arXiv:2608.12192v1 [cs.AI] 12 Aug 2026

How to Spend Your Oracle Budget: Practical Guidance for Protein Structure Prediction Models

Aleksandra Kalisz [1 2] **Jack Simons** [1] **Krisztina Sinkovics** [1] **Noam Ghenassia** [1] **Shikha Surana** [1]
Henry Moss [3] **Paul Duckworth** [1]

Abstract

Foundation models for protein structure prediction remain unreliable on certain targets. External oracles can flag and correct these failures, but biological oracles are expensive, making oracle budget a critical constraint. Existing guidance methods, such as FK-steering, DPO, and Best K-of-N sampling, differ in how they spend this budget, yet no systematic comparison exists to guide method selection. To bridge this gap, we benchmark these methods alongside the recently proposed Optimisation Over Outputs (O3), which applies off-the-shelf optimisers within a generative model's latent subspace. We extend the usage of O3 to protein structure prediction models. Overall, our work provides the first practical reference for oracle budget-aware guidance. Our evaluation on two protein targets, calmodulin (1CLL) and E. coli aspartate transcarbamoylase (9EEH), reveals that no single method consistently dominates across all budgets and oracles. Specifically, O3 proves most effective at low oracle budgets, while FK-steering and DPO demonstrate improved performance as the budget increases. We distil these findings into actionable recommendations for practitioners operating under real-world oracle-budget constraints.

1. Introduction

Foundation models for structure prediction such as AlphaFold3 (Abramson et al., 2024), Chai-1 (Chai Discovery et al., 2024), and Boltz-2 (Passaro et al., 2025) can accurately predict structures from sequence information alone. Downstream applications, including drug discovery and antibody therapeutic design, rely on these predictions being correct. In practice these predictions are not always reliable: a model may collapse onto a single conformation when several are functionally relevant (Wayment-Steele et al., 2023), produce physically implausible geometries (Buttenschoen et al., 2024), or assemble large biomolecular complexes incorrectly (Yin & Pierce, 2023). A typical procedure to flag unreliable predictions is via the usage of external oracles, which assign a score to a given prediction, e.g. a molecular dynamics simulation of conformational stability or binding energy (Hollingsworth & Dror, 2018). A natural research direction is whether such oracles can be leveraged at training or inference time to return higher scoring structures, a process referred to as *guidance*. There are two dominant classes of guidance approaches, inference-time steering and fine-tuning; however their relative effectiveness is poorly understood, a gap in the literature we seek to fill.

Inference-time methods, such as Feynman-Kac steering (FK-steering) (Singhal et al., 2025), sample multiple interacting particles through the generative process and resample at intermediate steps using oracle scores. Fine-tuning methods, such as Direct Preference Optimisation (DPO) (Rafailov et al., 2023), instead update model parameters using oracle-scored samples. An even simpler approach to guidance with an oracle is Best K-of-N sampling, whereby we draw N samples and return the K with the highest oracle scores. All three approaches improve outputs given enough oracle evaluations, but biological oracles are usually expensive: a single molecular dynamics simulation can take days of GPU compute (Hollingsworth & Dror, 2018), and wet-lab assays are slower still.

As well as benchmarking the methods listed above, we also provide the first application of the newly proposed Optimisation Over Outputs (O3) framework of Willis et al. (2025) to a protein structure prediction model. O3 uses a small set of example model generations to build a low-dimensional subspace of the latent space over which standard optimisers can be directly applied. More details of the precise methodology are given in Willis et al. (2025).

We analyse the relative performances of O3, FK-steering, DPO, and Best K-of-N sampling applied to Boltz-2 (Pas-

[1]InstaDeep Ltd [2]University of Oxford [3]Lancaster University. Correspondence to: Paul Duckworth <p.duckworth@instadeep.com>.

Proceedings of the ICML 2026 Workshop on Structured Probabilistic Inference & Generative Modeling (SPIGM), Seoul, South Korea. 2026. Copyright 2026 by the author(s).

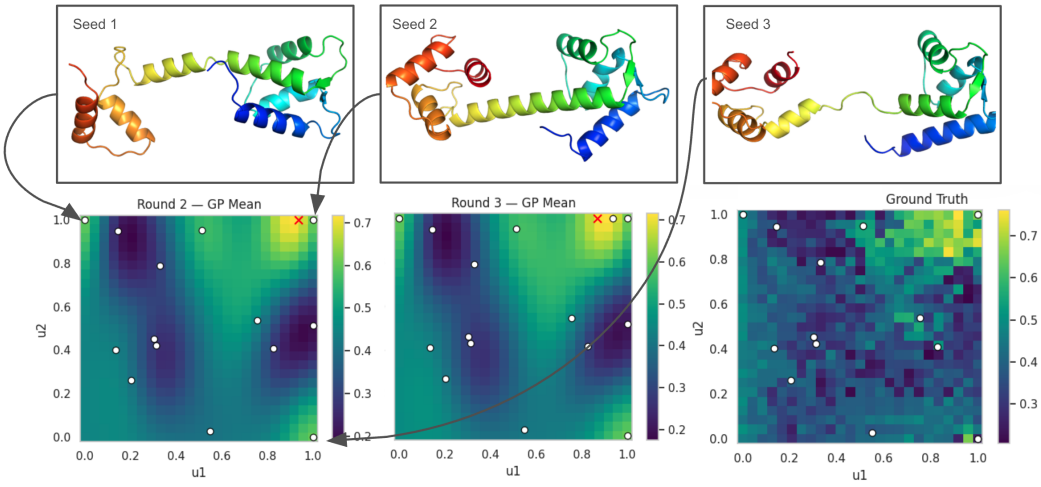

Figure 1. **Bayesian Optimisation via O3 in a Boltz-2.** $d = 3$ surrogate subspace created from three Calmodulin seed samples (top row), showing the 2D $\mathcal{U}$ space (bottom row), the GP training data (white dots) and mean predictive posterior over $\mathcal{U}$, the max acquisition point (red cross) for two acquisition rounds, and a 25x25 exhaustive ground-truth grid directly evaluating the oracle (bottom right).

