BINDGRPO: TRAINING PROTEIN BINDER GENERATORS TO BIND

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models of protein binders are typically trained by maximizing likelihood on natural complexes. Binding is considered primarily at inference: many candidates are sampled, scored by a structure predictor, and the highest-confidence ones are retained. Binding is therefore not part of the generator's training objective. We introduce BindGRPO to close this gap by post-training a target-conditioned binder generator using predicted binding as the reward. We formulate the denoising sampler as a Markov decision process and define Gaussian policy transitions between consecutive denoiser inputs with tractable likelihoods. The reward is AlphaFold3's interface confidence for the co-designed sequence. Training therefore does not require an experimentally determined target–binder complex. On the Cao benchmark, BindGRPO increases the in-silico success rate from 24.4% to 36.7%, a $1.51\times$ gain over the same generator before RL fine-tuning. The improvement persists under an independent AlphaFold2 evaluator and across most evaluation thresholds, suggesting that the gain is not specific to AlphaFold3. All code and data will be released.

1 INTRODUCTION

Proteins that bind a target protein are used as drugs and as laboratory tools. They have traditionally been discovered through immunization or large display-library screens. These methods are slow and provide limited control over the binding site (Laustsen et al., 2021). Computational design instead takes a target and a desired binding site as inputs and generates a binder for that site (Cao et al., 2022; Pacesa et al., 2025).

Generative models have made computational binder design practical. Two-stage methods generate target-conditioned backbone structures and then assign sequences with an inverse-folding model (Watson et al., 2023; Dauparas et al., 2022). More recent co-design models generate sequence and structure jointly (Stark et al., 2025; Didi et al., 2026). Several such approaches have produced experimentally validated binders (Watson et al., 2023; Zambaldi et al., 2024; Stark et al., 2025).

Most of these generators do not directly optimize binding during training. Instead, they are typically trained by maximum likelihood on natural protein structures and therefore learn the distribution represented in the training data rather than a target-specific binding objective. Current pipelines address this mismatch through inference-time filtering: they sample many candidates, score them with a structure predictor, and retain those above a confidence threshold (Bennett et al., 2023; Pacesa et al., 2025; Zambaldi et al., 2024). Such filtering substantially enriches for experimentally binding designs, as shown by Zambaldi et al. (2024) in an analysis of the 6.4×10^5 designs from Cao et al. (2022). However, filtering leaves the generator itself unchanged. Binding therefore remains a selection criterion rather than a property optimized during generation.

Post-training uses the filtering score as a reward to update the generator. In language models, reinforcement learning supplements next-token prediction with rewards based on human preferences (Christiano et al., 2017; Ouyang et al., 2022) or verified answers (Shao et al., 2024). Protein reward optimization has so far focused primarily on sequences, restricted structural regions, or fixed backbones (Section 2).

We introduce BindGRPO, a reinforcement-learning method for post-training target-conditioned generators of complete binder sequences and structures. BindGRPO fine-tunes the BoltzGen denoiser

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.

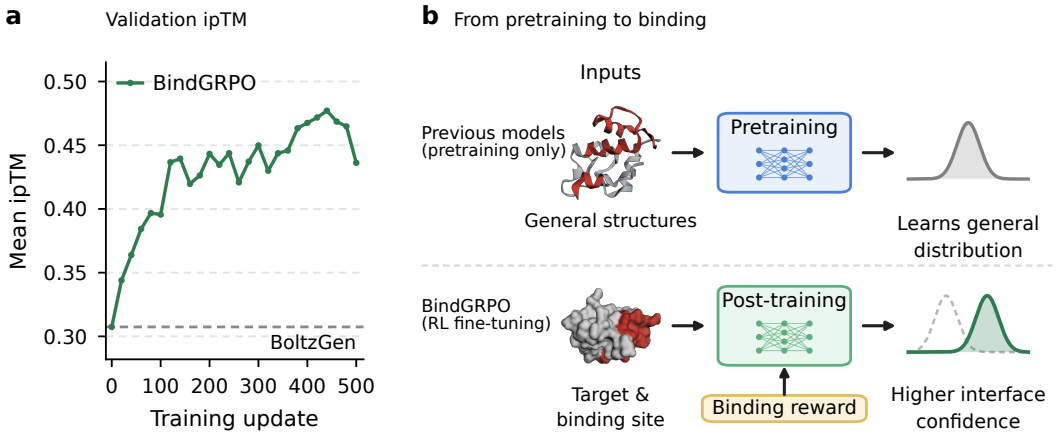


Figure 1: **a**, Mean AlphaFold3 ipTM on the validation set during GRPO training. The dashed line is pretrained BoltzGen. **b**, During pretraining, the model learns from natural protein structures. BindGRPO post-trains the same generator using a predicted-binding reward for a specified target and binding site. Red marks the masked regions or specified binding site.

using AlphaFold3's interface confidence as a proxy for binding. The reward is computed from the generator's co-designed sequence together with the target, so training does not require an experimentally determined target–binder complex (Figure 1).

Applying GRPO requires a policy with a tractable probability density, but BoltzGen's EDM sampler does not define one directly. Each sampling iteration separates a deterministic denoising update from parameter-independent stochastic augmentation and noise injection. We compose each denoising update with the subsequent noise injection to define a policy transition between consecutive denoiser inputs. Conditional on the sampled rigid augmentation, this transition has an exact Gaussian density. During training, we score the generator's co-designed sequences directly.

On the Cao et al. (2022) benchmark, BindGRPO increases the in-silico design success rate from 24.4% to 36.7% relative to BoltzGen, a $1.51\times$ gain. Benchmark sequences are reassigned with ProteinMPNN (Dauparas et al., 2022), so this primary result reflects improved generated backbones rather than only improved decoded sequences. The co-designed sequences show the same direction of improvement, from 5.0% to 10.3%, although at a lower absolute success rate. The gain is reproduced with an independent AlphaFold2 evaluator, from 28.5% to 42.0%, and persists across most evaluation thresholds (Section 6.3). Decomposing the success criterion, all three component conditions improve, with the largest gain in interface PAE, the condition most closely related to the training reward (Section 6.4).

Our contributions are:

- **Post-training for binder generators.** We fine-tune a target-conditioned sequence-and-structure generator with reinforcement learning using a predicted-binding reward. The resulting generator can be reused across targets, with filtering still applied at inference (Section 4).
- **The EDM sampler as a policy.** We define Gaussian policy transitions between consecutive denoiser inputs and recompute their densities during training. Following Liu et al. (2025), we normalize log-densities per coordinate over binder atoms (Section 4.1).
- **Training target selection.** We characterize both reward level and within-group reward variation and use them to construct the training set. Because GRPO normalizes rewards within each target group, targets whose designs receive nearly identical rewards provide little policy-gradient signal even when their mean reward is suitable (Section 5.1, Appendix D).

2 Related Work

Approaches to protein binder design. Generative binder design methods learn a parameterized model that can sample designs for new targets. Two-stage methods first generate a backbone (Watson et al., 2023; Zhang et al., 2025) and then assign sequences with a separate inverse-folding model (Dauparas et al., 2022), whereas co-design methods generate structure and sequence jointly (Qu et al., 2025; Stark et al., 2025; Didi et al., 2026). Hallucination methods train no generator; instead, they directly optimize the designed sequence against the confidence of a frozen structure predictor (Pacesa et al., 2025; Cho et al., 2025). RL fine-tuning, as in this work, instead updates the generator's parameters.

RL fine-tuning on diffusion and flow models. Treating the denoising process as an MDP (Black et al., 2024) gives three families of online methods, which differ in how the reward reaches the parameters. *Policy-gradient* methods apply REINFORCE or a PPO-style ratio to per-step log-probabilities and handle non-differentiable rewards (Black et al., 2024; Fan et al., 2023; Liu et al., 2025; Xue et al., 2025). BindGRPO belongs here. *Differentiable-reward* methods reduce variance by differentiating the reward. They either backpropagate through a truncated sampling chain (Clark et al., 2024) or replace that backpropagation with an adjoint equation (Domingo-Enrich et al., 2025). *Reward-weighted regression* keeps the update in pretraining's form by weighting the score- or flow-matching (Lipman et al., 2023) loss with an advantage (Xue et al., 2026).

Offline preference optimization. DPO-style alignment (Rafailov et al., 2023) is often grouped with RL, but it optimizes a closed-form objective over preference pairs rather than expected reward. Wallace et al. (2024) and Yang et al. (2024) extended it to diffusion models. ProteinDPO (Widatalla et al., 2026) aligns ESM-IF1, a structure-conditioned *sequence* model, to experimental $\Delta\Delta G$. AbDPO (Zhou et al., 2024) applies DPO to a diffusion model over antibody CDRs using Rosetta interface energy (Alford et al., 2017). Both fix the structure or restrict the design space to a loop, and both optimize stability or interface energy rather than target-conditioned binding.

Reward optimization applied to protein generators. Offline reward-weighted regression aligns an SE(3) flow generator (Huguet et al., 2024), and antibody-CDR methods optimize Rosetta interface energy by preference tuning (Wen et al., 2025) or trajectory balance (Abir et al., 2025). Online RL has been applied to protein language models (Stocco et al., 2024), inverse folding (Wang et al., 2026) and sequence diffusion (Sheng et al., 2026; Ye et al., 2026). The most closely related algorithm applies GRPO to an AlphaFold3-style score network for pocket inpainting on a fixed scaffold (Fan et al., 2025). Others use the same reward signal only at inference, by descending on sequence logits per design (Mille-Fragoso et al., 2025), ranking with a learned co-folding scorer (Levine-Gottreich et al., 2026), or filtering after generation (Bennett et al., 2023). In all of these methods, the generated object is a sequence, a restricted structural region or a fixed scaffold. BindGRPO instead trains a target-conditioned model to jointly generate the complete binder sequence and structure using a predicted-binding reward.

3 Preliminaries

Denoising as a Markov decision process. Given context c, a conditional diffusion model generates a clean sample x_0 from initial noise $x_T \sim \mathcal{N}(0, \sigma_T^2 I)$ through T denoising steps (Ho et al., 2020; Song et al., 2021). We index these steps by $t = T, \ldots, 1$. For samplers with Gaussian transitions, Black et al. (2024) formulate denoising as a Markov decision process (MDP). The state is $s_t = (c, t, x_t)$, the action is $a_t = x_{t-1}$, and the policy is:

$$\pi_\theta(x_{t-1} \mid x_t, c) = \mathcal{N}\big(x_{t-1}; \mu_\theta(x_t, t, c), b_t^2 I\big)\,, \tag{1}$$

where the model determines the mean and the schedule fixes the transition standard deviation $b_t > 0$.

The environment sets $s_{t-1} = (c, t-1, a_t)$. The trajectory's log-density is the sum of the transition log-densities and the parameter-independent initial log-density. This allows policy-gradient optimization of a terminal reward $R(x_0, c)$ without evaluating the marginal $p_\theta(x_0 \mid c)$ or differentiating the reward. All intermediate rewards are zero. Section 4.1 derives a policy of the form in Equation 1 for the EDM sampler.

Group-relative policy optimization. GRPO estimates its baseline from a group of samples without fitting a value function (Shao et al., 2024). For each conditioning input c, the behavior policy $\pi_{\theta_{\text{old}}}$ generates K trajectories with rewards $R_k = R(x_0^{(k)}, c)$. Each trajectory receives the advantage:

$$\hat{A}_k = \frac{R_k - \text{mean}(R_1, \ldots, R_K)}{\text{std}(R_1, \ldots, R_K) + \delta}, \tag{2}$$

where the mean and the population standard deviation are computed within the current group, and $\delta > 0$ stabilises the denominator. The same advantage is used at every step of a trajectory.

