs. The per-DOF ratio corrects for the different dimensionalities of the two outputs.

Binding energy, RF3 interface min-iPAE, and the structural-validity penalty contribute comparable gradient magnitudes and together dominate the decomposed policy update. All three are relatively latent-biased after per-DOF normalization:

$$R^{\text{bind}}_{\text{non-C}\alpha/\text{bb}} = 1.53, \qquad R^{\text{RF3}}_{\text{non-C}\alpha/\text{bb}} = 1.49, \qquad R^{\text{valid}}_{\text{non-C}\alpha/\text{bb}} = 1.66. \tag{55}$$

The structural-validity penalty exhibits the strongest latent bias and contributes the largest individual share of the decomposed policy gradient (31%). RF3 min-iPAE has the largest absolute Cα backbone-gradient component among the major reward terms, although it remains relatively latent-biased after per-DOF normalization. The diversity reward exhibits a similar latent preference but contributes only approximately 3% under its current weight.

These gradient-level decompositions characterize how optimization pressure is distributed across the coupled flows, but do not by themselves determine the resulting sequence and backbone displacement. We examine these output-level changes next.

N.5 Output-Level Changes Along the Instrumented $W = 8$ Lineage

We complement the gradient decomposition with measurements of amino-acid composition and Cα geometry along the same $W = 8$ lineage. Feature shifts are standardized by the pretrained model's own standard deviation, computed separately for each ligand and then averaged across the four small-molecule binder targets. This normalization measures how far each feature moves relative to variability already present in the pretrained distribution.

Sequence composition provides only one observable projection of the non-Cα latent flow, which also contains side-chain and other atomistic information. We therefore do not interpret sequence-versus-Cα feature displacement as a direct measurement of the latent/backbone gradient ratio.

Table 12 shows the evolution of the $W = 8$ lineage. Sequence-associated changes become more pronounced over training. At step 47, within the gradient-probed window, alanine frequency is shifted by $+2.38$ pretrained-model standard deviations and sequence entropy by -3.63. At the later checkpoint, the corresponding shifts increase to $+3.48$ for alanine frequency, -4.35 for sequence entropy, and $+3.09$ for hydrophobic-residue fraction. In raw composition, mean alanine frequency increases from 4.5% in the pretrained model to 11.9% at the later $W = 8$ checkpoint. Figure 8 further shows that this shift reflects a broader redistribution of amino-acid usage rather than an isolated change in alanine frequency.

The Cα backbone also undergoes substantial reorganization. At the later checkpoint, the fraction of long-range contacts decreases by 1.00 standard deviations and relative contact order by 0.89, while the fraction of Cα pairs separated by 8–10 Å increases by 1.55. In contrast, radius of gyration changes by only -0.04 standard deviations. Thus, late in training the overall structural scale is nearly preserved while the internal contact topology shifts toward more local packing.

Table 12: **Output-level changes along the instrumented $W = 8$ SARP lineage.** Values report shifts relative to the pretrained model in units of the pretrained-model standard deviation, computed per ligand and averaged across the four small-molecule binder targets. Step 47 lies within the gradient-probed window, whereas step 177 is the later checkpoint used for the output-level comparison.

Feature	W=8 **@47**	W=8 **@177**
Long-range contact fraction	-0.26	-1.00
Relative contact order	-0.61	-0.89
Cα distance 6–8 Å	+1.68	+1.32
Cα distance 8–10 Å	+1.47	+1.55
Cα distance $>$ 20 Å	-1.32	-0.75
Radius of gyration	-0.92	-0.04
Alanine fraction	+2.38	+3.48
Sequence entropy	-3.63	-4.35
Hydrophobic fraction	+1.27	+3.09
Helix fraction	+1.04	+1.82