saro et al., 2025) across a range of budgets, with the goal to improve the quality of the predicted protein structures. We find that our new application of O3 to Boltz-2 dominates at low-to-mid budgets that we evaluate on, while FK-steering and DPO perform better with larger budgets. This qualitative trade-off is demonstrated in our experimental study reported in Section 5. The primary goal of this paper is straightforward: *provide the first practical reference for which guidance strategies are effective under different oracle evaluation budgets.*

2. Related Work

Protein structure prediction. Recent foundation models for protein structure prediction, including AlphaFold3 (Abramson et al., 2024), Chai-1 (Chai Discovery et al., 2024), and Boltz-2 (Passaro et al., 2025), share a common architecture: an attention-based trunk, typically a Pairformer, that processes the input sequence into single and pair representations, which is then used as conditioning information to a diffusion block over atomic coordinates. Throughout our experiments we use the open-source model Boltz-2.

Latent space optimisation. Black-box optimisation describes optimisation of an objective function which is explicitly unknown *a priori* but can be queried through function evaluations. These evaluations can be expensive, noisy, and derivative-free. A popular approach to such problems is Bayesian Optimisation (BO) (see e.g. Garnett (2023)). However, standard BO approaches begin to fail in high-dimensional or highly structured spaces. To tackle this shortcoming, several Latent space Bayesian Optimisation (LSBO) approaches have been proposed which instead perform BO in the latent space associated with a generative model. Examples of latent space Bayesian optimisation include VAE-BO (Gómez-Bombarelli et al., 2018), weighted retraining (Tripp et al., 2020), LOL-BO (Maus et al., 2022), and COWBOYS (Moss et al., 2025). Generative models often have latent spaces that are too large to apply these methods directly, fortunately, recent work from Bodin et al. (2024) and Willis et al. (2025) proposes an alternative approach that extracts low-dimensional subspaces from generative models, over which standard optimisers can be applied.

Inference-time guidance. Inference-time guidance biases the sampling trajectory of a diffusion model toward a target distribution without modifying its weights. Classifier guidance (Dhariwal & Nichol, 2021) perturbs the sampling trajectory by adding the gradient of an externally-trained noise-aware classifier to the denoising step, while classifier-free guidance (Ho & Salimans, 2022) trains a single model on both conditional and unconditional inputs and combines their score estimates at inference time. We choose Feynman-Kac steering (Singhal et al., 2025) as a baseline for this work since it does not require propagating gradients from the oracle, and does not require an auxiliary network.

Reward-based fine-tuning. DDPO (Black et al., 2023) casts sampling as a multi-step Markov decision process and applies policy-gradient updates with the oracle as reward; DRaFT (Clark et al., 2023) backpropagates the oracle gradient through the sampling chain when the oracle is differentiable. DPO (Rafailov et al., 2023) sidesteps explicit reward modelling by fine-tuning on pairs of oracle-scored samples to increase the relative likelihood of the preferred one, and Diffusion-DPO (Wallace et al., 2023) adapts this objective to the diffusion likelihood. We choose DPO as our fine-tuning baseline because it requires only sample pairs and oracle scores.

Typically, biological oracles are non-differentiable, meaning that gradient information is not available, making many common guidance approaches unsuitable.

3. Guidance Methods

We denote our frozen generative model $g : \mathcal{Z} \to \mathcal{X}$, that maps random noise $z \sim p_0$ from within a latent space $z \in \mathcal{Z}$ to structures $x \sim p_1$ in data space $x \in \mathcal{X}$. An oracle $r : \mathcal{X} \to \mathbb{R}$ scores each generated output. In our experiments g is the protein structure prediction model Boltz-2 (Passaro et al., 2025) with $\mathcal{Z} = \mathcal{X} = \mathbb{R}^{3n}$ for an n-atom system and $p_0 = \mathcal{N}(0, I)$. The goal of guidance is to use this model to generate a batch of K candidate outputs whose scores under r should be as high as possible, under the constraint that we only make N oracle queries. We will now introduce three guidance methods, explaining the key modifications and design choices that were required to apply them to protein structure prediction models.

3.1. Bayesian Optimisation via O3

We now describe how the O3 framework of Willis et al. (2025) is used to guide the generative process of Boltz-2. The algorithm relies on two key components, each necessitating a number of oracle queries: (i) the construction of a suitable low-dimensional subspace $\mathcal{U}$ within which generation is constrained (the first M queries), and (ii) a resource-efficient optimization procedure to effectively explore this subspace (the remaining $N - M$ queries).

(i) Constructing $\mathcal{U}$. Given a total budget of N, we first sample $M < N$ initial structures from g and score them with r. Exploiting the determinism of g, we collect the d latents corresponding to the d highest scoring structures[1] as *seeds*, $Z = [z_1, \ldots, z_d]^\top$. Here d determines the dimension of the resulting optimisation space, $d-1$. We then follow the guidelines of O3 (Willis et al., 2025), and construct $\mathcal{U} \subset [0, 1]^{d-1}$ via a *surrogate chart* $\phi : \mathcal{U} \to \mathcal{Z}$ that decodes each $u \in \mathcal{U}$ to a corresponding latent z through two maps (see Algorithm 1). The first, a Knothe–Rosenblatt transform (Knothe, 1957; Rosenblatt, 1952) $\phi : \mathcal{U} \to \mathbb{S}_+^{d-1}$, maps u to a non-negative weight vector w on the simplex. The second, a LOL projection (Bodin et al., 2024) $\ell(w, Z) := w^\top Z$, projects these weight vectors back onto the support of our isotropic Gaussian prior p_0. This is effective because the original seeds themselves were drawn from p_0 and the weights are on the simplex. Once d seeds are chosen, the subspace is fixed, and never re-built. Optimisation occurs within $\mathcal{U}$, where $\mathcal{U}$ has dimensionality $d - 1$ and crucially is independent of the dimensionality of $\mathcal{Z}$, thus opening up opportunities for efficient optimization. In Appendix Fig. A.2 we select two seeds $d = 2$, and linearly interpolate between them using 5 intermediate weights, displaying their corresponding generations $g(\cdot) \sim p_1$.