GRPO maximizes the PPO-style objective (Schulman et al., 2017):

$$\begin{aligned} \mathcal{J}(\theta) = \mathbb{E}\Bigg[\frac{1}{KT} \sum_{k=1}^{K} \sum_{t=1}^{T} \Big\{ \min\Big(& \rho_{k,t}(\theta)\hat{A}_k, \\ & \text{clip}\big(\rho_{k,t}(\theta), 1-\varepsilon, 1+\varepsilon\big)\hat{A}_k \Big) \\ & - \lambda_{KL}\, \mathbb{D}_{\text{KL}}\Big[\pi_\theta(\cdot \mid x_t^{(k)}, c) \,\Big\|\, \pi_{\text{ref}}(\cdot \mid x_t^{(k)}, c) \Big] \Big\} \Bigg], \end{aligned} \tag{3}$$

where $\rho_{k,t}(\theta) = \pi_\theta(x_{t-1}^{(k)} \mid x_t^{(k)}, c) / \pi_{\theta_{\text{old}}}(x_{t-1}^{(k)} \mid x_t^{(k)}, c)$ is the whole-action likelihood ratio. The expectation is over trajectories sampled from the behavior policy, and the KL penalty regularizes the current policy towards the frozen pretrained reference π_{ref} at each visited state (Fan et al., 2023). Its coefficient is given in Appendix A.2. Section 4.1 uses a coordinate-averaged ratio instead.

Protein binder generators. A binder generator takes a target sequence and structure $(s^{\text{target}}, x^{\text{target}})$ and a binder length L. An encoder maps these inputs to conditioning $c = \text{Enc}(s^{\text{target}}, x^{\text{target}}, L)$, which guides coordinate generation. Two-stage methods generate a backbone and then assign its sequence, while co-design methods generate both jointly:

$$\text{two-stage:} \qquad x^{\text{binder}} \sim p_\theta(\cdot \mid c), \qquad s^{\text{binder}} \sim q_\phi(\cdot \mid x^{\text{binder}}), \tag{4}$$

$$\text{co-design:} \qquad (x^{\text{binder}}, s^{\text{binder}}) \sim p_\theta(\cdot \mid c). \tag{5}$$

Here q_ϕ is a separately trained inverse-folding model (Dauparas et al., 2022). Candidate binder sequences are typically folded together with the target using a structure predictor (Evans et al., 2022; Abramson et al., 2024) and filtered by confidence thresholds (Cao et al., 2022; Pacesa et al., 2025; Zambaldi et al., 2024).

We apply RL fine-tuning to BoltzGen (Stark et al., 2025), which co-designs the sequence and structure. BoltzGen represents each designed residue with fourteen atom slots. It recovers residue identity from the generated coordinates by counting the atoms superposed on each backbone atom. It uses a modified EDM sampler (Karras et al., 2022). Each iteration applies a rigid augmentation and a churn step that adds noise, followed by a deterministic denoising update. Section 4.1 defines the policy for this sampler.

4 BINDGRPO

BindGRPO fine-tunes the diffusion module of BoltzGen (Stark et al., 2025) with a binder-coordinate variant of the group-relative objective in Equation 3. The model generates binders conditioned on a target, and AlphaFold3's interface confidence provides the terminal reward.

4.1 EDM SAMPLER AS A MARKOV DECISION PROCESS

Each iteration first centers and rigidly transforms the coordinates, then adds Gaussian noise, denoises, aligns, and takes an Euler step (Algorithm 1). We define a policy transition between consecutive noisy denoiser inputs. It combines the denoising Euler map $E_{\theta,t}$ with the next iteration's augmentation and noise injection. For $t \in \{2, \ldots, T\}$ with $b_{t-1} > 0$, this gives the Gaussian policy of Equation 1:

$$\begin{aligned} \pi_\theta(\hat{x}_{t-1} \mid \hat{x}_t, c, g_{t-1}) &= \mathcal{N}\big(\hat{x}_{t-1}; \mu_{\theta,t-1}, b_{t-1}^2 \mathbf{I}\big)\,, \\ \mu_{\theta,t-1} &= \text{ctr}(E_{\theta,t}(\hat{x}_t, c))\, \mathbf{R}_{t-1} + v_{t-1}. \end{aligned} \tag{6}$$

Algorithm 1: EDM sampler and policy transitions. Green marks the quantities recorded for RL. Each transition combines **B** of one iteration with **A** of the next; sampling is unchanged.

Input: noise levels $\sigma_T > \cdots > \sigma_0 = 0$, churn settings $(\gamma_{\text{churn}}, \sigma_{\text{churn}})$, step scale α, noise scale η, denoiser D_θ, fixed conditioning c
Output: design x_0, recorded stochastic transitions
Sample $x_T \sim \mathcal{N}(0, \sigma_T^2 \mathbf{I})$;
foreach $t \in \{T, \ldots, 1\}$ **do**
  /* **A. Augmentation and noise injection** */
  Sample $\epsilon_t \sim \mathcal{N}(0, \mathbf{I})$ and rigid motion $g_t = (\mathbf{R}_t, v_t)$;
  $\gamma_t \leftarrow \gamma_{\text{churn}} \mathbf{1}\{\sigma_{t-1} > \sigma_{\text{churn}}\}$;
  $\hat{\sigma}_t \leftarrow \sigma_t(1 + \gamma_t)$;
  $\mu_t \leftarrow \text{ctr}(x_t)\mathbf{R}_t + v_t$;
  $b_t \leftarrow \eta^2 \sqrt{\hat{\sigma}_t^2 - \sigma_t^2}$;
  $\hat{x}_t \leftarrow \mu_t + b_t \epsilon_t$;
  if $t < T$ *and* $b_t > 0$ **then** record $(\hat{x}_{t+1}, \hat{x}_t)$, g_t, and $\bar{\ell}_{\theta,t+1}$ (Equation 7);
  /* **B. Denoising, alignment, and Euler update** */
  $D_t \leftarrow D_\theta(\hat{x}_t, \hat{\sigma}_t; c)$;
  $\tilde{x}_t \leftarrow \text{align}(\hat{x}_t, D_t)$; `// Kabsch; output held fixed for gradients`
  $d_t \leftarrow (\tilde{x}_t - D_t)/\hat{\sigma}_t$;
  $x_{t-1} \leftarrow \tilde{x}_t + \alpha(\sigma_{t-1} - \hat{\sigma}_t) d_t$;

Here, ctr centers valid coordinates. The next iteration, indexed by $t-1$, applies the parameter-independent rigid transform $g_{t-1} = (\mathbf{R}_{t-1}, v_{t-1})$. We hold this transform fixed when comparing policies. Both target and binder coordinates evolve during sampling. Initialization, zero-noise transitions, and the final Euler output contribute no direct log-density terms.

The sampler adds independent Gaussian noise to each coordinate, so the transition log-density is the sum of coordinate-wise log-densities. For training, we average these terms over valid binder coordinates, denoted by $\mathcal{B}$, including auxiliary atom slots, and omit the target-coordinate terms:

$$\bar{\ell}_{\theta,t} = \frac{1}{3|\mathcal{B}|} \sum_{a \in \mathcal{B}} \sum_{d=1}^{3} \log \mathcal{N}\left(\hat{x}_{t-1,a,d}; \mu_{\theta,t-1,a,d}, b_{t-1}^2\right), \qquad \bar{\rho}_t = \exp\left(\bar{\ell}_{\theta,t} - \bar{\ell}_{\theta_{\text{old}},t}\right). \tag{7}$$

We substitute $\bar{\rho}_t$ for the whole-action ratio in Equation 3 and likewise average the reference KL over binder coordinates $\mathcal{B}$. Following Liu et al. (2025), coordinate averaging removes the explicit scaling with binder size. This ratio is a training surrogate rather than a full-complex importance ratio. Appendix A.1 gives the derivation.

4.2 Reward

The reward is AlphaFold3's (Abramson et al., 2024) interface predicted TM-score (ipTM) (Evans et al., 2022). AlphaFold3 folds the target with the binder sequence decoded from BoltzGen's terminal coordinates. The target uses MSAs, and the binder uses neither searched MSAs nor templates. Generated coordinates are not supplied to AlphaFold3. The binder sequence is decoded from the generated coordinates (Section 3), so the reward also depends on those coordinates. The reference KL penalizes changes to the generator's transition means. To keep rollouts efficient, we compute the reward using 2 diffusion steps at step scale 1.0, rather than AlphaFold3's default 200 steps and step scale of 1.5. On ProteinMPNN sequences assigned to pretrained BoltzGen backbones, this approximation largely preserves within-target ipTM rankings (median Spearman $\rho = 0.967$) and predicted binder structures (median Cα RMSD 0.50 Å; Figure 4, Appendices A.2 and B.2).

We use this reward since Zambaldi et al. (2024) evaluated confidence metrics against experimental binding measurements for 6.4×10^5 designs from Cao et al. (2022), without prior confidence filtering. Filtering by AlphaFold3 confidence yielded a higher fraction of experimental binders than the AlphaFold2 filters used in that comparison. ipTM was among the best individual predictors. On the same design set, Bennett et al. (2023) show that the confidence of a structure prediction model

separates binders from non-binders better than the physics-based scores, including Rosetta interface energy (Alford et al., 2017). Confidence of this kind predicts whether a design binds rather than how tightly (Pacesa et al., 2025), supporting its use as a reward for binder design. Nevertheless, ipTM remains an *in silico* proxy, not a direct measurement of binding (Section 7).

4.3 Training

Only the diffusion module is trained. The conditioning network and all other modules remain frozen. The reference policy π_{ref} is fixed at the initial diffusion weights. We score co-designed sequences directly.

For each target, the current policy generates $K = 32$ designs, scored by AlphaFold3 as in Section 4.2. We compute advantages using the rewards within this group (Equation 2) and clip them to limit the influence of outliers. We then reevaluate the recorded transitions and optimize the objective in Equation 3 using Equation 7. During optimization, the recorded states and actions remain fixed while only their likelihoods under the updated policy are recomputed.

Each optimizer step combines $M = 8$ target groups, giving 256 designs per update. We train the model for 500 updates, using 40 denoising steps per trajectory. Appendix A.1 and A.2 give the complete settings.

5 Experimental Setup

5.1 Training targets

A target is the structure and sequence of a protein together with the surface patch a binder is designed against (the hotspot). We train on a subset of the Human Bindome (Wenckstern et al., 2026), whose hotspot is the consensus contact set of designed binders against that protein. We use the PDB (Berman et al., 2000) only for validation, where the hotspot is the interface of a natural partner chain. Appendix B.1 gives the construction in full.

Training target selection. For each candidate target, we generate 40 BoltzGen designs and score them with AlphaFold3 (Appendix B.1). We retain targets with mean ipTM in $[0.2, 0.8)$ and an average within-group standard deviation of at least 0.0784 (Figure 5a). Under the within-group normalization in Equation 2, identical rewards yield zero advantages, regardless of the mean reward.

Leakage control. We remove any training target that is the same protein as a target in either evaluation panel, or homologous to one. The screen is based on *sequence* identity, where each test sequence is searched against the training set with MMseqs2 (Steinegger & Söding, 2017) and flagged at $\geq 30\%$ identity over $\geq 50\%$ coverage of either sequence. We remove the entire sequence cluster because removing only the flagged entry could leave homologous sequences in the pool.

5.2 Evaluation

We report results on the target panel of Cao et al. (2022). Each generator receives the same target chain, hotspot residues and binder length defined in Cao et al. (2022), and samples 100 designs. ProteinMPNN (Dauparas et al., 2022) then assigns eight sequences to every designed backbone. Each sequence is evaluated with AlphaFold3 (Abramson et al., 2024), and with the AlphaFold2 initial-guess protocol of Bennett et al. (2023), using the configuration of Zambaldi et al. (2024) (Section 6.3). We also evaluate seven additional targets from Liu et al. (2026) that do not overlap with those in Cao et al. (2022). We use the binder lengths and hotspots specified by Liu et al. (2026). Appendix B.2 gives every setting.