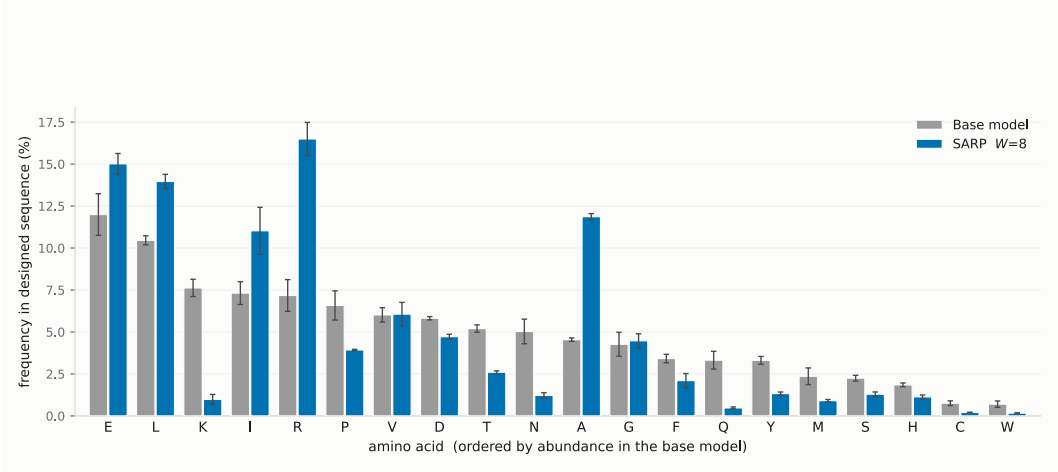

Figure 8: **Amino-acid composition before and after SARP post-training.** Mean residue frequencies are shown for the pretrained base model and the later $W = 8$ SARP checkpoint (step 177), averaged across four small-molecule binder targets. Error bars indicate variability across ligand targets.

These measurements reinforce an important distinction: gradient norms describe where optimization pressure is applied, and the resulting sequence and backbone changes reflect the coupled effect of reinforcement, endpoint supervision, and the model's joint sequence–structure parameterization.

O Ablation on Beam-Search Scale and Supervision Duration

We investigate two coupled design choices in search-augmented post-training: the strength of beam search and the duration of search supervision. We first study beam-search scale on the small-molecule binder benchmark. Across all beam-width configurations, we keep the number of search branches fixed and compare model performance. A search candidate is retained for supervision only if its score S matches or exceeds the largest S among the 32 native rollout samples in the corresponding group (`acceptance_threshold` = `max_rollout`). Figure 9 characterizes the resulting search-time targets and their acceptance rate, while Table 13 reports downstream generation performance. Increasing beam width generally improves the quality or availability of accepted search targets, but these gains do not translate monotonically to the final generator.

We next examine how long the searched endpoints should remain under direct supervision. We vary the duration of Search-SFT under weak ($W = 2$) and strong ($W = 8$) search under a maximum

Table 13: **Beam-search width ablation.** Results are evaluated on the top 30% of 200 generated samples, ranked using the combined ΔG and min-iPAE criterion. Each block corresponds to one ligand, and rows compare beam widths $W \in \{2, 4, 8\}$ with four search branches and search supervision enabled throughout training. Best and second-best results within each ligand are shown in **bold** and underlined, respectively.

Method	Configuration		Evaluation					
	RL	Search Sup.	ΔG REU (↓)	min-iPAE Å (↓)	pLDDT (↑)	scRMSD Å (↓)	ligRMSD Å (↓)	Div. meanTM (↓)
FAD								
SARP (W=2)	✓	✓	**-1.05**	3.44	78.74	**1.134**	**8.94**	**0.385**
SARP (W=4)	✓	✓	-0.81	**3.28**	**78.93**	1.191	10.02	0.428
SARP (W=8)	✓	✓	-0.24	3.38	77.99	1.206	9.92	0.445
IAI								
SARP (W=2)	✓	✓	**-25.69**	4.41	77.27	1.22	**6.53**	0.432
SARP (W=4)	✓	✓	-22.30	4.52	76.97	1.353	7.02	**0.418**
SARP (W=8)	✓	✓	-22.24	**4.07**	**77.84**	**1.192**	7.89	0.447
OQO								
SARP (W=2)	✓	✓	**-16.63**	2.77	79.92	1.31	**5.47**	**0.414**
SARP (W=4)	✓	✓	-14.31	2.98	**80.04**	1.259	6.53	0.428
SARP (W=8)	✓	✓	-12.89	**2.72**	79.82	**1.239**	7.82	0.433
SAM								
SARP (W=2)	✓	✓	**-1.20**	4.08	80.47	1.20	**7.01**	**0.407**
SARP (W=4)	✓	✓	-0.12	**2.81**	**80.64**	1.144	7.92	0.450
SARP (W=8)	✓	✓	0.20	3.62	80.04	**1.138**	7.55	0.426