[1] O3 requires a deterministic generative model, and so we converted the stochastic Boltz-2 generation process to its equivalent probability-flow formulation (Karras et al., 2022) and disable stochastic SE(3) augmentation.

Algorithm 1 Mapping from $\mathcal{U}$ to protein structures

$u = [u_1, \ldots, u_{d-1}]^\top$ // selected by optimiser
$w \leftarrow \phi(u)$ // Knothe–Rosenblatt transform
$z \leftarrow \ell(w, Z) := w^\top Z$ // LOL projection
$x \leftarrow g(z)$ // decoded structure, ready for scoring

(ii) Bayesian optimization over $\mathcal{U}$. Given a subspace $\mathcal{U}$, we use Bayesian optimisation (Garnett, 2023, BO) — the gold-standard for low-dimensional black-box optimisation — to efficiently utilise our remaining $N - M$ oracle budget. We begin by fitting a Gaussian process (GP) (Williams & Rasmussen, 2006) surrogate model on $d + 2$ initial points, where d of them are the seed latents projected onto $\mathcal{U}$ and the remaining 2 are sampled randomly from within $\mathcal{U}$. Each subsequent round uses Log Expected Improvement (Ament et al., 2023) to drive data acquisition, each time updating the GP model with the newly acquired point. In order to provide additional training data, we also project the d seeds into the $\mathcal{U}$ subspace using the surrogate chart ϕ, and include those as additional training data points (as demonstrated in Fig. 1). We use an RBF kernel, constant mean function, single task GP, and Monte Carlo acquisition function sampler from BOTorch (Balandat et al., 2020).

3.2. FK-Steering

Feynman-Kac steering (Singhal et al., 2025) is an inference-time method which uses the signal from an oracle to alter the sampling trajectory towards high-reward regions of the target distribution via Sequential Monte Carlo. The procedure generates M interacting stochastic processes called particles. Particles are resampled proportionally to weights w calculated for each particle i using the oracle reward r evaluated on the denoised predictions from x_t^i (for denoising timestep t):

$$w(x_t^i) = \frac{g(x_t^i \mid x_{t_{\text{prev}}}^i)}{g_{\text{proposal}}(x_t^i \mid x_{t_{\text{prev}}}^i)} \lambda \exp(r(x_t^i) - r(x_{t_{\text{prev}}}^i))$$

where λ controls the scale of the reward signal, and $g_{\text{proposal}}(x_t \mid x_{t_{\text{prev}}})$ is the proposal transition kernel. In our experiments, since most high-quality biological oracles are non-differentiable, we use the original BOLTZ-2 transition kernel $g(x_t^i \mid x_{t_{\text{prev}}}^i)$, demonstrating the gradient-free nature of FK-steering. The rewards are calculated only at the resampling steps and there are several ways to combine them across the reward trajectory (difference, max, average);

in this work we use the reward difference, which favours particles with increasing rewards.

To restrict to an oracle budget of N and return a batch of K distinct samples, we require $M \geq K$ particles, each of which has access to N/M oracle calls. Practically, this ratio determines the number of resampling steps in our algorithm. Boltz-2 employs a particular noising schedule, which stops injecting noise in the last quarter of the trajectory in order to avoid atomic jitter. Since FK-steering relies on the diversity of particles, we only resample at intermediate denoising steps where Boltz-2 adds positive noise. We call the oracle once at the start of the denoising process, once at the end, and spread the remainder of the $N/M - 2$ budget across the first $3/4$ of the denoising trajectory.

3.3. DPO

Unlike our other approaches, Direct Preference Optimisation (DPO) (Rafailov et al., 2023) fine-tunes and directly modifies the generative model parameters based on oracle-scored sample pairs, with the aim of increasing the likelihood of generating high-scoring structures. DPO does not require a differentiable oracle (no explicit reward modeling is required, rather just pairs of structures and their relative oracle rankings) making it amenable to common biological oracles. However, we find that a large oracle budget is required to support effective fine-tuning at foundation model-scale.

Diffusion-DPO training objective. DPO updates the model's weights θ in a direction that maximises the margin between the implicit rewards of preferred and rejected samples under a Kullback-Leibler regularised policy, as measured by the DPO loss:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\mathbb{E}_{(\mathbf{x}_w, \mathbf{x}_l)} \left[\log \sigma \left(\beta \left(\log \frac{p_\theta(\mathbf{x}_w)}{p_{\text{ref}}(\mathbf{x}_w)} - \log \frac{p_\theta(\mathbf{x}_l)}{p_{\text{ref}}(\mathbf{x}_l)} \right) \right) \right],$$

where $\beta > 0$ controls the strength of the KL penalty against moving the new model p_θ too far from the pre-trained reference model p_{ref} (initialized at the pre-trained Boltz-2 checkpoint), and σ represents the sigmoid function. Preference pairs always contain a preferred structure $\mathbf{x}_w$ and a rejected structure $\mathbf{x}_l$, each sampled from our N scored structures and lie above and below the median oracle score respectively. Obtaining log-likelihoods from diffusion models is prohibitively expensive (Song et al., 2020), we follow Wallace et al. (2023), and approximate the log-likelihood ratio in the DPO loss using differences in the individual training losses ($\mathcal{L}_{\text{DSM}}$) of the reference and fine-tuned model as $\log(p_\theta(\mathbf{x})/p_{\text{ref}}(\mathbf{x})) \approx \mathcal{L}^{\text{ref}}_{\text{DSM}}(\mathbf{x}) - \mathcal{L}^{\theta}_{\text{DSM}}(\mathbf{x})$.