A sequence is counted as a success using the criteria of Zambaldi et al. (2024). Under AlphaFold3, success requires a minimum interchain PAE below 1.5, binder pTM above 0.8, and complex RMSD below $2.5\,\text{Å}$ against the generated backbone. Under the AlphaFold2 initial guess protocol, success requires an interface PAE below 7.0, binder RMSD below $1.5\,\text{Å}$, and pLDDT above 90. We report the rate per *design*: a design succeeds if any of its eight sequences does. Rates are pooled over

all eleven targets, and arms are compared by a Wilcoxon signed-rank test over the paired per-target rates. Section 6.4 instead evaluates each sequence separately.

We benchmark BindGRPO against four open-source generators we run ourselves: PXDesign (Team et al., 2025), RFdiffusion3 (Butcher et al., 2025), Proteina-Complexa (Didi et al., 2026) and ODesign (Zhang et al., 2025), each under the protocol above. Binder length is fixed to the Cao et al. (2022) specification rather than each method's preferred range, and each generator's own selection stage is disabled (Appendix B.2).

6 Results

6.1 Success rate on the Cao benchmark

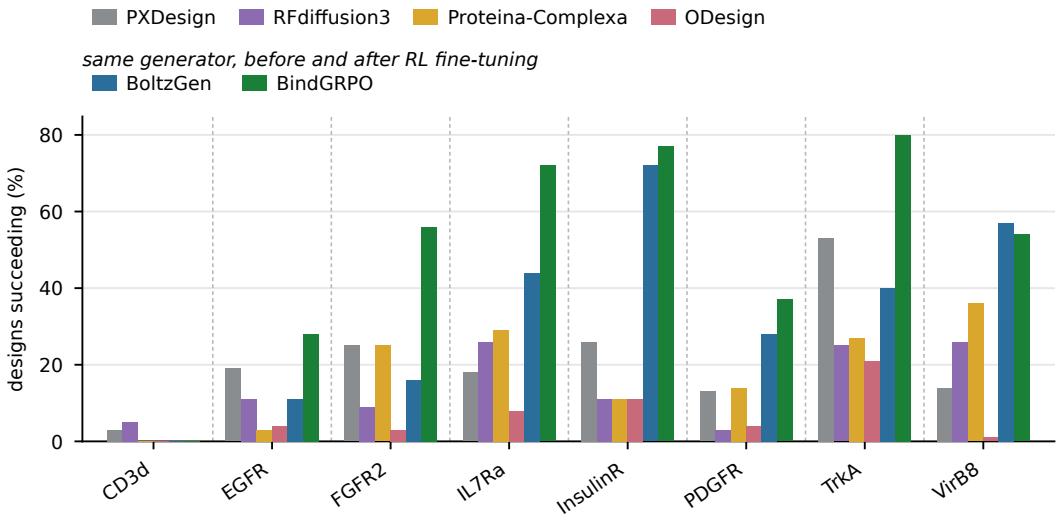


Figure 2: Per-target success rates on the Cao et al. (2022) benchmark. All methods were evaluated using the protocol in Section 5.2 ($n = 100$ designs per target). Each generator's own filtering or search stage is disabled. The two rightmost methods differ only in their denoiser weights. Three further targets, H3, TGF-β and Tie2, are omitted: no method produced a success on any of them. Rates reported in the text and tables are pooled over all eleven.

Figure 2 gives the success rate on each target, for every method under the same protocol of Section 5.2. Pooled over all eleven targets, BindGRPO reaches 36.7% (Wilson interval $[33.9, 39.6]$) against 24.4% ($[21.9, 27.0]$) for BoltzGen, a relative gain of $1.51\times$. BindGRPO improves on six of the eight targets for which any method produces a successful design, ties on CD3d, and performs worse on VirB8 (Wilcoxon signed-rank over per-target rates, $p = 0.031$). No method produces a successful design for the remaining three targets. BindGRPO reaches the highest success rate on this panel, $2.36\times$ that of PXDesign.

The sequences used for scoring were not produced directly by the generator. ProteinMPNN redesigns sequences from the backbone structures, so the improvement comes from the structural designs. Evaluating the co-designed sequences gives the same trend, but at a lower absolute success rate ($5.0\% \rightarrow 10.3\%$ per design, Appendix C.2).

6.2 Success rate on additional targets

We repeat the comparison of Section 6.1 on the seven additional targets in Liu et al. (2026), as shown in Table 1. BindGRPO reaches 23.9% ($[20.8, 27.2]$) against 21.1% ($[18.3, 24.3]$) for BoltzGen, a gain of $1.13\times$ (Wilcoxon $p = 0.313$). Restricted to the three single-chain targets, the gain is $1.14\times$, smaller than the $1.51\times$ improvement on Cao et al. (2022). Most of the gain comes from PD-L1 (46% to 62%): rates increase on four targets, remain unchanged on two, and decrease on TNF-α

Table 1: Success rate on the seven additional targets, per design ($n = 100$ per target). The last two columns average over all seven and over the three single-chain targets. Proteina-Complexa takes a single target chain and was run on those three only. Four of the seven are multi-chain assemblies, which the Cao et al. (2022) panel does not contain (Appendix B.2.1). Dashes denote unavailable results.

	BHRF1	SC2RBD	PD-L1	H1	VEGF-A	IL-17A	TNF-α	seven	three
BindGRPO	**100**	4	62	0	1	0	0	23.9	55.3
BoltzGen	99	1	46	0	0	0	2	21.1	48.7
PXDesign	84	**42**	**76**	**1**	**19**	0	**7**	**32.7**	**67.3**
Proteina-Complexa	95	1	43	–	–	–	–	–	46.3
RFdiffusion3	53	18	37	0	3	0	0	15.9	36.0
ODesign	57	8	1	0	2	0	0	9.7	22.0

Table 2: The criterion of Zambaldi et al. (2024) under two independent structure predictors, per design over all eleven targets, with 95% Wilson intervals as subscripts. Methods are compared by Wilcoxon signed-rank test over paired per-target rates. The thresholds differ between predictors (Section 5.2).

	AlphaFold3 (%)	AlphaFold2 initial guess (%)
BindGRPO	**36.7** $_{[33.9,\,39.6]}$	**42.0** $_{[39.1,\,44.9]}$
BoltzGen	24.4 $_{[21.9,\,27.0]}$	28.5 $_{[25.9,\,31.2]}$
PXDesign	15.5 $_{[13.5,\,17.8]}$	36.9 $_{[34.1,\,39.8]}$
Proteina-Complexa	13.2 $_{[11.3,\,15.3]}$	28.6 $_{[26.0,\,31.4]}$
RFdiffusion3	10.5 $_{[8.9,\,12.5]}$	26.6 $_{[24.1,\,29.3]}$
ODesign	4.7 $_{[3.6,\,6.1]}$	17.0 $_{[14.9,\,19.3]}$
BindGRPO / BoltzGen	$1.51\times$ ($p = 0.031$)	$1.48\times$ ($p = 0.016$)

(2% to 0%). PXDesign performs best on this panel, whereas BindGRPO performs best on the Cao et al. (2022).

6.3 Robustness to the evaluator and the threshold

We repeated evaluation of BoltzGen and BindGRPO designs with AlphaFold2 initial guess (Bennett et al., 2023), using the thresholds of Zambaldi et al. (2024). AlphaFold2 was not used during training. The initial guess starts AlphaFold2 from the design. Thus, this test measures whether AlphaFold2 preserves the proposed complex rather than whether it can reconstruct it from sequence alone.

BindGRPO outperforms BoltzGen under both predictors and has the highest AlphaFold3 success rate on this panel (Table 2). We re-ran the grids that Zambaldi et al. (2024) swept, scoring every combination of the three conditions per design over all eleven targets (Appendix B.2). BindGRPO scores above BoltzGen at 1,145 of 1,170 retained AlphaFold3 settings and 672 of 720 retained AlphaFold2 settings, significant at 27.3% of the AlphaFold3 and 70.0% of the AlphaFold2 settings (Figure 6, Appendix B.2). Statistical significance is not uniform across the grids, whose cutoffs vary in selectivity; it holds at the AlphaProteo-selected criteria for both predictors and at some adjacent settings. Thus, the improvement persists across predictors and most crucial threshold settings.

6.4 Decomposing the success criterion

Optimizing against a learned scorer can exploit its systematic errors, so an increase in ipTM alone does not show that the designs improved. The success criterion of Section 5.2 is a conjunction of three conditions, and interface PAE is the one most closely related to ipTM. We therefore evaluate the three conditions separately (Table 3). Here we count each sequence separately rather than per design. The three conditions are evaluated on a single folded complex, and a design's eight sequences can fail on different ones, so a per-design rate would not say which condition was responsible.

Table 3: The success criterion, decomposed. Each column is one condition scored on its own, per sequence over the 8,800 sequences per arm, with the median of the quantity it thresholds. Interface PAE is the condition the reward is closest to, and binder pTM the one it never sees.

	min interchain PAE < 1.5	binder pTM > 0.8	complex RMSD < 2.5 Å
Pass rate			
BoltzGen	18.8%	28.3%	40.1%
BindGRPO	30.5%	38.3%	49.3%
Ratio	1.63×	1.35×	1.23×
Median value			
BoltzGen	5.07	0.78	5.20 Å
BindGRPO	2.88	0.79	2.57 Å

All three conditions improve. The improvement is largest for interface PAE, the condition most closely related to the reward. The binder pTM is the sole failing condition for 11.4% of BindGRPO's failures compared with 5.5% of BoltzGen's. Most failures violate more than one condition (76.5% for BindGRPO and 84.6% for BoltzGen). Two of the four re-run baselines (Appendix C.1) show the binder pTM as their lowest-passing condition. For both BoltzGen and BindGRPO, interface PAE is the lowest-passing condition. This limitation is therefore not introduced by RL fine-tuning.

7 Limitations

The reward as a proxy for binding. We optimize against AlphaFold3's interface confidence, which predicts measured binding imperfectly. All binding results in this paper are *in silico*. We have not shown that the designs bind experimentally. ipTM measures predicted interface confidence rather than binding affinity. The reward is never differentiated through, so the same framework can use any scalar score computed from a final design. It could instead optimize specificity, expression, stability, affinity, or combinations of these properties.

Scope. The binders we generate and evaluate are miniproteins. Antibodies and cyclic peptides are more useful against many targets, and we have tested neither. The objective does not require a particular binder type. However, our experiments are limited to miniproteins because they are the only binder class with an established *in silico* benchmark. Both models show near-zero success on multimeric targets. RL training used single-chain targets, and these experiments show little improvement on multimers.

No wet-lab validation. We do not synthesize or assay any of the generated designs, so nothing reported here is evidence that they bind in the laboratory. We use the criterion calibrated by Zambaldi et al. (2024) on the 6.4×10^5 experimentally characterized designs of Cao et al. (2022), where it enriches for measured binding. However, enrichment on their dataset does not directly measure binding in our study.

8 Conclusion

We introduce BindGRPO, which post-trains a target-conditioned binder generator via RL fine-tuning using a binding reward before candidate filtering. We formulate the sampler as a Markov decision process and train its denoiser with a policy-gradient objective based on structure-predicted interface confidence. On the Cao et al. (2022) benchmark, BindGRPO increases the success rate from 24.4% to 36.7% relative to BoltzGen. The gain reproduces under a second, independent structure predictor. More broadly, reinforcement learning moves target properties from post-hoc selection into the generator's training objective.