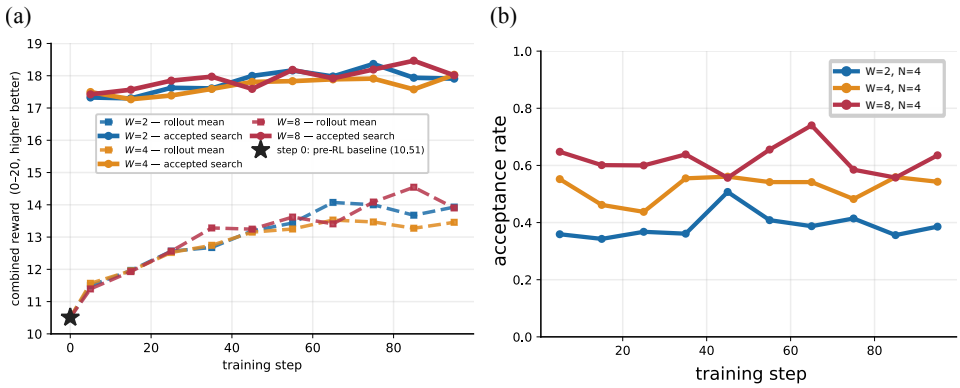

Figure 9: **Effect of beam-search scale on search-time targets in the small-molecule binder benchmark.** Search candidates are ranked using the combined score defined in Equation 44, with higher values indicating better designs. A candidate is accepted for supervision only if its score is at least as high as the best score among the corresponding 32 native rollout samples. All configurations use four search branches. **(a)** Mean score of the accepted searched candidates during post-training. Because acceptance is defined relative to the rollout maximum, these curves correspond to the selected supervision targets rather than an unconditional average over all search candidates. **(b)** Fraction of searched candidates that satisfy the acceptance criterion. Increasing search scale generally increases the frequency with which search identifies candidates exceeding the native rollout maximum.

training budget of 200 optimization steps (Table 14). The effect of supervision duration depends strongly on search scale, particularly for structural diversity. Under weak search, full supervision remains highly competitive and gives the best diversity (meanTM $= 0.410$), together with the best min-iPAE, pLDDT, and scRMSD. Turning off Search-SFT at step 90 mainly improves ΔG (-12.69 vs. -11.15 REU), while turning it off at step 135 gives the lowest ligRMSD (6.89 vs. 6.99 Å). In contrast, under stronger search, persistent supervision produces a much more concentrated generator, with meanTM increasing to 0.438. Turning off Search-SFT at step 45 reduces meanTM to 0.412

Table 14: **Interaction between beam-search scale and search-supervision duration.** Results are macro-averaged over FAD, IAI, OQO, and SAM on the top 30% of 200 generations per ligand. "Full" keeps search supervision active throughout post-training. Best results within each beam width are shown in **bold**, and second-best results are underlined.

Beam width	SFT turn-off	$\Delta G \downarrow$ REU	min-iPAE $\downarrow$ Å	pLDDT $\uparrow$	scRMSD $\downarrow$ Å	ligRMSD $\downarrow$ Å	Div. $\downarrow$ meanTM
$W = 2$	Full	-11.15	**3.67**	**79.10**	**1.22**	6.99	**0.410**
	90	**-12.69**	4.28	78.55	1.31	7.14	0.421
	135	-12.49	4.57	77.67	1.36	**6.89**	0.415
$W = 8$	Full	-8.79	**3.45**	78.92	1.20	8.30	0.438
	45	**-12.02**	3.94	**79.54**	**1.14**	7.03	**0.412**
	90	-11.34	3.73	79.33	1.25	**6.82**	0.415
	140	-10.71	4.27	78.50	1.27	7.59	0.421