Online and offline DPO. We evaluate two variants of DPO that differ in how oracle queries are distributed across training. The *offline* variant draws all N structures in a single batch, forms preference pairs once, and trains to convergence on the resulting fixed dataset. The *online* variant interleaves sampling and training, at each of the E epochs: we generate N/E structures from the current model, score the structures with the oracle, form preference pairs, update parameters for one epoch; until a total budget of $N = (N/E) \times E$ queries has been utilised. Critically, the reference model p_{ref} is reset to the current policy at the start of each epoch, thus softening the KL regularisation. However, this resetting weakens the influence of the pre-trained model — a tradeoff we investigate in Section 5.4.

4. Experimental Setup

As our main experiment, we benchmark the four methods by comparing the predicted structures for the calmodulin protein from its amino acid sequence (of length 144). As ground truth, we compare against the 1CLL structure (Chattopadhyaya et al., 1992) consisting of 1184 atom coordinates from the Protein Data Bank (PDB) (Berman et al., 2000). Additionally, we benchmark on E. coli aspartate transcarbamoylase (PDB: 9EEH), with results in Appendix B. For all experiments, the base model we use, g, is Boltz-2 (Passaro et al., 2025) and for 1CLL $\mathcal{Z} = \mathbb{R}^{3552}$, and 9EEH $\mathcal{Z} = \mathbb{R}^{21,696}$. All methods are permitted N oracle calls in total and return a batch of K candidate structures. Next, we describe the TM-score oracle used in the main experiment.

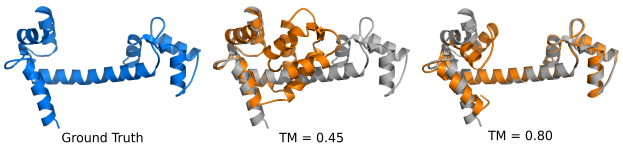

Figure 3. **TM-score on 1CLL.** Left: the 1CLL crystal structure (blue). Centre: a vanilla Boltz-2 sample (orange) overlaid on the ground truth (grey) at TM-score 0.45. Right: a guided sample at TM-score 0.80. Structure pairs with TM-score above 0.5 are considered to share the same fold.

Oracle. The reward, r, is the TM-score (Zhang & Skolnick, 2004) of a generated structure x against the ground-truth 1CLL backbone x_{GT}. To compare x with x_{GT}, we find the optimal rigid alignment, namely the rotation and translation of x that minimise the distances to the corresponding residues of x_{GT}. The TM-score is then

$$r(x) = \max \frac{1}{n} \sum_{i=1}^{n_{\text{ali}}} \frac{1}{1 + (d_i(x, x_{\text{GT}})/C)^2},$$

where n is the number of residues in the ground truth structure (144 for 1CLL); $n_{\text{ali}} \leq n$ is the number of residues from x_{GT} matched to a residue in x by the optimal alignment, with residues that have no nearby counterpart excluded from the sum; d_i is the distance between the i-th

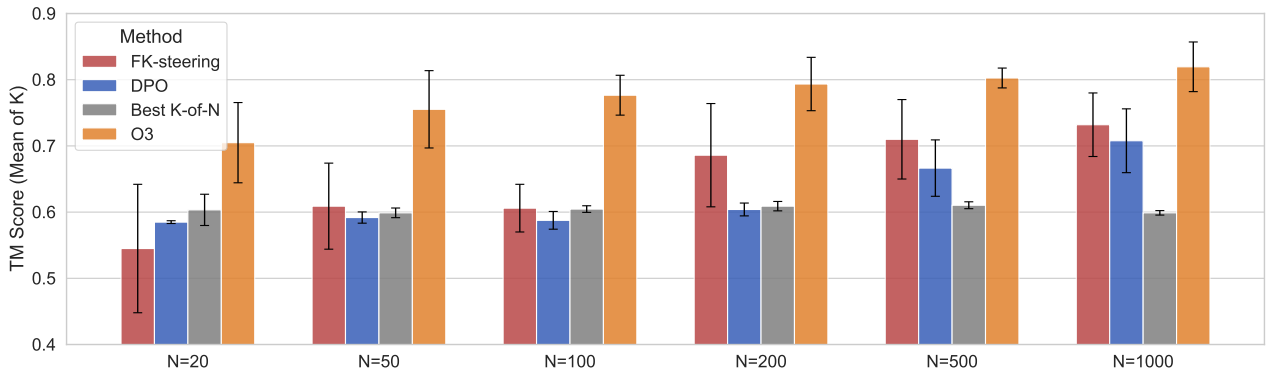

Figure 2. **Mean-of-K TM-score across oracle budgets.** Comparison of FK-steering, DPO (in the online setting), Best K-of-N, and O3 on 1CLL across the six (N, K) configurations of Section 4. Bars are the mean over 5 random seeds; error bars are ± 1 std. Higher is better. O3 method dominates at budgets evaluated, while FK-steering and DPO improve steadily with N.

aligned residue pair[2]; $C = 1.24\sqrt[3]{n-15} - 1.8$ is a length-dependent normalisation chosen so that random structure pairs yield approximately constant TM-score; and the maximum is over the rigid alignments described above. Scores lie in $(0, 1]$, with values above 0.5 indicating the same fold and below 0.2 indicating random pairs; Fig. 3 shows examples on 1CLL. We use TM-score because it is relatively fast to compute, thus enabling our extensive experimentation. However, we emphasise that our analysis exposes trade-offs between the studied methodologies in settings where indeed the oracle is expensive or non-differentiable, a setting common in biology.