AI use statement

We used generative AI tools to formulate the mathematical claims of Section 4.1 and write their derivation, to give feedback on the experimental design, to implement parts of the method and the

analysis code, to clean and reformat the datasets, to support and interpret the analysis of results, to create and modify the figures, to draft and edit parts of this paper, and to summarize and identify related literature. We did not use them to generate synthetic datasets, to propose or refine the research hypothesis, or for translation. The remaining tasks carrying a required disclosure do not apply here.

We have reviewed all AI-assisted work and take responsibility for it. Every figure is generated from the underlying data by a released script, and AI-assisted code was reviewed and tested by the authors.

Ethics statement

Designing proteins that bind a chosen target is dual-use. Two things limit what this paper adds to that risk. BoltzGen and the structure predictor we use as a reward are both already public, so our contribution is a higher rate of designs passing an existing filter rather than a capability that did not exist before. And every target here is a benchmark protein from published de novo design panels. Synthesis of any designed sequence remains subject to the same provider screening as any other order. The work involves no human subjects and no personal data. Every data source is public and openly licensed: the PDB under CC0 1.0, and the Human Bindome under CC BY 4.0 (Berman et al., 2000; Wenckstern et al., 2026).

Reproducibility statement

All code and data will be released. Every component we build on is open: BoltzGen (Stark et al., 2025), the PDB (Berman et al., 2000), and the Human Bindome (Wenckstern et al., 2026). The main text gives the target pool and the leakage screen (Section 5.1), the evaluation protocol (Section 5.2) and the optimization settings (Section 4.3), and the appendix gives the full hyperparameters, dataset construction and per-target results. One dependency is not ours to release: the reward is computed with the official AlphaFold3 implementation (Abramson et al., 2024) and parameters obtained from Google DeepMind, which are granted case by case and may not be redistributed, so reproducing training requires that access to be requested separately.

References

Abrar Rahman Abir, Haz Sameen Shahgir, Md Rownok Zahan Ratul, Md Toki Tahmid, Greg Ver Steeg, and Yue Dong. Abflownet: Optimizing antibody-antigen binding energy via diffusion-gflownet fusion. *arXiv*, 2025. URL https://arxiv.org/abs/2505.12358.

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016): 493–500, 2024. doi: 10.1038/s41586-024-07487-w. URL https://doi.org/10.1038/s41586-024-07487-w.

Rebecca F. Alford, Andrew Leaver-Fay, Jeliazko R. Jeliazkov, Matthew J. O'Meara, Frank P. DiMaio, Hahnbeom Park, Maxim V. Shapovalov, P. Douglas Renfrew, Vikram K. Mulligan, Kalli Kappel, Jason W. Labonte, Michael S. Pacella, Richard Bonneau, Philip Bradley, Jr. Dunbrack, Roland L., Rhiju Das, David Baker, Brian Kuhlman, Tanja Kortemme, and Jeffrey J. Gray. The rosetta all-atom energy function for macromolecular modeling and design. *Journal of Chemical Theory and Computation*, 13(6):3031–3048, 06 2017. ISSN 1549-9618. doi: 10.1021/acs.jctc.7b00125. URL https://doi.org/10.1021/acs.jctc.7b00125.

Nathaniel R. Bennett, Brian Coventry, Inna Goreshnik, Buwei Huang, Aza Allen, Dionne Vafeados, Ying Po Peng, Justas Dauparas, Minkyung Baek, Lance Stewart, Frank DiMaio, Steven

De Munck, Savvas N. Savvides, and David Baker. Improving de novo protein binder design with deep learning. *Nature Communications*, 14(1):2625, 2023. doi: 10.1038/s41467-023-38328-5. URL https://doi.org/10.1038/s41467-023-38328-5.

Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.235. URL https://doi.org/10.1093/nar/28.1.235.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=YCWjhGrJFD.

Jasper Butcher, Rohith Krishna, Raktim Mitra, Rafael I. Brent, Yanjing Li, Nathaniel Corley, Paul Kim, Jonathan Funk, Simon Mathis, Saman Salike, Aiko Muraishi, Helen Eisenach, Tuscan Rock Thompson, Jie Chen, Yuliya Politanska, Enisha Sehgal, Brian Coventry, Odin Zhang, Bo Qiang, Kieran Didi, Max Kazman, Frank DiMaio, and David Baker. De novo design of all-atom biomolecular interactions with rfdiffusion3. *bioRxiv*, 2025. doi: 10.1101/2025.09.18.676967. URL https://www.biorxiv.org/content/early/2025/09/18/2025.09.18.676967.

Longxing Cao, Brian Coventry, Inna Goreshnik, Buwei Huang, William Sheffler, Joon Sung Park, Kevin M. Jude, Iva Marković, Rameshwar U. Kadam, Koen H. G. Verschueren, Kenneth Verstraete, Scott Thomas Russell Walsh, Nathaniel Bennett, Ashish Phal, Aerin Yang, Lisa Kozodoy, Michelle DeWitt, Lora Picton, Lauren Miller, Eva-Maria Strauch, Nicholas D. DeBouver, Allison Pires, Asim K. Bera, Samer Halabiya, Bradley Hammerson, Wei Yang, Steffen Bernard, Lance Stewart, Ian A. Wilson, Hannele Ruohola-Baker, Joseph Schlessinger, Sangwon Lee, Savvas N. Savvides, K. Christopher Garcia, and David Baker. Design of protein-binding proteins from the target structure alone. *Nature*, 605(7910):551–560, 2022. doi: 10.1038/s41586-022-04654-9. URL https://doi.org/10.1038/s41586-022-04654-9.

Yehlin Cho, Martin Pacesa, Zhidian Zhang, Bruno E. Correia, and Sergey Ovchinnikov. Boltzdesign1: Inverting all-atom structure prediction model for generalized biomolecular binder design. *bioRxiv*, 2025. doi: 10.1101/2025.04.06.647261. URL https://www.biorxiv.org/content/early/2025/04/06/2025.04.06.647261.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J. Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1vmSEVL19f.

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2026/09/20 2022. doi: 10.1126/science.add2187. URL https://doi.org/10.1126/science.add2187.

Kieran Didi, Zuobai Zhang, Guoqing Zhou, Danny Reidenbach, Zhonglin Cao, Sooyoung Cha, Tomas Geffner, Christian Dallago, Jian Tang, Michael M. Bronstein, Martin Steinegger, Emine Kucukbenli, Arash Vahdat, and Karsten Kreis. Scaling atomistic protein binder design with generative pretraining and test-time compute. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qmCpJtFZra.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control.

In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xQBRrtQM8u.

Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, Olaf Ronneberger, Sebastian Bodenstein, Michal Zielinski, Alex Bridgland, Anna Potapenko, Andrew Cowie, Kathryn Tunyasuvunakool, Rishub Jain, Ellen Clancy, Pushmeet Kohli, John Jumper, and Demis Hassabis. Protein complex prediction with alphafold-multimer. *bioRxiv*, 2022. doi: 10.1101/2021.10.04.463034. URL https://www.biorxiv.org/content/early/2022/03/10/2021.10.04.463034.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 79858–79885. Curran Associates, Inc., 2023. doi: 10.52202/075280-3497. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/fc65fab891d83433bd3c8d966edde311-Paper-Conference.pdf.

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*, 2025. doi: 10.64898/2025.12.28.696754. URL https://www.biorxiv.org/content/early/2025/12/28/2025.12.28.696754.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Guillaume Huguet, James Vuckovic, Kilian Fatras, Eric Thibodeau-Laufer, Pablo Lemos, Riashat Islam, Cheng-Hao Liu, Jarrid Rector-Brooks, Tara Akhound-Sadegh, Michael Bronstein, Alexander Tong, and Avishek Joey Bose. Sequence-augmented se(3)-flow matching for conditional protein generation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 33007–33036. Curran Associates, Inc., 2024. doi: 10.52202/079017-1039. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/39ca8893ea38905a9d2ffe786e85af0f-Paper-Conference.pdf.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577. Curran Associates, Inc., 2022. doi: 10.52202/068431-1926. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/a98846e9d9cc01cfb87eb694d946ce6b-Paper-Conference.pdf.

Andreas H. Laustsen, Victor Greiff, Aneesh Karatt-Vellatt, Serge Muyldermans, and Timothy P. Jenkins. Animal immunization, in vitro display technologies, and machine learning for antibody discovery. *Trends in Biotechnology*, 39(12):1263–1273, 2026/09/20 2021. doi: 10.1016/j.tibtech.2021.03.003. URL https://doi.org/10.1016/j.tibtech.2021.03.003.

Simon Levine-Gottreich, Paulina Kulytė, Jacob Stern, David Grayson, Jonathan Edward King, and Umberto Lupo. Origin-1: Experimentally validated generative AI platform for de novo antibody design against "zero-prior" epitopes. In *ICLR 2026 Workshop on Generative and Experimental Perspectives for Biomolecular Design*, 2026. URL https://openreview.net/forum?id=ZPBtdB0icX.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Cong Liu, Milong Ren, Jiaqi Guan, Chengyue Gong, Jinyuan Sun, Xinshi Chen, and Wenzhi Xiao. ProtDBench: A unified benchmark of protein binder design and evaluation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=bWLfplRNzt.

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. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 40783–40818. Curran Associates, Inc., 2025. doi: 10.52202/085713-1362. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/3a10c46572628d58cb44fb705f25cbbf-Paper-Conference.pdf.

Luis S. Mille-Fragoso, John N. Wang, Claudia L. Driscoll, Haoyu Dai, Talal Widatalla, Xiaowei Zhang, Brian L. Hie, and Xiaojing J. Gao. Efficient generation of epitope-targeted de novo antibodies with germinal. *bioRxiv*, 2025. doi: 10.1101/2025.09.19.677421. URL https://www.biorxiv.org/content/early/2025/09/24/2025.09.19.677421.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Martin Pacesa, Lennart Nickel, Christian Schellhaas, Joseph Schmidt, Ekaterina Pyatova, Lucas Kissling, Patrick Barendse, Jagrity Choudhury, Srajan Kapoor, Ana Alcaraz-Serna, Yehlin Cho, Kourosh H. Ghamary, Laura Vinué, Brahm J. Yachnin, Andrew M. Wollacott, Stephen Buckley, Adrie H. Westphal, Simon Lindhoud, Sandrine Georgeon, Casper A. Goverde, Georgios N. Hatzopoulos, Pierre Gönczy, Yannick D. Muller, Gerald Schwank, Daan C. Swarts, Alex J. Vecchio, Bernard L. Schneider, Sergey Ovchinnikov, and Bruno E. Correia. One-shot design of functional protein binders with bindcraft. *Nature*, 646(8084):483–492, 2025. doi: 10.1038/s41586-025-09429-6. URL https://doi.org/10.1038/s41586-025-09429-6.

Wei Qu, Jiawei Guan, Rui Ma, kezhai, Weikun.Wu, and Haobo Wang. P(all-atom) is unlocking new path for protein design. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=yXRixu0ONY.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. doi: 10.52202/075280-2338. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv*, 2024. URL https://arxiv.org/abs/2402.03300.

Fang Sheng, Mohammad Noaeen, and Zahra Shakeri. Prodcarl: Reinforcement learning-aligned diffusion models for de novo antimicrobial peptide design. *arXiv*, 2026. URL https://arxiv.org/abs/2602.00157.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Hannes Stark, Felix Faltings, MinGyu Choi, Yuxin Xie, Eunsu Hur, Timothy O'Donnell, Anton Bushuiev, Talip Uçar, Saro Passaro, Weian Mao, Mateo Reveiz, Roman Bushuiev, Tomáš Pluskal, Josef Sivic, Karsten Kreis, Arash Vahdat, Shamayeeta Ray, Jonathan T. Goldstein, Andrew Savinov, Jacob A. Hambalek, Anshika Gupta, Diego A. Taquiri-Diaz, Yaotian Zhang, A. Katherine Hatstat, Angelika Arada, Nam Hyeong Kim, Ethel Tackie-Yarboi, Dylan Boselli, Lee Schnaider, Chang C. Liu, Gene-Wei Li, Denes Hnisz, David M. Sabatini, William F. DeGrado, Jeremy Wohlwend, Gabriele Corso, Regina Barzilay, and Tommi Jaakkola. Boltzgen: Toward universal binder design. *bioRxiv*, 2025. doi: 10.1101/2025.11.20.689494. URL https://www.biorxiv.org/content/early/2025/11/24/2025.11.20.689494.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017. doi: 10.1038/nbt.3988. URL https://doi.org/10.1038/nbt.3988.