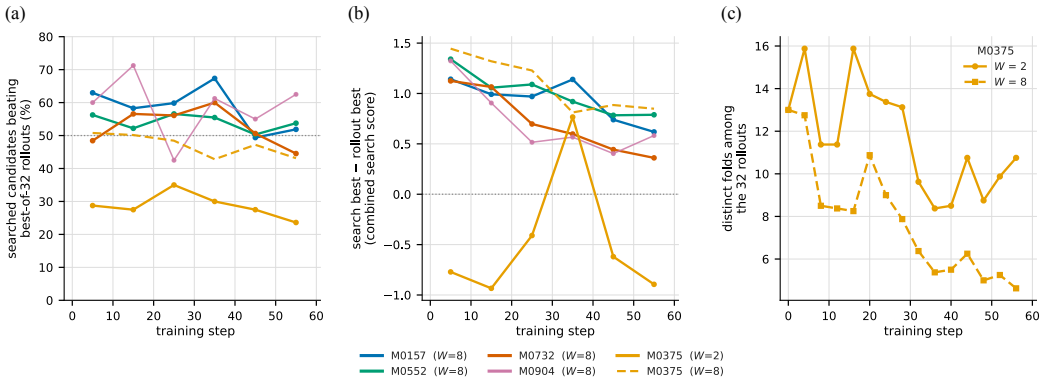

Figure 10: **Search advantage and diversity dynamics during enzyme post-training.** **(a)** Fraction of searched candidates whose reward exceeds the best of the corresponding 32 native rollout samples. Search frequently identifies candidates beyond the native rollout maximum. M0375 uses beam width $W = 2$ in the main configuration; the dashed curve shows the same target with $W = 8$. **(b)** Reward margin between the best searched endpoint and the best native rollout. The margin generally decreases over training, indicating that native rollouts progressively approach the quality of search-discovered endpoints. **(c)** Structural diversity of the 32 native rollouts during post-training on M0375, measured by the number of distinct Foldseek clusters. Wider search supervision ($W = 8$) leads to a substantially faster reduction in rollout diversity than $W = 2$.

while also improving ΔG from -8.79 to -12.02 REU, pLDDT from 78.92 to 79.54, and scRMSD from 1.20 to 1.14 Å. Full supervision retains an advantage only in min-iPAE.

To further examine the training dynamics underlying this trade-off, we perform a complementary analysis in the enzyme-design setting. Figure 10 compares searched endpoints with the native rollout distribution during post-training. Search frequently identifies candidates that outperform even the best of 32 native rollouts, while the reward margin between the best searched endpoint and the best native rollout generally decreases over training, indicating that the improving policy progressively approaches the quality of search-discovered endpoints. Moreover, on M0375, increasing the beam width from $W = 2$ to $W = 8$ causes a substantially faster reduction in the structural diversity of the native rollouts. These dynamics are consistent with the duration ablation above: search can provide a useful optimization signal, particularly earlier in training, but stronger and persistent supervision can increasingly concentrate the learned distribution.

Overall, these results show that search scale and supervision duration should not be optimized independently. Stronger search can provide more competitive supervised endpoints, but increasing search strength does not consistently improve downstream generation, and prolonged supervision under strong search can substantially reduce structural diversity. Conversely, weaker search can tolerate longer supervision without the same degree of concentration. The appropriate Search-SFT

Table 15: Endpoint-source ablation for small-molecule binder design. SARP-Rollout and SARP use full endpoint supervision under the same 200-step optimization horizon. Results are macro-averaged over FAD, IAI, OQO, and SAM on the top 30% of ranked designs. Best results are shown in bold, and second-best results are underlined.

Method	$\Delta G \downarrow$	min-iPAE $\downarrow$	scRMSD $\downarrow$	ligRMSD $\downarrow$	pLDDT $\uparrow$	Div. $\downarrow$
Base model	1.86	5.82	3.61	9.87	69.35	**0.351**
SARP-Rollout	**-11.20**	<u>3.88</u>	**1.19**	<u>7.32</u>	**79.21**	0.426
SARP	<u>-11.15</u>	**3.67**	<u>1.22</u>	**6.99**	<u>79.10</u>	<u>0.410</u>

schedule therefore depends on the strength of the search procedure and the evolving gap between searched endpoints and the current policy's own rollouts.

P Ablation on Search-Derived vs. Rollout-Derived Endpoint Supervision

To isolate the effect of the source of supervised endpoints, we compare SARP against a matched rollout-derived control, denoted SARP-Rollout. SARP uses high-reward endpoints obtained through reward-guided search and SARP-Rollout replaces these targets with high-reward endpoints selected directly from the ordinary on-policy rollout group. All other optimization components, including DGPO, reference regularization, and the endpoint-supervision objective, are unchanged.

SARP-Rollout is intended as a *source-control ablation* rather than as an independently tuned competing method. We therefore match the training horizon and supervision exposure to the corresponding SARP configuration, so that the primary difference is the source of the supervised endpoints. For the small-molecule binder experiments, both variants use full endpoint supervision under the same 200-step optimization horizon. For enzyme design, SARP-Rollout uses the same target-specific total training horizon and the same number of supervised steps as the corresponding SARP run. Because the enzyme horizons are inherited from the SARP runs rather than independently optimized for SARP-Rollout, these results should not be interpreted as the best achievable performance of an independently tuned rollout-distillation method.