For PDB: 9EEH experiments, we use the MolProbity oracle, described in Appendix B.1.

Budget settings. We evaluate each method at six (N, K) configurations spanning low to moderate oracle-call budgets: $(20, 2)$, $(50, 5)$, $(100, 10)$, $(200, 20)$, $(500, 50)$, and $(1000, 100)$. We are specifically interested in $K > 1$ settings throughout because downstream stages of biological design pipelines, such as wet-lab assay screening, often test batches of candidates rather than committing to a single proposal.

Evaluation metrics. At each (N, K) configuration, we report two performance metrics: the *max-of-K* score, the best single structure returned, and the *mean-of-K* score, the average score across the returned batch. The former measures single-shot design quality; the latter measures batched-screening quality. Each plot in Section 5 reports one of the two, with the complementary results presented in our appendices. All experiments are repeated over 5 random seeds unless stated otherwise.

[2]it is a distance between the C_α atoms of the two residues.

5. Results

Section 5.1 discusses the comparison of the four methods on PDB: 1CLL target, across six (N, K) configurations introduced in Section 4. Sections 5.2 to 5.4 elaborate on method-specific findings and ablations for O3, FK-steering, and DPO, respectively. PDB: 9EEH results are provided in Appendix B.

5.1. Model Comparison Across Budgets

Fig. 2 reports the mean-of-K TM-score for our four methods (and Appendix Section A.1 presents the max-of-K results). O3 outperforms all baselines at low-mid budgets ($N \leq 1000$) with the mean value plateauing at ~ 0.81. Best K-of-N is roughly flat at ~ 0.60 across all budgets. FK-steering and DPO improve with increased N, but neither are competitive with simple Best K-of-N at low budgets ($N \leq 100$), with FK-steering achieving only ~ 0.55 at $N = 20$. From $N = 200$ to $N = 1000$, FK-steering performs better, reaching ~ 0.73 at $N = 1000$; DPO improves with N but does not match FK-steering or O3 within our range of settings, reaching ~ 0.71 at $N = 1000$. O3 is the only method that meaningfully improves on the Best K-of-N baseline at low oracle budgets ($N \leq 100$), supporting the case for example-defined latent-subspace optimisation when oracle calls are scarce. Appendix B.2 reports a model comparison across budgets on a different protein (PDB:9EEH) and MolProbity oracle (Appendix B.1).

5.2. O3

Fig. 4 isolates the optimisation step in O3, comparing Bayesian optimisation in the subspace against uniform random sampling in the same subspace for $N = 100$. We create the subspace $\mathcal{U}$ by selecting to best d seeds from $\mathcal{M}$. Random sampling in $\mathcal{U}$ is shown to improve over the Best

K-of-N sampling baseline, demonstrating the value of the low-dimensional subspace. Furthermore, the superior performance of O3 seen in Fig. 2 comes from a combination of both the optimiser and the subspace construction.

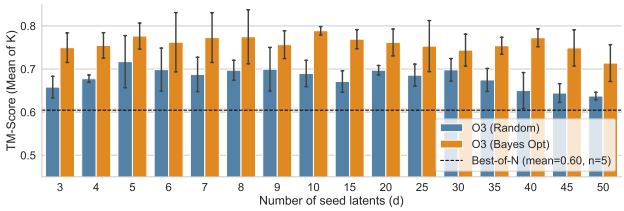

Figure 4. **Bayesian optimisation versus random sampling in the O3 subspace.** Mean-of-K TM-score at $N = 100$ for O3 with Bayesian optimisation and O3 with uniform random sampling in the same subspace, on 1CLL. The Best K-of-N baseline is invariant under our fixed K/N ratio. Bars are means over 3 random seeds; error bars are ± 1 std.

In the experiment summarised by Fig. 5 we sweep across subspace dimensions d for different available budgets. The key takeaway is that best-performing d varies with the budget — different oracle-call regimes favour different subspace dimensions. For example, when $N = 20$ the best-performing dimension is $d = 6$; however, for $N = 200$ dimension $d = 10$ yields better predictions. It is clear there is a trade-off between the richness of the O3 subspace, guaranteed by increasing the number of seed latents d, and the resulting difficulty of the optimisation task in that subspace. Additional O3 ablations and results are presented in Appendices A.2 and B.3.

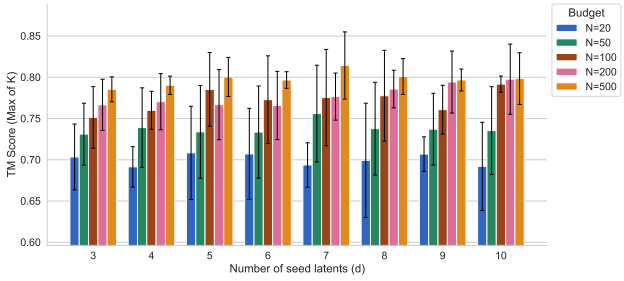

Figure 5. **Performance for different O3 subspace dimensions.** Max-of-K TM-score for O3 with Bayesian optimisation, across subspace dimensions d for different budgets, on 1CLL. The best-performing d varies with the budget. Bars are means over 3 random seeds; error bars are ± 1 std.

5.3. FK-steering

We ablate three of FK-steering's hyperparameters: the number of particles, K, the number of resampling steps N/K, and the reward-scaling coefficient λ. Fig. 6 shows that, with K and N fixed, increasing λ improves performance across all budgets, however, it likely hinders output diversity. A high $\lambda = 50$ amplifies the signal from the oracle, forcing the algorithm to greedily resample the best-performing particles across the trajectory, driving the mean-of-K and max-of-K (see Fig. A.7) closer. In Fig. A.8 we vary K per fixed budgets of N and find that increasing the number of resampling steps N/K at the cost of particle population K generally yields better performance at higher oracle budgets. Additional FK-Steering ablations are available in Appendix A.3.