Filippo Stocco, Maria Artigues-Lleixa, Andrea Hunklinger, Michele Garibbo, Talal Widatalla, Marc Guell, and Noelia Ferruz. Reinforcement learning guides generative protein language models, 2024. URL https://arxiv.org/abs/2412.12979.

Protenix Team, Milong Ren, Jinyuan Sun, Jiaqi Guan, Cong Liu, Chengyue Gong, Yuzhe Wang, Lan Wang, Qixu Cai, Xinshi Chen, and Wenzhi Xiao. Pxdesign: Fast, modular, and accurate de novo design of protein binders. *bioRxiv*, 2025. doi: 10.1101/2025.08.15.670450. URL https://www.biorxiv.org/content/early/2025/08/16/2025.08.15.670450.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238, June 2024.

Ziwen Wang, Jiajun Fan, Ruihan Guo, Thao Nguyen, Heng Ji, and Ge Liu. Proteinzero: Self-improving protein generation via online reinforcement learning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=pVvm0CQAr6.

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, Basile I. M. Wicky, Nikita Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sappington, Susana Vázquez Torres, Anna Lauko, Valentin De Bortoli, Emile Mathieu, Sergey Ovchinnikov, Regina Barzilay, Tommi S. Jaakkola, Frank DiMaio, Minkyung Baek, and David Baker. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023. doi: 10.1038/s41586-023-06415-8. URL https://doi.org/10.1038/s41586-023-06415-8.

Yibo Wen, Chenwei Xu, Jerry Yao-Chieh Hu, Kaize Ding, and Han Liu. Pareto-optimal energy alignment for designing nature-like antibodies. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 46294–46321. Curran Associates, Inc., 2025. doi: 10.52202/085713-1541. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/41f994cacb08c9d2efc9340117e454f0-Paper-Conference.pdf.

Julius Wenckstern, Anna M. Díaz-Rovira, Julia Kuhn, Arvid Ban, Rahma Hamdani, Roser Pruaño-Milla, Evgenia Elizarova, Sandrine Georgeon, Kaiden Thompson, Matthias Hinterndorfer, Devanarayanan Siva Sankar, Maximilian Dünnebacke, David Desscan, Sreenath Nair, Marcelo Querino Lima Afonso, Jennifer Fleming, Sameer Velankar, Andrea Ablasser, Paola Picotti, Georg Winter, Mikko Taipale, and Bruno E. Correia. The human bindome: A proteome-scale atlas of designed binder candidates. *bioRxiv*, 2026. doi: 10.64898/2026.07.30.741542. URL https://www.biorxiv.org/content/early/2026/07/30/2026.07.30.741542.

Talal Widatalla, Ashir A. Borah, Samuel H. King, Claudia L. Driscoll, Rafael Rafailov, and Brian L. Hie. Aligning protein-generative models to experimental fitness with proteindpo. *Nature Methods*, 23(9):1805–1813, 2026. doi: 10.1038/s41592-026-03137-3. URL https://doi.org/10.1038/s41592-026-03137-3.

Shuchen Xue, Chongjian GE, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=nLY2pOYBrJ.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, and Ping Luo. Dancegrpo: Unleashing grpo on visual generation. *arXiv*, 2025. URL https://arxiv.org/abs/2505.07818.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Xinwu Ye, He Cao, Hao Li, Bin Feng, Zijing Liu, Xiangru Tang, Yu Li, and Shenghua Gao. Pushing biomolecular utility-diversity frontiers with supergroup relative policy optimization. *arxiv*, 2026. URL https://arxiv.org/abs/2605.08659.

Vinicius Zambaldi, David La, Alexander E. Chu, Harshnira Patani, Amy E. Danson, Tristan O. C. Kwan, Thomas Frerix, Rosalia G. Schneider, David Saxton, Ashok Thillaisundaram, Zachary Wu, Isabel Moraes, Oskar Lange, Eliseo Papa, Gabriella Stanton, Victor Martin, Sukhdeep Singh, Lai H. Wong, Russ Bates, Simon A. Kohl, Josh Abramson, Andrew W. Senior, Yilmaz Alguel, Mary Y. Wu, Irene M. Aspalter, Katie Bentley, David L. V. Bauer, Peter Cherepanov, Demis Hassabis, Pushmeet Kohli, Rob Fergus, and Jue Wang. De novo design of high-affinity protein binders with alphaproteo. *arXiv*, 2024. URL https://arxiv.org/abs/2409.08022.

Odin Zhang, Xujun Zhang, Haitao Lin, Cheng Tan, Qinghan Wang, Yuanle Mo, Qiantai Feng, Gang Du, Yuntao Yu, Zichang Jin, Ziyi You, Peicong Lin, Yijie Zhang, Yuyang Tao, Shicheng Chen, Jack Xiaoyu Chen, Chenqing Hua, Weibo Zhao, Runze Ma, Yunpeng Xia, Kejun Ying, Jun Li, Yundian Zeng, Lijun Lang, Peichen Pan, Hanqun Cao, Zihao Song, Bo Qiang, Jiaqi Wang, Pengfei Ji, Lei Bai, Jian Zhang, Chang yu Hsieh, Pheng Ann Heng, Siqi Sun, Tingjun Hou, and Shuangjia Zheng. Odesign: A world model for biomolecular interaction design. *arXiv*, 2025. URL https://arxiv.org/abs/2510.22304.

Xiangxin Zhou, Dongyu Xue, Ruizhe Chen, Zaixiang Zheng, Liang Wang, and Quanquan Gu. Antigen-specific antibody design via direct energy-based preference optimization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 120861–120891. Curran Associates, Inc., 2024. doi: 10.52202/079017-3840. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/daef77101ba5711084a57442c8cf2709-Paper-Conference.pdf.

A Method and Training Details

A.1 Policy formulation and training objective

Sampling and policy transitions. For target j, the frozen conditioning network, including the Pairformer, encodes the target sequence and structure, hotspot residues, and requested binder length into c_j. With c_j fixed, the diffusion model denoises target and binder coordinates jointly. Binder coordinates include auxiliary atom slots that encode residue identity.

Let $\sigma_T > \cdots > \sigma_0 = 0$ be the noise schedule for T denoising steps. Starting from initial noise $x_T \sim \mathcal{N}(0, \sigma_T^2 I)$, the sampler iterates over $t = T, T-1, \ldots, 1$ and returns the clean design x_0. At step t, the BoltzGen sampler centers the valid coordinates, applies a random rigid transform $g_t = (\mathbf{R}_t, v_t)$, and adds Gaussian noise:

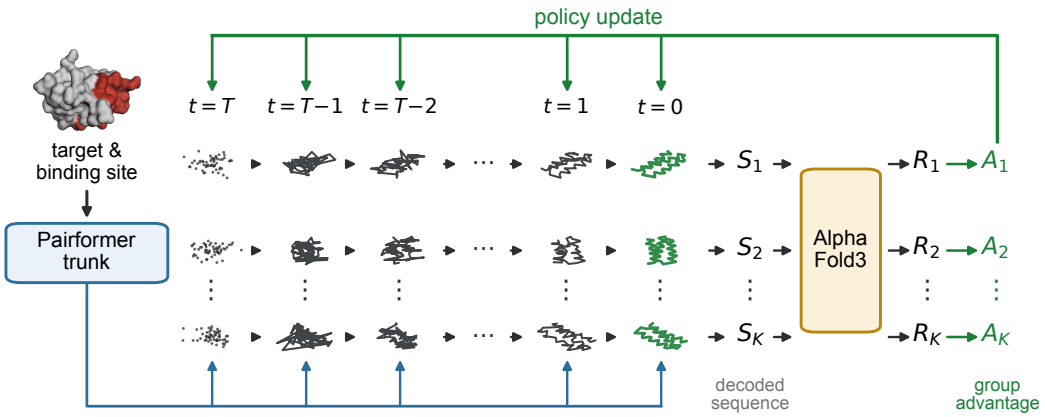


Figure 3: **Sampling and training.** Conditioning is shared across trajectories for a target. Each trajectory produces a co-designed sequence and structure. AlphaFold3 evaluates the decoded sequence with the target, and the resulting advantage is assigned to the recorded stochastic transitions. Their losses are combined into one parameter update.

$$
\begin{aligned}
\gamma_t &= \gamma_{\text{churn}} \mathbf{1}\{\sigma_{t-1} > \sigma_{\text{churn}}\}, \\
\widehat{\sigma}_t &= \sigma_t(1+\gamma_t), \\
b_t &= \eta^2 \sqrt{\widehat{\sigma}_t^2 - \sigma_t^2}, \\
\widehat{x}_t &= \text{ctr}(x_t)\mathbf{R}_t + v_t + b_t \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I).
\end{aligned}
\tag{8}
$$

The model predicts clean coordinates $D_{\theta,t} = D_\theta(\widehat{x}_t, \widehat{\sigma}_t; c_j)$. Before the Euler update, the Kabsch algorithm rotates and translates the noisy complex to minimize squared distances to the prediction over all valid target and binder coordinate slots. The same rigid transform is applied to the target and binder, preserving their relative arrangement:

$$
\begin{aligned}
\widetilde{x}_t &= \text{Align}(\widehat{x}_t, D_{\theta,t}), \\
x_{t-1} &= \widetilde{x}_t + \alpha(\sigma_{t-1} - \widehat{\sigma}_t)\frac{\widetilde{x}_t - D_{\theta,t}}{\widehat{\sigma}_t} =: E_{\theta,t}(\widehat{x}_t, c_j).
\end{aligned}
\tag{9}
$$

This superposition removes global translation and rotation differences before computing the denoising direction. It is recomputed for each prediction, but gradients do not propagate through the aligned coordinates $\widetilde{x}_t$. They still propagate through $D_{\theta,t}$ in the Euler update.

A policy transition connects consecutive noisy denoiser inputs. The Euler update at step t produces x_{t-1}. Augmentation and noise injection at the next step, indexed by $t-1$, produce $\widehat{x}_{t-1}$. For $t \in \{2, \ldots, T\}$ with $b_{t-1} > 0$, the transition density is

$$
\begin{aligned}
\pi_\theta(\widehat{x}_{t-1} \mid \widehat{x}_t, c_j, g_{t-1}) &= \mathcal{N}(\widehat{x}_{t-1}; \mu_{\theta,t-1}, b_{t-1}^2 I), \\
\mu_{\theta,t-1} &= \text{ctr}(E_{\theta,t}(\widehat{x}_t, c_j))\,\mathbf{R}_{t-1} + v_{t-1}.
\end{aligned}
\tag{10}
$$

The state consists of c_j, the time index t, and $\widehat{x}_t$; the action is $\widehat{x}_{t-1}$. The rigid transform has a parameter-independent distribution $q(g)$. The augmented transition density is $q(g_{t-1})\pi_\theta$, and q cancels when policies are compared using the same sampled rigid transformations. Gaussian noise is added after centering, so the next noisy state does not need to remain centered.