Small-molecule binder design. Table 15 compares full-supervision SARP and SARP-Rollout under the same 200-step training horizon. The two endpoint sources yield very similar overall binder quality. SARP-Rollout achieves slightly better binding energy, scRMSD, and pLDDT, whereas search-derived SARP achieves better min-iPAE, ligand RMSD, and structural diversity. In particular, mean structural similarity decreases from 0.426 for SARP-Rollout to 0.410 for SARP, indicating modestly greater diversity under search-derived supervision. These results suggest that, for small-molecule binder optimization, endpoint source has relatively little effect on local quality under this training budget, while still affecting the distribution of generated structures.

Enzyme design. The distinction becomes substantially larger in the target-specific enzyme setting (Table 16). Rollout-derived supervision retains local geometric quality comparable to search-derived supervision: averaged over the five targets, SARP-Rollout achieves catalytic-site RMSD of 1.35 Å and backbone RMSD of 0.53 Å, compared with 1.38 Å and 0.59 Å for SARP. However, the diversity of successful designs differs markedly. SARP produces 42.2 unique passing Foldseek clusters per target on average, compared with 21.0 for SARP-Rollout, while also achieving a higher average pass rate (69.4% vs. 62.2%).

Together, the two settings indicate that search-derived supervision is not simply a stronger local fitting signal. Rollout-derived supervision can match or occasionally outperform search-derived supervision on individual geometric metrics, particularly in the small-molecule setting. The larger difference appears in which successful regions of structure space are reinforced: search-derived supervision preserves greater diversity, with the effect becoming especially pronounced on the harder enzyme-design tasks.

Table 16: Endpoint-source ablation for target-specific enzyme design. SARP-Rollout uses the same total training horizon and endpoint-supervision duration as the corresponding SARP run. Results use 100 standard generations without inference-time search. Train #step is reported as $a(b)$, where a is the total number of post-training steps and b is the number of steps with endpoint supervision enabled. Best results within each target are shown in bold, and second-best results are underlined.

Target	Method	Train #step	Cat. match ↑	Cat. RMSD ↓	BB RMSD ↓	Lig. clash ↓	Pass ↑	Unique pass ↑
M0732	Base model	–	0.998	2.95	1.49	0.008%	20%	17
	SARP-Rollout	60(20)	**1.000**	**1.09**	**0.52**	0.010%	80%	22
	SARP	60(20)	**1.000**	1.11	0.58	**0.006%**	**89%**	**44**
M0157	Base model	–	0.999	2.74	1.77	**0.001%**	18%	17
	SARP-Rollout	90(30)	**1.000**	1.35	**0.54**	0.007%	63%	**43**
	SARP	90(30)	**1.000**	**1.34**	0.58	0.003%	**74%**	**43**
M0552	Base model	–	0.998	3.09	1.52	0.004%	11%	7
	SARP-Rollout	90(30)	**1.000**	1.42	**0.47**	**0.000%**	58%	31
	SARP	90(30)	**1.000**	**1.36**	0.57	0.003%	**72%**	**54**
M0904	Base model	–	**1.000**	2.39	1.28	**0.000%**	21%	12
	SARP-Rollout	140(20)	0.999	**1.06**	**0.55**	0.001%	**92%**	1
	SARP	140(20)	**1.000**	1.20	0.60	**0.000%**	88%	**52**
M0375	Base model	–	0.999	3.18	1.49	0.011%	5%	3
	SARP-Rollout	80(20)	0.999	**1.85**	**0.56**	**0.003%**	18%	8
	SARP	80(20)	**1.000**	1.91	0.63	0.008%	**24%**	**18**
Average	Base model	–	0.999	2.87	1.51	0.005%	15.0%	11.2
	SARP-Rollout	–	**1.000**	**1.35**	**0.53**	**0.004%**	62.2%	21.0
	SARP	–	**1.000**	1.38	0.59	**0.004%**	**69.4%**	**42.2**

Q TRAJECTORY-BASED GRPO IMPLEMENTATION

For comparison with the trajectory-free DGPO update in Section 3.1.3, we implement a direct Flow-GRPO-style baseline (Liu et al., 2026) that treats the base model's generation process as a long-horizon policy trajectory. This baseline uses the same rollout groups and group-relative advantages $A^{(g)}$ defined in Section 3.1.2 and it differs only in how the advantages are used to update the policy. Unlike DGPO that operates on re-noised final samples, trajectory-based GRPO requires caching intermediate rollout states and computing old/new policy ratios along the denoising trajectory.