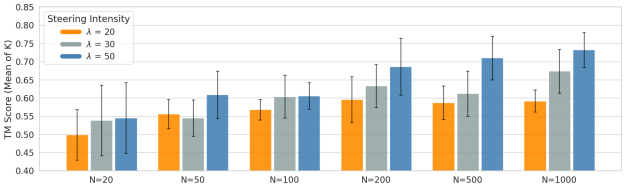

Figure 6. **Effect of FK-steering λ hyperparameter across different budgets** N. Mean-of-K TM-score achieved by FK-steering on 1CLL. Bars are means over 10 random seeds; error bars are ± 1 std. Higher λ increases the signal from the rewards and leads to better performance across budgets.

5.4. DPO

Fig. 7 compares online and offline DPO across oracle budgets. (Max-of-K counterpart of this result is reported in Appendix A.4). Online DPO improves steadily with N, reaching a mean TM-score of 0.708 and a max TM-score of 0.783 at $N = 1000$. In contrast, offline DPO shows little sensitivity to budget, plateauing at 0.55–0.57 across all budgets. The gap between the two variants widens with scale: negligible at $N = 100$, it becomes substantial at $N = 500$. These results suggest that the gains from DPO arise primarily from adaptive on-policy resampling, rather than from additional optimisation steps alone, consistent with prior observations that online preference optimisation methods can significantly outperform offline counterparts under a fixed budget (Tang et al., 2024). We also find that training hyperparameters such as the preference batch size and the number of samples per epoch can meaningfully affect performance at large N, though a controlled study isolating each factor is left for future work.

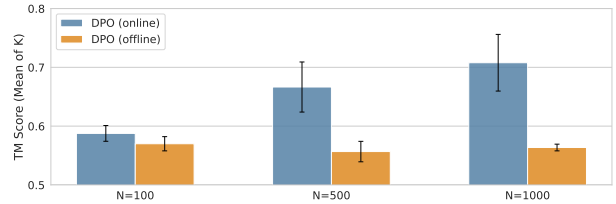

Figure 7. **Online vs. offline DPO across oracle budgets.** Mean-of-K TM-score achieved by online and offline DPO on 1CLL, for oracle budgets $N \in \{100, 500, 1000\}$. Bars are means over 5 random seeds; error bars show ± 1 std. Online DPO improves consistently with N, while offline DPO remains flat across budgets.

6. Conclusions

Notably, our work represents the first application of O3 to protein structure prediction. We demonstrate that this novel approach equips practitioners with a highly effective tool, yielding exceptional results particularly in budget-constrained tasks. Using Boltz-2 and two protein targets along with the TM-score and MolProbity oracles, we compare O3, FK-steering, DPO, and Best K-of-N sampling under oracle-budget constraints, thus providing a practical reference for budget-aware guidance in biological settings. Across the budgets we evaluated, O3 dominates the performance. At ≤ 100 queries, it is the only method that meaningfully improves on the Best K-of-N baseline, while FK-steering and DPO both steadily improve with N. As the only method that updates the model parameters, DPO is the natural candidate at substantially larger budgets than we evaluate here.

The conclusion from this work can be summarised in the following piece of practical advice: use O3 at constrained oracle budgets. FK-steering can be competitive at moderate budgets, and reach for DPO once budgets are large enough to fine-tune. An additional practical consideration is batch size and required batch diversity. DPO amortises its training budget into the model weights, making sampling large subsequent batches cheap, whereas inference-time and search methods usually discard oracle values afterwards. Future work will include additional protein targets paired with other, potentially expensive and non-differentiable, oracles.

Impact Statement

Our work is a methodological comparison of guidance methods for protein-structure foundation models, aimed at helping practitioners spend limited biological oracle budgets effectively, including for downstream applications such as drug discovery and therapeutic design. As with all general-purpose tools for protein design, the same techniques could in principle be applied to harmful targets; our experiments use only the publicly deposited calmodulin structure 1CLL and E. coli aspartate transcarbamoylase structure 9EEH and pose no specific dual-use risk in themselves.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., Bodenstein, S. W., Evans, D. A., Hung, C.-C., O'Neill, M., Reiman, D., Tunyasuvunakool, K., Wu, Z., Žemgulytė, A., Arvaniti, E., Beattie, C., Bertolli, O., Bridgland, A., Cherepanov, A., Congreve, M., Cowen-Rivers, A. I., Cowie, A., Figurnov, M., Fuchs, F. B., Gladman, H., Jain, R., Khan, Y. A., Low, C. M. R., Perlin, K., Potapenko, A., Savy, P., Singh, S., Stecula, A., Thillaisundaram, A., Tong, C., Yakneen, S., Zhong, E. D., Zielinski, M., Žídek, A., Bapst, V., Kohli, P., Jaderberg, M., Hassabis, D., and Jumper, J. M. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, May 2024. ISSN 1476-4687. URL http://dx.doi.org/10.1038/s41586-024-07487-w.

Ament, S., Daulton, S., Eriksson, D., Balandat, M., and Bakshy, E. Unexpected improvements to expected improvement for bayesian optimization. *Advances in neural information processing systems*, 36:20577–20612, 2023.

Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020. URL http://arxiv.org/abs/1910.06403.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The Protein Data Bank. *Nucleic Acids Research*, 28(1): 235–242, January 2000. ISSN 1362-4962. URL http://dx.doi.org/10.1093/nar/28.1.235.

Black, K., Janner, M., Du, Y., Kostrikov, I., and Levine, S. Training Diffusion Models with Reinforcement Learning, 2023. URL https://arxiv.org/abs/2305.13301.