The transitions included in the loss are

$$\mathcal{I} = \{t \in \{2, \dots, T\} : b_{t-1} > 0\}. \tag{11}$$

Initialization, zero-noise transitions, and the final Euler output contribute no direct log-density terms. All sampling steps are executed to obtain the terminal design. The 40-step schedule in Table 4 yields 33 recorded stochastic transitions.

Binder-coordinate density and reference regularization. Let $\mathcal{B}_j$ contain the valid binder coordinate slots, including auxiliary atoms but excluding padding. For trajectory k, the coordinate-averaged log-density is

$$\bar{\ell}_{\theta,jkt} = \frac{1}{3|\mathcal{B}_j|} \sum_{a \in \mathcal{B}_j} \sum_{d=1}^{3} \log \mathcal{N}\big(\widehat{x}_{j,k,t-1,a,d}; \mu_{\theta,jkt,a,d}, b_{t-1}^2\big)\,. \tag{12}$$

Here $\mu_{\theta,jkt}$ is the mean of the transition from step t to step $t-1$ for target j and trajectory k. The ratio used in the clipped objective is

$$\bar{\rho}_{\theta,jkt} = \exp\big(\bar{\ell}_{\theta,jkt} - \bar{\ell}_{\theta_{\mathrm{old}},jkt}\big)\,. \tag{13}$$

The replay batch was sampled using model parameters θ_{old}. Following Flow-GRPO (Liu et al., 2025), we average log-densities over binder coordinates. The resulting ratio is the geometric mean of the coordinate-wise likelihood ratios, rather than an importance ratio for the full complex. We set the clipping range to $\varepsilon = 10^{-3}$, as in the Flow-GRPO reference implementation (Table 4).

Because the current and pretrained reference policies share the same transition variance, their KL divergence per binder coordinate is

$$\bar{D}_{jkt}^{\mathrm{KL}} = \frac{1}{3|\mathcal{B}_j|} \sum_{a \in \mathcal{B}_j} \frac{\|\mu_{\theta,jkt,a} - \mu_{\mathrm{ref},jkt,a}\|_2^2}{2b_{t-1}^2} \tag{14}$$

Advantages and loss. Let R_{jk} be the terminal AlphaFold3 ipTM for design k on target j. We normalize rewards using the mean and standard deviation of the K rewards in the current target group, then clip the advantages to $[-3, 3]$:

$$\begin{aligned} m_j &= \frac{1}{K} \sum_{k=1}^{K} R_{jk}, \\ s_j &= \sqrt{\frac{1}{K} \sum_{k=1}^{K} (R_{jk} - m_j)^2}, \\ A_{jk} &= \mathrm{clip}\left(\frac{R_{jk} - m_j}{s_j + \delta}, -3, 3\right). \end{aligned} \tag{15}$$

$$\mathcal{L}_{jkt}^{\mathrm{policy}} = -\min[\bar{\rho}_{\theta,jkt} A_{jk}, \mathrm{clip}(\bar{\rho}_{\theta,jkt}, 1-\varepsilon, 1+\varepsilon) A_{jk}]\,. \tag{16}$$

For M target groups, the full loss is

$$\mathcal{L}(\theta) = \frac{1}{M} \sum_{j=1}^{M} \frac{1}{K} \sum_{k=1}^{K} \frac{1}{|\mathcal{I}|} \sum_{t \in \mathcal{I}} \Big[\mathcal{L}_{jkt}^{\mathrm{policy}} + \lambda_{\mathrm{KL}} \bar{D}_{jkt}^{\mathrm{KL}}\Big]\,. \tag{17}$$

Each recorded transition receives the same terminal advantage for its trajectory. We average log-densities over binder coordinates, stop gradients through alignment, and train only on recorded stochastic transitions. These choices define a training surrogate rather than the full policy gradient of the sampler.

Table 4: **Training configuration.**

Category	Setting	Value
Model	Initial checkpoint	BoltzGen diverse
Model	Trainable parameters	Diffusion module
Rollout	Designs per target, K	32
Rollout	Targets per optimizer update, M	8
Rollout	Designs per optimizer update	256
Rollout	Training schedule	500 updates; four epochs
Sampler	Denoising steps, T	40
Sampler	Euler step scale, α	1.8
Sampler	Noise scale, η	0.95
Sampler	Noise schedule	Dilated power-law schedule
Sampler	Dimensionless schedule endpoints	$s_{\max} = 160$; $s_{\min} = 0.0004$
Sampler	Data scale and power	$\sigma_{\text{data}} = 16$; $\rho = 7$
Sampler	Dilation factor and interval	$h = 2.667$; $[u, v] = [0.6, 0.8]$
Sampler	Churn magnitude and cutoff	$\gamma_{\text{churn}} = 0.8$; $\sigma_{\text{churn}} = 1$
Sampler	Recorded transitions	All positive-noise transitions; 33 per trajectory
Reward	Scalar reward	Raw AlphaFold3 ipTM
Reward	AlphaFold3 diffusion steps	2
Reward	AlphaFold3 step scale	1.0
Reward	AlphaFold3 recycles	10
Reward	Samples per design	1
Objective	Ratio clipping range, ε	10^{-3}
Objective	Reference KL coefficient, λ_{KL}	0.01
Objective	Advantage normalization	Current target group
Objective	Advantage clipping bound, $A_{\max}$	3
Objective	Advantage denominator offset, δ	10^{-4}
Optimizer	AdamW learning rate	3×10^{-5}
Optimizer	Weight decay	10^{-4}
Optimizer	Gradient-norm clipping bound	1.0
Validation	Regular evaluation interval	20 updates
Validation	Additional evaluations	4 designs per target at updates 375 and 500

A.2 Training configuration

We initialize the policy from the BoltzGen diverse checkpoint (Stark et al., 2025): `huggingface:boltzgen/boltzgen1_diverse:boltzgen1_diverse.ckpt`. Only the diffusion module is trained; the conditioning network and the prediction heads remain frozen. Table 4 lists the hyperparameters.

Reward validation. Training rewards use AlphaFold3 with two diffusion steps and a step scale of 1.0, compared with 200 steps and a step scale of 1.5 for benchmark evaluation; both use ten recycles. We compared these settings on 100 ProteinMPNN sequences per Cao target, assigned to pretrained BoltzGen backbones, keeping sequences, cached target MSAs, and model seeds fixed (Figure 4). Across 1,100 paired predictions, ipTM had a Spearman correlation of $\rho = 0.971$, a median within-target correlation of $\rho = 0.967$, and a near-constant offset of -0.047. The median binder Cα RMSD was 0.50 Å, with 91% of pairs below 2 Å.

The BoltzGen sampler uses the dilated noise schedule:

$$\sigma_t = \sigma_{\text{data}} \left[s_{\max}^{1/\rho} + f\left(\frac{T-t}{T-1}\right) \left(s_{\min}^{1/\rho} - s_{\max}^{1/\rho}\right) \right]^{\rho}, \qquad t = 1, \ldots, T, \qquad \sigma_0 = 0. \tag{18}$$

Writing $w = v - u$ and $\kappa = (1 - hw)/(1 - w)$, the dilation map for normalized sampling progress $\tau \in [0, 1]$ is

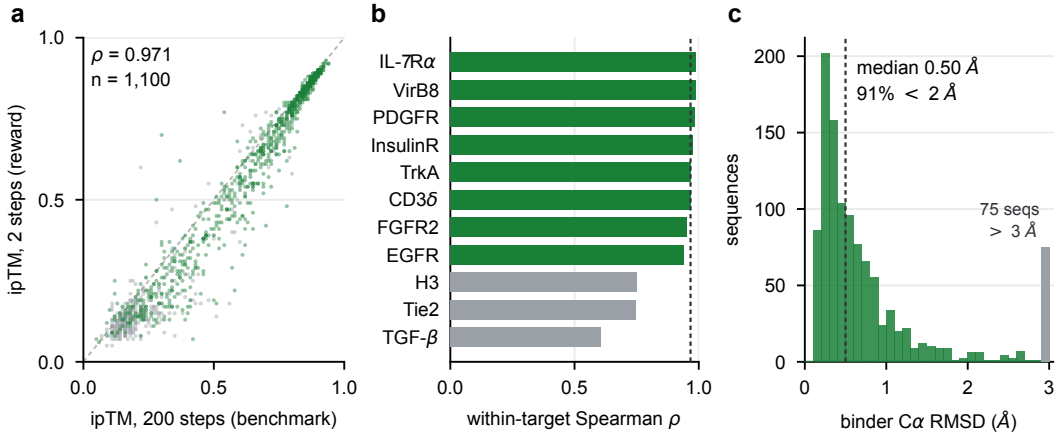


Figure 4: **Reward validation.** Paired AlphaFold3 predictions for 100 ProteinMPNN sequences per Cao target on pretrained BoltzGen backbones. The settings use 2 or 200 diffusion steps at step scales 1.0 or 1.5, respectively; both use ten recycles and identical sequences, cached target MSAs, and seeds. (a) Paired ipTM; grey marks targets on which no method succeeds. (b) Within-target Spearman correlation; the dashed line marks the median. (c) Binder Cα RMSD, with the 75 pairs above 3 Å shown as the grey bar.

$$f(\tau) = \begin{cases} \tau/\kappa, & \tau < \kappa u, \\ u + (\tau - \kappa u)/h, & \kappa u \le \tau < \kappa u + hw, \\ v + (\tau - \kappa u - hw)/\kappa, & \tau \ge \kappa u + hw. \end{cases} \tag{19}$$

The terminal coordinates are decoded into a binder sequence, with unassigned residues replaced by glycine. AlphaFold3 receives the target sequence and cached features, followed by the binder sequence without searched MSAs or templates (Abramson et al., 2024; Evans et al., 2022).

B Data and Evaluation

B.1 Candidate target pools and data splits

Target and hotspot definitions. A target input comprises a protein sequence and structure, hotspot residues, and a requested binder length. We characterized eligible training candidates from the PDB and Human Bindome.

For PDB complexes (Berman et al., 2000), the hotspot contains target residues with a heavy atom within 5.0 Å of a single designated protein partner. The PDB hotspot filter additionally requires at least 15 validated residues. For Human Bindome targets (Wenckstern et al., 2026), it is the consensus contact set across approximately 30 BindCraft designs, retaining residues contacted in at least 30% of designs.

Characterization and selection. Each candidate is evaluated with five independently seeded groups of eight BoltzGen designs. Characterization uses 40 denoising steps, step scale 1.8, and noise scale 1.003 on the dilated schedule. Each decoded sequence is scored with AlphaFold3 using two diffusion steps, step scale 1.0, ten recycles, and one diffusion sample.