First, we summarize several adaptations to apply GRPO to the coupled, partially latent Complexa-style generator:

- **Dual-modality trajectory.** The base model generates each structure through two coupled trajectories, one for explicit C_α backbone coordinates and one for non-C_α atomic latents. We therefore compute transition probabilities and importance ratios separately for the backbone and latent modalities.
- **Per-modality clipping.** We apply PPO clipping separately to the backbone and latent ratios before summing the surrogate losses. This gives each modality its own trust region and prevents excessive drift in one modality from suppressing the learning signal in the other.
- **Shared final-structure advantage.** Like DGPO, the same group-relative advantage $A^{(g)}$ is used for both modalities because the rewards are computed on the final all-atom structure and generally depend jointly on backbone placement and non-C_α atomic detail.
- **Stochastic-step subset.** Importance ratios are computed only on selected stochastic denoising steps. Deterministic ODE tail steps in the default Proteína-Complexa sampler have zero transition noise and are therefore excluded because they do not define stochastic transition probabilities. Flow-GRPO (Liu et al., 2026) also proposes a fast variant that samples only a single SDE step along the denoising trajectory, uses ODE integration for the remaining steps, and applies the policy update only at that stochastic step. We tested this strategy, but found it less suitable for our protein generation setting: a small number of stochastic steps did not provide sufficient within-group diversity and deviated from the standard Proteína-Complexa inference schedule used for high-quality generation. We therefore compute GRPO updates on selected stochastic steps from the full Proteína-Complexa SDE rollout.

Algorithm 2 summarizes the implementation.

Algorithm 2 Trajectory-Based GRPO Training Loop

Require: Online policy θ, reference policy θ_{ref}, behavior policy θ_{old}, design dataset $\mathcal{D}$, group size G, integration steps $K = 400$, selected stochastic steps $\mathcal{S}$.

Initialize θ from the pretrained base model weights.
Set $\theta_{\mathrm{ref}} \leftarrow \theta$ and freeze θ_{ref}.
Set $\theta_{\mathrm{old}} \leftarrow \theta$.
for each optimizer step **do**
  Sample a design condition $\mathcal{C} \sim \mathcal{D}$.
  Generate G stochastic rollouts from θ_{old} under the same condition $\mathcal{C}$.
  for each rollout $g = 1, \ldots, G$ **do**
    Cache the full denoising trajectory $\{\boldsymbol{x}_{\tau_k}^{(g,m)}\}_{k=0}^{K}$ for $m \in \{\mathrm{bb}, \mathrm{lat}\}$.
    Cache behavior log-probabilities $\log \pi_{\theta_{\mathrm{old}}}^{m}$ on selected stochastic steps $k \in \mathcal{S}$.
    Decode the final generated structure $\boldsymbol{x}_1^{(g)}$.
    Compute task rewards and structural penalties on $\boldsymbol{x}_1^{(g)}$.
  end for
  Compute group-relative advantages $\{A^{(g)}\}_{g=1}^{G}$ from the rollout rewards.
  for each selected stochastic step $k \in \mathcal{S}$ **do**
    for each rollout g and modality $m \in \{\mathrm{bb}, \mathrm{lat}\}$ **do**
      Recompute current log-probability $\log \pi_\theta^m\Big(\boldsymbol{x}_{\tau_{k+1}}^{(g,m)} \mid \boldsymbol{x}_{\tau_k}^{(g)}\Big)$.
      Compute importance ratio $\rho_k^{(g,m)} = \exp\big(\log \pi_\theta^m - \log \pi_{\theta_{\mathrm{old}}}^m\big)$.
    end for
  end for
  Compute the per-modality clipped policy loss $\mathcal{L}_{\mathrm{policy}}$.
  Compute the reference regularization $\mathcal{L}_{\mathrm{KL}}$ against θ_{ref}.
  Update θ by minimizing $\mathcal{L}_{\mathrm{GRPO}} = \mathcal{L}_{\mathrm{policy}} + \beta_{\mathrm{kl}}\mathcal{L}_{\mathrm{KL}}$.
  if optimizer step is divisible by N_{old} **then**
    Re-synchronize the behavior policy: $\theta_{\mathrm{old}} \leftarrow \theta$.
  end if
end for

Denoising trajectory as a policy. For each design condition $\mathcal{C}$, we generate a group of G samples with the stochastic Proteína-Complexa sampler over the integration schedule $0 = \tau_0 < \tau_1 < \cdots < \tau_K = 1$, where $K = 400$. For each sample g and modality $m \in \{\mathrm{bb}, \mathrm{lat}\}$, we retain the trajectory $\{\boldsymbol{x}_{\tau_k}^{(g,m)}\}_{k=0}^{K}$.