Bodin, E., Stere, A., Margineantu, D. D., Ek, C. H., and Moss, H. Linear combinations of latents in generative

models: subspaces and beyond, 2024. URL https://arxiv.org/abs/2408.08558.

Buttenschoen, M., Morris, G. M., and Deane, C. M. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024. ISSN 2041-6539. URL http://dx.doi.org/10.1039/d3sc04185a.

Chai Discovery, Boitreaud, J., Dent, J., McPartlon, M., Meier, J., Reis, V., Rogozhnikov, A., and Wu, K. Chai-1: Decoding the molecular interactions of life. October 2024. URL http://dx.doi.org/10.1101/2024.10.10.615955.

Chattopadhyaya, R., Meador, W. E., Means, A. R., and Quiocho, F. A. Calmodulin structure refined at 1.7 å resolution. *Journal of Molecular Biology*, 228(4):1177–1192, December 1992. ISSN 0022-2836. URL http://dx.doi.org/10.1016/0022-2836(92)90324-D.

Chen, V. B., Arendall, W. B., Headd, J. J., Keedy, D. A., Immormino, R. M., Kapral, G. J., Murray, L. W., Richardson, J. S., and Richardson, D. C. MolProbity: all-atom structure validation for macromolecular crystallography. *Acta Crystallographica Section D: Biological Crystallography*, 66(1):12–21, 2010. doi: 10.1107/S0907444909042073.

Clark, K., Vicol, P., Swersky, K., and Fleet, D. J. Directly Fine-Tuning Diffusion Models on Differentiable Rewards, 2023. URL https://arxiv.org/abs/2309.17400.

Dhariwal, P. and Nichol, A. Diffusion Models Beat GANs on Image Synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Garnett, R. *Bayesian optimization*. Cambridge University Press, 2023.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, 4(2):268–276, January 2018. ISSN 2374-7951. URL http://dx.doi.org/10.1021/acscentsci.7b00572.

Ho, J. and Salimans, T. Classifier-Free Diffusion Guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Hollingsworth, S. A. and Dror, R. O. Molecular Dynamics Simulation for All. *Neuron*, 99(6):1129–1143, September 2018. ISSN 0896-6273. URL http://dx.doi.org/10.1016/j.neuron.2018.08.011.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the Design Space of Diffusion-Based Generative Models, 2022. URL https://arxiv.org/abs/2206.00364.

Knothe, H. Contributions to the theory of convex bodies. *Michigan Mathematical Journal*, 4(1), January 1957. ISSN 0026-2285. URL http://dx.doi.org/10.1307/mmj/1028990175.

Maus, N., Jones, H. T., Moore, J. S., Kusner, M. J., Bradshaw, J., and Gardner, J. R. Local Latent Space Bayesian Optimization over Structured Inputs, 2022. URL https://arxiv.org/abs/2201.11872.

Miller, R. C., Patterson, M. G., Bhatt, N., Pei, X., and Ando, N. Cooperativity in e. coli aspartate transcarbamoylase is tuned by allosteric breathing. *Nature Communications*, 17(1):4285, mar 2026. ISSN 2041-1723. doi: 10.1038/s41467-026-70909-y. URL https://doi.org/10.1038/s41467-026-70909-y.

Moss, H., Ober, S. W., and Diethe, T. Return of the latent space cowboys: Re-thinking the use of vaes for bayesian optimisation of structured spaces. In *International Conference on Machine Learning*, pp. 44956–44970. PMLR, 2025.

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., Kwabi-Addo, D., Beaini, D., Jaakkola, T., and Barzilay, R. Boltz-2: Towards Accurate and Efficient Binding Affinity Prediction. June 2025. URL http://dx.doi.org/10.1101/2025.06.14.659707.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, 2023. URL https://arxiv.org/abs/2305.18290.

Rosenblatt, M. Remarks on a Multivariate Transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, September 1952. ISSN 0003-4851. URL http://dx.doi.org/10.1214/aoms/1177729394.

Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKeown, K., and Ranganath, R. A General Framework for Inference-time Scaling and Steering of Diffusion Models, 2025. URL https://arxiv.org/abs/2501.06848.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Tang, Y., Guo, D. Z., Zheng, Z., Calandriello, D., Cao, Y., Tarassov, E., Munos, R., Ávila Pires, B., Valko, M., Cheng, Y., and Dabney, W. Understanding the performance gap between online and offline alignment algorithms. *arXiv preprint arXiv:2405.08448*, 2024.

Tripp, A., Daxberger, E., and Hernández-Lobato, J. M. Sample-Efficient Optimization in the Latent Space of Deep Generative Models via Weighted Retraining, 2020. URL https://arxiv.org/abs/2006.09191.

Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., and Naik, N. Diffusion Model Alignment Using Direct Preference Optimization, 2023. URL https://arxiv.org/abs/2311.12908.

Wayment-Steele, H. K., Ojoawo, A., Otten, R., Apitz, J. M., Pitsawong, W., Hömberger, M., Ovchinnikov, S., Colwell, L., and Kern, D. Predicting multiple conformations via sequence clustering and AlphaFold2. *Nature*, 625(7996):832–839, November 2023. ISSN 1476-4687. URL http://dx.doi.org/10.1038/s41586-023-06832-9.

Williams, C. J., Headd, J. J., Moriarty, N. W., Prisant, M. G., Videau, L. L., Deis, L. N., Verma, V., Keedy, D. A., Hintze, B. J., Chen, V. B., Jain, S., Lewis, S. M., Arendall, W. B., Snoeyink, J., Adams, P. D., Lovell, S. C., Richardson, J. S., and Richardson, D. C. MolProbity: More and better reference data for improved all-atom structure validation. *Protein Science*, 27(1):293–315, 2018. doi: 10.1002/pro.3330.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Willis, S., Stere, A. I., Margineantu, D. D., Oldroyd, H. T., Fozard, J. A., Ek, C. H., Moss, H., and Bodin, E. Defining latent spaces by example: optimisation over the outputs of generative models, 2025. URL https://arxiv.org/abs/2509.23800.