For rewards R_{jrk}, with group $r = 1, \ldots, 5$ and design $k = 1, \ldots, 8$, we calculate

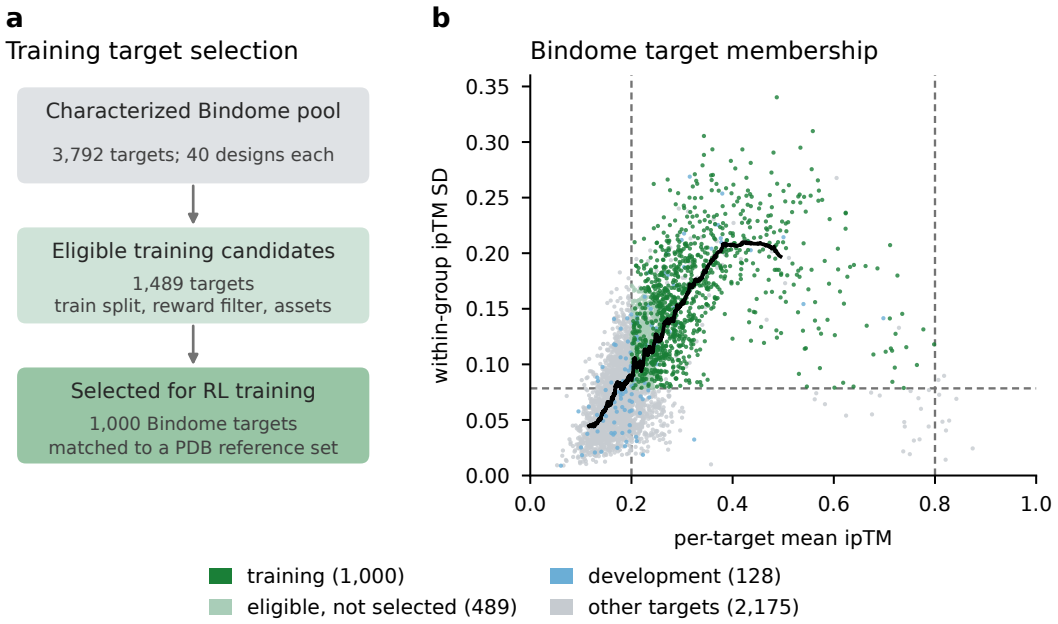


Figure 5: **Training target selection.** (a) Of 3,792 characterized Human Bindome targets, 1,489 satisfy the training-split, input-availability, and reward criteria: mean AlphaFold3 ipTM in $[0.2, 0.8)$ and within-group standard deviation of at least 0.0784. Matching to the PDB reference set selects 1,000 training targets. (b) Mean AlphaFold3 ipTM and within-group standard deviation for all 3,792 targets, each characterized with 40 designs. Green marks training targets, light green marks eligible but unselected targets, blue marks Human Bindome Validation targets, and gray marks other candidates. Dashed lines show the reward thresholds. The black line is a running mean.

$$
\begin{aligned}
m_j^{\text{scan}} &= \frac{1}{40}\sum_{r=1}^{5}\sum_{k=1}^{8} R_{jrk}, \\
s_j^{\text{scan}} &= \frac{1}{5}\sum_{r=1}^{5}\sqrt{\frac{1}{8}\sum_{k=1}^{8}(R_{jrk}-\bar{R}_{jr})^2}, \\
\bar{R}_{jr} &= \frac{1}{8}\sum_{k=1}^{8} R_{jrk}.
\end{aligned}
\tag{20}
$$

A target passes the reward filter when $m_j^{\text{scan}} \in [0.2, 0.8)$ and $s_j^{\text{scan}} \geq 0.0784$. The second statistic measures variation within eight-design groups. After the sequence-based split described below and checks for the required target, hotspot, and AlphaFold3 inputs, 1,489 Human Bindome targets are eligible for training.

We select 1,000 out of 1,489 Human Bindome Targets. Only the selected Human Bindome targets are used for training.

Sequence separation and validation sets. We jointly cluster PDB and Human Bindome target sequences with MMseqs2 at 30% identity and 50% coverage of both sequences (Steinegger & Söding, 2017). After excluding clusters matching Cao and Protdbench targets, we split the remaining clusters into training and validation sets (Widatalla et al., 2026). Validation clusters with above-threshold matches to training are moved into training until no such matches remain.

B.2 BENCHMARK PROTOCOL

Inputs and generation. The Cao benchmark contains eleven targets (Cao et al., 2022). Every generator receives the same target sequence, residue span, hotspot residues, and binder lengths. Target constructs contain 63–210 residues and requested binders contain 61–65 residues. The supplied reference binder is removed before generation. Hotspot indices are mapped to the residue numbering expected by each model.

Each generator produces 100 designs per target. Generator-specific ranking and filtering are disabled: PXDesign uses its inference stage without downstream confidence filters; Proteina-Complexa uses single-pass generation without reward-based search; all generated BoltzGen, BindGRPO, RFdiffusion3, and ODesign designs enter evaluation. This protocol evaluates generation without downstream selection. Appendix C.3 compares generated and ProteinMPNN-assigned sequences.

BoltzGen and BindGRPO use 500 denoising steps. For each target, 50 designs use Euler step scale 1.8 and noise scale 0.95, and 50 use step scale 2.0 and noise scale 0.88.

Sequence assignment. ProteinMPNN assigns eight sequences to each generated backbone using checkpoint `v_48_020`, temperature 0.1 (Dauparas et al., 2022). Only the binder sequence is redesigned; the target remains fixed as structural and sequence context.

For sequence assignment, complexes are represented with binder chain A and target chain B, assigned by their specified roles. The same protocol is used for generators that produce a sequence and those that produce only a backbone.

AlphaFold3 prediction and success criterion. AlphaFold3 predicts each target–binder complex from the target sequence with cached MSAs and the candidate binder sequence without searched MSAs (Abramson et al., 2024). Generated coordinates are not supplied to AlphaFold3. Each sequence receives one diffusion sample with 200 diffusion steps at step scale 1.5 and ten recycles. Training rewards use two AlphaFold3 diffusion steps at step scale 1.0 (Table 4); Appendix A.2 compares the two settings.

Let E_{ab} be the reported expected aligned error between residues a and b, and let $\mathcal{T}$ and $\mathcal{B}$ denote target and binder residues. The minimum interchain PAE is

$$P_{\min} = \min_{a\in\mathcal{T},\, b\in\mathcal{B}} \min(E_{ab}, E_{ba}). \tag{21}$$

We use the binder-chain pTM reported by AlphaFold3, denoted $T_{\mathcal{B}}$.

Complex RMSD is calculated over matched target and binder Cα atoms after a single rigid-body superposition of the full complex. A sequence passes when

$$P_{\min} < 1.5, \qquad T_{\mathcal{B}} > 0.8, \qquad \mathrm{RMSD}_{\mathrm{complex}} < 2.5\ \text{Å}. \tag{22}$$

These cutoffs follow the reported AlphaProteo criteria (Zambaldi et al., 2024), using the definitions given above. Passing them is a computational design criterion, not a measurement of binding.

AlphaFold2 with an initial guess. For BoltzGen and BindGRPO, the AlphaFold2 evaluation reuses the generated backbones and ProteinMPNN sequences. It uses the monomer `model_1_ptm` with the initial-guess procedure (Bennett et al., 2023) and the evaluation configuration described by AlphaProteo (Zambaldi et al., 2024). The input is the concatenated target–binder sequence without an MSA. The target is templated; the binder is not. The generated complex supplies the initial coordinates, with backbone atoms retained and residue identities set to the candidate sequence.

Interface PAE is the mean of the two target–binder PAE blocks. pLDDT is averaged over the full complex. Binder RMSD is measured after superposing the predicted binder on the generated binder. A sequence passes when

$$\mathrm{iPAE} < 7.0, \qquad \mathrm{RMSD}_{\mathrm{binder}} < 1.5\ \text{Å}, \qquad \mathrm{pLDDT} > 90. \tag{23}$$

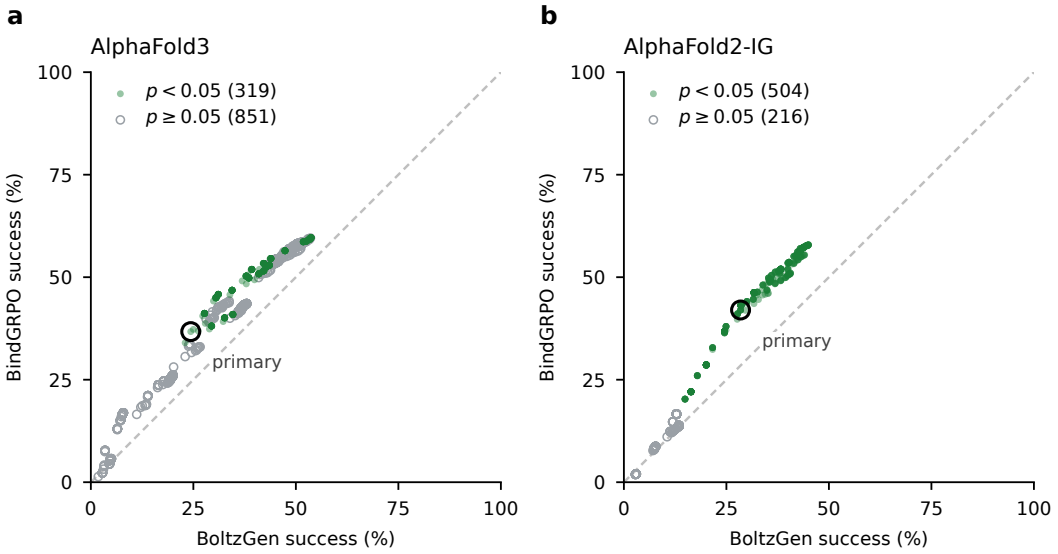


Figure 6: **Sensitivity to evaluation thresholds.** Design-level success rates for BindGRPO and BoltzGen across the retained (a) AlphaFold3 and (b) AlphaFold2 threshold grids on all eleven targets. Each point represents one threshold combination. BindGRPO is above the dashed equal-rate line at 1,145 of 1,170 AlphaFold3 settings and 672 of 720 AlphaFold2 settings; the circled point marks the primary thresholds. Open circles denote comparisons with $p \geq 0.05$ under a target-level Wilcoxon signed-rank test over the paired per-target rates. The points share evaluation samples.

The initial guess and template are distinct inputs. The generated binder is used as an initial guess, not as a binder template.

Aggregation and threshold analysis. For design d, let $Y_{d\ell}$ indicate whether sequence ℓ passes all three criteria for the chosen predictor. Design-level success is

$$Y_d^{\text{design}} = \mathbf{1}\left\{ \max_{\ell=1,\ldots,8} Y_{d\ell} = 1 \right\}. \tag{24}$$

Per-target design success is averaged over 100 backbones. Reported Cao panel means give equal weight to each of the eleven targets. Per-sequence rates use all 800 sequences per target and are reported separately from the any-of-eight design rate.

The AlphaFold3 threshold grid uses minimum interchain PAE values from 0 to 7 in increments of 0.5, binder pTM values from 0 to 1 in increments of 0.05, and complex RMSD values from 0 to 3 Å in increments of 0.5. The AlphaFold2 grid uses interface PAE values from 0 to 11 in increments of 0.5, pLDDT values from 60 to 95 in increments of 5, and binder RMSD values from 0 to 3 Å in increments of 0.5. The reported analysis retains 1,170 AlphaFold3 and 720 AlphaFold2 settings after discarding those where both models produce fewer than ten successful backbones across all eleven targets.

Because per-target Cao AlphaFold3 success rates range from 0% to 80%, designs are not exchangeable across targets. The reported comparisons therefore use two-sided exact target-level Wilcoxon signed-rank tests over the paired per-target rates, with 95% Wilson intervals on the pooled rates for descriptive purposes only. Zero differences are omitted, and tied absolute differences receive average ranks. A paired t-test over the Cao per-target rates gives $p = 0.032$. Tests are unadjusted for multiple comparisons or checkpoint selection. Threshold settings reuse the same designs and are not independent experiments. Likewise, eight sequences assigned to one backbone are not eight independent backbones. Per-sequence tests and pooled intervals should therefore not be interpreted as uncertainty over new target proteins.

Table 5: **Baseline generator results.** Design success is the percentage of 100 backbones per target for which at least one of eight ProteinMPNN sequences passes the AlphaFold3 criterion. Component pass rates and medians use 8,800 sequences per generator across all eleven targets.