Intuitively, each stochastic sampler step consists of a deterministic prediction plus injected SDE noise. The deterministic part gives the next state that the sampler would take without additional noise, analogous to an ODE update. The SDE sampler then adds Gaussian noise around this mean, making the realized next state a sample from a Gaussian transition kernel. At each stochastic step, the sampler defines a Gaussian transition whose mean is given by a score-corrected Langevin drift. Specifically, the score associated with the flow-matching velocity is

$$\boldsymbol{s}_\theta^m\Big(\boldsymbol{x}_{\tau_k}^{(g)}, \tau_k\Big) = \frac{\tau_k\, \boldsymbol{v}_\theta^m\Big(\boldsymbol{x}_{\tau_k}^{(g)}, \tau_k\Big) - \boldsymbol{x}_{\tau_k}^{(g,m)}}{\alpha_m^2(1-\tau_k)}, \tag{56}$$

where α_m is the modality-specific reference scale. The one-step transition mean is then

$$\boldsymbol{\mu}_\theta^m\Big(\boldsymbol{x}_{\tau_k}^{(g)}, \tau_k\Big) = \boldsymbol{x}_{\tau_k}^{(g,m)} + \Big[\boldsymbol{v}_\theta^m\Big(\boldsymbol{x}_{\tau_k}^{(g)}, \tau_k\Big) + g_{\tau_k}\boldsymbol{s}_\theta^m\Big(\boldsymbol{x}_{\tau_k}^{(g)}, \tau_k\Big)\Big]\,\Delta\tau_k, \tag{57}$$

with $\Delta\tau_k = \tau_{k+1} - \tau_k$ and SDE noise schedule g_{τ_k}. The per-step noise scale is

$$\sigma_{\tau_k}^m = \sqrt{2 g_{\tau_k} \lambda_{\mathrm{noise}}^m \Delta\tau_k}, \tag{58}$$

where $\lambda_{\mathrm{noise}}^m$ is the modality-specific sampler noise coefficient. The sampled next state is therefore

$$\boldsymbol{x}_{\tau_{k+1}}^{(g,m)} = \boldsymbol{\mu}_\theta^m\Big(\boldsymbol{x}_{\tau_k}^{(g)}, \tau_k\Big) + \sigma_{\tau_k}^m \boldsymbol{\xi}_k^{(g,m)}, \qquad \boldsymbol{\xi}_k^{(g,m)} \sim \mathcal{N}(0, I). \tag{59}$$

Here $\boldsymbol{x}_{\tau_k}^{(g)} = (\boldsymbol{x}_{\tau_k}^{(g,\mathrm{bb})}, \boldsymbol{x}_{\tau_k}^{(g,\mathrm{lat})})$ denotes the full joint state, while $\boldsymbol{\mu}_\theta^m$ predicts the next-state mean for modality m.

The corresponding per-step log-probability is

$$\begin{aligned}\log \pi_\theta^m\left(\boldsymbol{x}_{\tau_{k+1}}^{(g,m)} \mid \boldsymbol{x}_{\tau_k}^{(g)}\right) &= \log \mathcal{N}\left(\boldsymbol{x}_{\tau_{k+1}}^{(g,m)}; \boldsymbol{\mu}_\theta^m\left(\boldsymbol{x}_{\tau_k}^{(g)}, \tau_k\right), (\sigma_{\tau_k}^m)^2 I\right) \\ &= -\frac{d_m^{(g)}}{2}\log(2\pi(\sigma_{\tau_k}^m)^2) - \frac{\left\|\boldsymbol{x}_{\tau_{k+1}}^{(g,m)} - \boldsymbol{\mu}_\theta^m\left(\boldsymbol{x}_{\tau_k}^{(g)}, \tau_k\right)\right\|_{\text{mask}}^2}{2(\sigma_{\tau_k}^m)^2}.\end{aligned} \tag{60}$$

During rollout, these log-probabilities are cached under the behavior policy θ_{old} for selected stochastic steps $\mathcal{S} \subseteq \{0, \ldots, K-1\}$. Deterministic ODE tail steps with $\sigma_{\tau_k}^m = 0$ are excluded from $\mathcal{S}$ because they do not define stochastic transition probabilities for importance-ratio estimation.

Importance ratios and clipped objective. At each selected step $k \in \mathcal{S}$, we compute a per-modality importance ratio

$$\rho_k^{(g,m)} = \exp\left[\log \pi_\theta^m\left(\boldsymbol{x}_{\tau_{k+1}}^{(g,m)} \mid \boldsymbol{x}_{\tau_k}^{(g)}\right) - \log \pi_{\theta_{\text{old}}}^m\left(\boldsymbol{x}_{\tau_{k+1}}^{(g,m)} \mid \boldsymbol{x}_{\tau_k}^{(g)}\right)\right]. \tag{61}$$

As in DGPO, we use the same scalar group-relative advantage $A^{(g)}$ for both modalities because the rewards are computed on the final all-atom structure. The trajectory-based GRPO policy loss is the clipped PPO surrogate (Schulman et al., 2017), averaged over selected steps, samples, and modalities:

$$\mathcal{L}_{\text{policy}} = -\mathbb{E}_{k\in\mathcal{S}}\left[\frac{1}{G}\sum_{g=1}^{G}\sum_{m\in\{\text{bb,lat}\}}\min\left(\rho_k^{(g,m)}A^{(g)}, \text{clip}\left(\rho_k^{(g,m)}, 1-\epsilon, 1+\epsilon\right)A^{(g)}\right)\right], \qquad \epsilon = 0.2. \tag{62}$$

We apply clipping separately to the backbone and latent ratios before summing the surrogate losses. This gives each modality its own trust region, which is useful because the backbone and latent trajectories have different scales and optimization dynamics. If one modality drifts outside the clipping range, only that modality's update is clipped, while the other modality can still receive a learning signal. A joint-ratio alternative would multiply the backbone and latent ratios before clipping, but this can overly couple the two updates: drift in one modality can suppress the update for both, while opposite ratio changes can partially cancel. We therefore use per-modality clipping for the GRPO baseline.

This objective directly increases the probability of trajectory transitions that lead to above-average final samples and decreases the probability of transitions that lead to below-average samples.

Reference regularization. We regularize the online policy toward the reference policy using the KL divergence between the Gaussian transition kernels. For equal variances, this reduces to a scaled squared difference between transition means:

$$\mathcal{L}_{\text{KL}} = \mathbb{E}_{k\in\mathcal{S}}\left[\frac{1}{G}\sum_{g=1}^{G}\sum_{m\in\{\text{bb,lat}\}}\frac{1}{2(\sigma_{\tau_k}^m)^2}\left\|\boldsymbol{\mu}_\theta^m\left(\boldsymbol{x}_{\tau_k}^{(g)}, \tau_k\right) - \boldsymbol{\mu}_{\theta_{\text{ref}}}^m\left(\boldsymbol{x}_{\tau_k}^{(g)}, \tau_k\right)\right\|^2\right]. \tag{63}$$

The final GRPO baseline objective is

$$\mathcal{L}_{\text{GRPO}} = \mathcal{L}_{\text{policy}} + \beta_{\text{kl}}\mathcal{L}_{\text{KL}}. \tag{64}$$

Engineering details. In the real implementation, we maintain two frozen network copies in addition to the online policy θ. The *reference policy* θ_{ref} is used in the KL regularizer $\mathcal{L}_{\text{KL}}$ and is initialized from the pretrained weights, then kept fixed throughout training. The *behavior policy* θ_{old} is used to evaluate the cached rollout log-probabilities $\log \pi_{\theta_{\text{old}}}^m$ and the resulting importance ratios $\rho_k^{(g,m)}$. It is also initialized from the pretrained weights, but is re-synchronized with the online policy, $\theta_{\text{old}} \leftarrow \theta$, every $N_{\text{old}} = 5$ optimizer steps.

Keeping θ_{old} close to the online policy limits rollout staleness and helps the clipped surrogate remain a local policy update, while the fixed reference policy provides an anchor that discourages drift away from the pretrained model. In our GRPO experiments, we use $\beta_{\text{kl}} = 0.005$ and a learning rate of 3×10^{-6}. This baseline provides a direct trajectory-level group-relative RL comparison to DGPO, but requires storing the 400-step rollout trajectory and evaluating old/new transition probabilities at selected denoising steps. These memory and compute costs are precisely what the trajectory-free DGPO update avoids.