	BoltzGen	**PXDesign**	**Proteina-Complexa**	**RFdiffusion3**	**ODesign**
Design success by target (%)					
CD3δ	0	3	0	5	0
EGFR	11	19	3	11	4
FGFR2	16	25	25	9	3
IL-7Rα	44	18	29	26	8
InsulinR	72	26	11	11	11
PDGFR	28	13	14	3	4
TrkA	40	53	27	25	21
VirB8	57	14	36	26	1
H3	0	0	0	0	0
TGF-β	0	0	0	0	0
Tie2	0	0	0	0	0
Mean	24.4	15.5	13.2	10.5	4.7
Component pass rate, per sequence (%)					
Binder pTM > 0.8	28.3	9.1	18.0	13.0	2.5
Minimum interchain PAE < 1.5	18.8	15.8	17.0	10.3	8.1
Complex RMSD < 2.5 Å	40.1	40.8	30.4	27.9	23.1
Median value, per sequence					
Binder pTM	0.780	0.750	0.750	0.740	0.660
Complex RMSD (Å)	5.20	4.21	7.94	7.66	9.08
Minimum interchain PAE	5.07	5.59	5.94	7.39	7.96

B.2.1 Additional targets

We also evaluated the pretrained generators and BindGRPO on an additional seven-target panel. Four targets contain multiple chains: an influenza H1 haemagglutinin HA1–HA2 protomer, the VEGF-A homodimer, the IL-17A homodimer, and the TNF-α homotrimer. We retain the full target assembly and evaluate minimum interchain PAE between the binder and all target chains. For symmetric target copies, complex RMSD uses the best compatible target-chain permutation.

PXDesign reached 19% on VEGF-A and 7% on TNF-α. On VEGF-A, its median binder pTM and complex RMSD were 0.840 and 1.80 Å. None of the evaluated pretrained generators succeeded on IL-17A.

For BoltzGen and BindGRPO, this panel uses the same 500-step generation and ProteinMPNN sequence-assignment protocol as the Cao benchmark, with 100 backbones and 800 sequences per target. All $5{,}600$ AlphaFold3 predictions per model are included and scored with the three-part AlphaFold3 criterion defined above. Backbone success increases from $148/700$ (21.14%) for BoltzGen to $167/700$ (23.86%) for BindGRPO, a gain of 2.71 percentage points. Sequence success increases from $903/5{,}600$ (16.13%) to $1{,}017/5{,}600$ (18.16%), a gain of 2.04 percentage points.

The backbone success rate increases on four targets, is unchanged on two, and decreases on TNF-α (2% to 0%; Table 1). PD-L1 accounts for 16 of the net increase of 19 successful backbones. Neither model succeeds on H1 or IL-17A. The difference is not significant under the target-level Wilcoxon signed-rank test ($p = 0.313$).

C Additional Benchmark Results

C.1 Baseline results by target

Table 5 reports design success and component metrics for the five pretrained generators under the protocol in Appendix B.2.

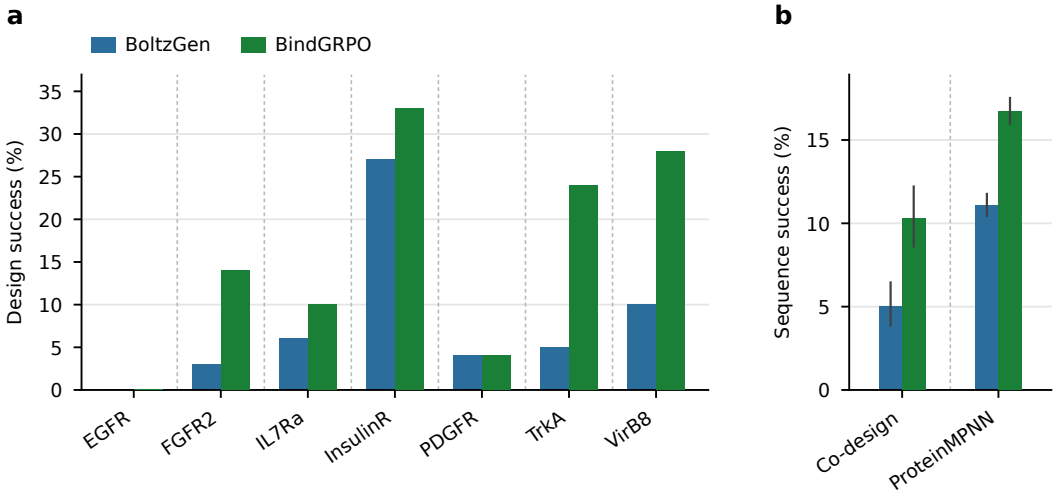


Figure 7: **Co-designed and ProteinMPNN sequences.** (a) Co-design success by target for BoltzGen and BindGRPO. CD3d, H3, TGF-β, and Tie2 are omitted because neither model succeeds with either sequence source; EGFR remains because ProteinMPNN succeeds there. (b) Panel rates for both sequence sources, pooled over all eleven targets. Co-design uses 1,100 sequences from 1,100 backbones per model; ProteinMPNN uses 8,800 sequences from those backbones. ProteinMPNN rates are per sequence, not any-of-eight. The displayed 95% Wilson intervals do not account for dependence among sequences from a shared backbone.

C.2 Evaluation of co-designed sequences

We also evaluate the binder sequence decoded directly from each BoltzGen or BindGRPO design, without ProteinMPNN. Each backbone contributes one sequence and receives the same AlphaFold3 prediction and scoring protocol as in Appendix B.2.

Across all eleven targets, co-designed sequences pass at 5.0% for BoltzGen (55/1100) and 10.3% for BindGRPO (113/1100). This is a $2.05\times$ increase over BoltzGen, but the difference is not significant at the target level (Wilcoxon $p = 0.063$) and is driven mainly by TrkA and VirB8.

ProteinMPNN per-sequence rates are 11.1% and 16.7%, respectively. The rate ratios relative to co-design are 2.22 and 1.63. These ratios compare per-sequence success pooled over all eleven targets.

C.3 Native sequences of the baseline generators

The protocol of Appendix B.2 reassigns sequences to every backbone with ProteinMPNN. Three baselines also generate their own sequences, which we evaluate under the same protocol as a control. PXDesign is excluded because its designs carry no residue identities.

Native sequences yield lower success rates for all three generators (Table 6). Proteina-Complexa reaches 2.9% against 13.2% with ProteinMPNN. RFdiffusion3 produces one passing native design, and ODesign produces none. The target-level differences are significant for Proteina-Complexa ($p = 0.016$), ODesign ($p = 0.016$), and RFdiffusion3 ($p = 0.008$). For each generator, native sequences produce no more successful designs than ProteinMPNN on any target.

Each backbone has one native sequence but eight ProteinMPNN-assigned sequences. Per-sequence success is higher with ProteinMPNN for all three generators. Sequence reassignment improves all generators for which native sequences are available, including BindGRPO.

Table 6: **Native versus reassigned sequences for the baseline generators.** Native uses the one sequence each generator emits per backbone; ProteinMPNN uses the eight assigned to it. Per-design counts a backbone as successful if any of its sequences passes. Brackets give 95% Wilson intervals; p-values are target-level Wilcoxon signed-rank tests on the eleven per-target counts.

	Native		ProteinMPNN		
Generator	per design	per sequence	per design	per sequence	p
Proteina-Complexa	$2.9_{[2.1,\,4.1]}$	2.9	$13.2_{[11.3,\,15.3]}$	4.4	0.016
RFdiffusion3	$0.1_{[0.0,\,0.5]}$	0.1	$10.5_{[8.9,\,12.5]}$	3.1	0.008
ODesign	$0.0_{[0.0,\,0.3]}$	0.0	$4.7_{[3.6,\,6.1]}$	1.0	0.016

D Optimization ablations

We compare training pool, reference regularization, learning rate, and batch allocation. Let M denote targets per update and K designs per target.

Training-pool comparison. In a separate comparison of full and reward-filtered PDB target pools, the filtered-pool run ended with higher mean AlphaFold3 ipTM on the validation set. (0.361 versus 0.328, averaged over the last 5 checkpoints).

Reference regularization. Without reference regularization ($\lambda_{\mathrm{KL}} = 0$), validation ipTM increased to 0.5765, while Cao backbone success fell to 22.00%. The regularized runs had lower validation reward but higher Cao success at update 500 (Figure 8). Validation reward alone therefore did not reliably track benchmark success.

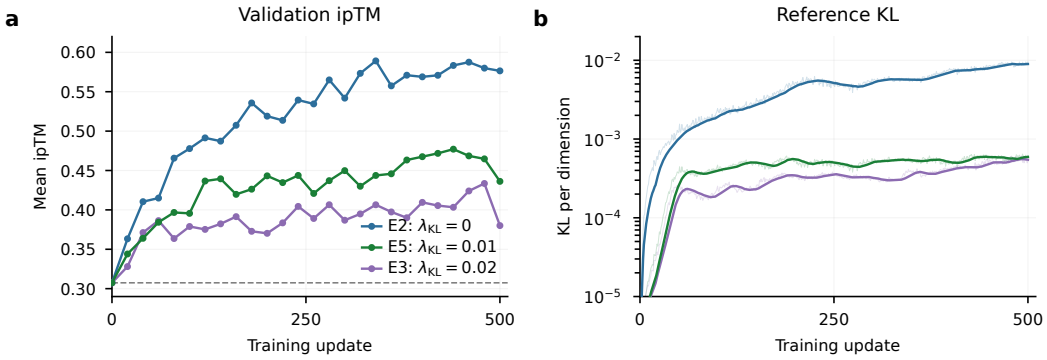


Figure 8: **Reference regularization.** (a) Validation ipTM; the dashed line marks pretrained BoltzGen. (b) Unweighted reference KL per binder coordinate dimension, measured before each update on each run's sampled states. Thin lines show individual updates; thick lines show trailing 20-update means.

Learning rate and group size. With $M = 8$, we compared two constant learning rates and two group sizes over 250 updates (Figure 9). At $K = 16$, the higher learning rate reduced both final and late-average ipTM. At $K = 32$, it improved the final score, but the late averages were nearly identical. Increasing K doubled the sampling budget.

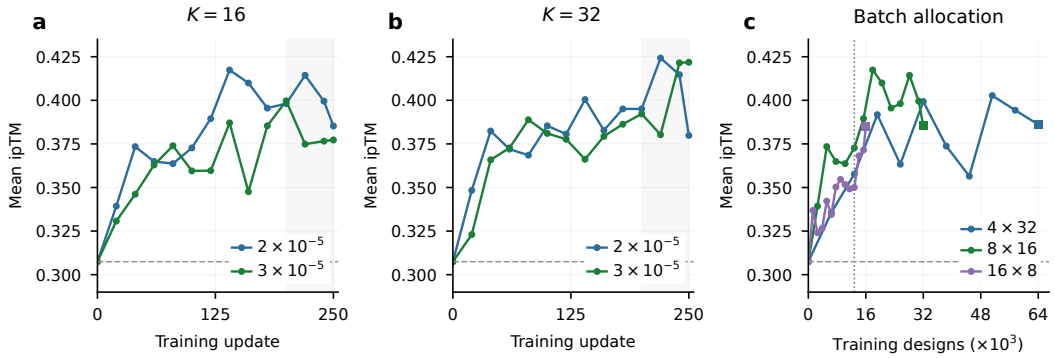


Figure 9: **Learning rate and batch allocation.** Validation ipTM against (a,b) training updates and (c) training designs. (a,b) Learning-rate comparisons at $M = 8$ and $K = 16$ or 32; shading marks the late-evaluation interval. (c) Batch allocation at $MK = 128$ and learning rate 2×10^{-5}. Legend entries give $M \times K$. The vertical dotted line marks 12,800 designs; square endpoints mark 2,000 target visits. Horizontal dashed lines mark pretrained BoltzGen.

Batch allocation. With $MK = 128$ designs per update and a learning rate of 2×10^{-5}, the $(M, K) = (8, 16)$ allocation achieved the highest validation ipTM after 100 updates, or 12,800 designs. After 2,000 target visits, all three allocations reached similar scores, but used different sampling budgets (Figure 9).