Yin, R. and Pierce, B. G. Evaluation of AlphaFold antibody–antigen modeling with implications for improving predictive accuracy. *Protein Science*, 33(1), December 2023. ISSN 1469-896X. URL http://dx.doi.org/10.1002/pro.4865.

Zhang, Y. and Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, October 2004. ISSN 1097-0134. URL http://dx.doi.org/10.1002/prot.20264.

A. Additional Results on PDB: 1CLL

A.1. Complementary Model Comparison Across Budgets

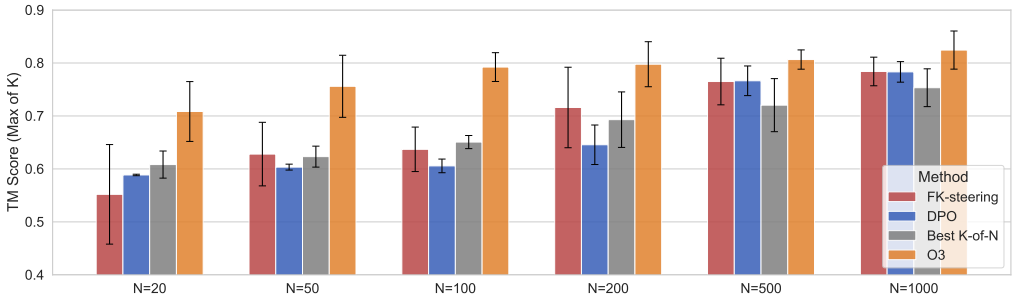

Figure A.1. **Max-of-K TM-score across oracle budgets.** Comparison of FK-steering, DPO, Best K-of-N, and O3 on 1CLL across the six (N, K) configurations of Section 4, reporting the maximum TM-score across the returned batch of K candidate structures. Bars are the mean over 5 random seeds; error bars are ± 1 std. Higher is better.

A.2. Additional O3 Details and Results on PDB: 1CLL

O3 Experimental Details

Table A.1. Experimental configurations for O3 protein structure optimisation (Boltz-2 generator, TM-score oracle on 1CLL). N: total Oracle budget; K: top-K batch selection; M: initial samples filtered to create subspace $\mathcal{U}$; d: number of seed latents used to create the $d-1$ dimensional $\mathcal{U}$; $d+2$: initial points used to fit the Gaussian process in $\mathcal{U}$; n_{rounds}: number of BO acquisition rounds is set to $N - M - 2$ for filtered seed-latents (and $N - (d+2)$ for random seed latents); Wall-clock time of experiments run using a H100 GPU with 80Gb RAM.

Task	N	K	M	d	n_{rounds}	wall-clock time (mins)
`n20_k2`	20	6	10	5	8	4.5 ($\pm$2.7)
`n50_k5`	50	7	25	7	23	6.1 ($\pm$0.4)
`n100_k10`	100	10	50	5	48	10.6 ($\pm$ 1.4)
`n200_k20`	200	10	100	10	98	20.6 ($\pm$8.6)
`n500_k50`	500	7	250	30	248	48.4588 ($\pm$2.2)
`n1000_k100`	1000	100	400	50	598	153.5 ($\pm$1.6)

LOL Interpolation

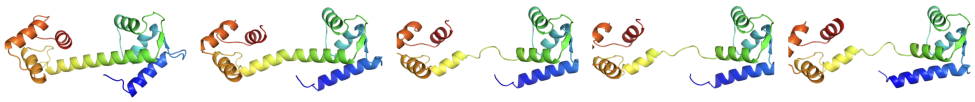

Figure A.2. **LOL interpolation trajectory** in a Boltz-2 $d = 2$ surrogate subspace of two Calmodulin examples, using 5 intermediate values of $(w, 1-w)$ for $w \in (0, 1)$. Structures vary smoothly along the interpolation while remaining within the support of Boltz-2. For more details see (Bodin et al., 2024).

COMPLEMENTARY O3 METRICS

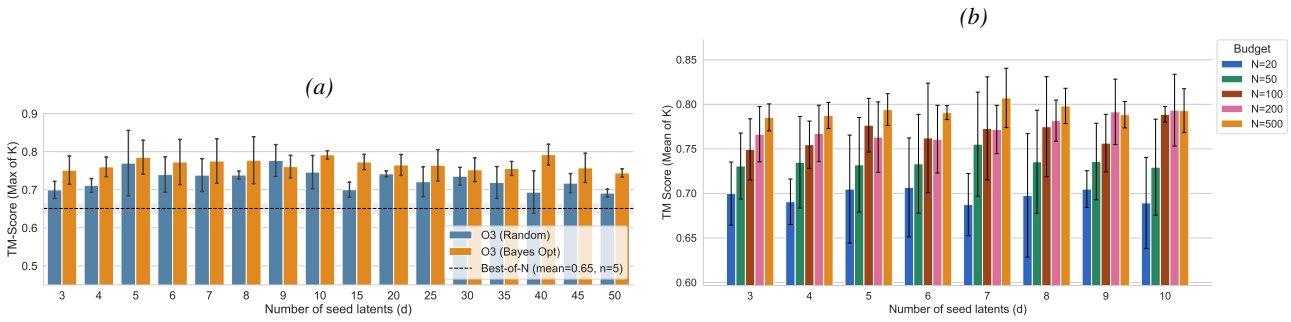

Figure A.3. **Complementary O3 metrics.** (a) Max-of-K counterpart to Fig. 4: BO versus random sampling at $N = 100$ and $K = 10$. (b) Mean-of-K counterpart to Fig. 5: subspace dimension sweep across budgets.

O3 ABLATE BUDGET ALLOCATION

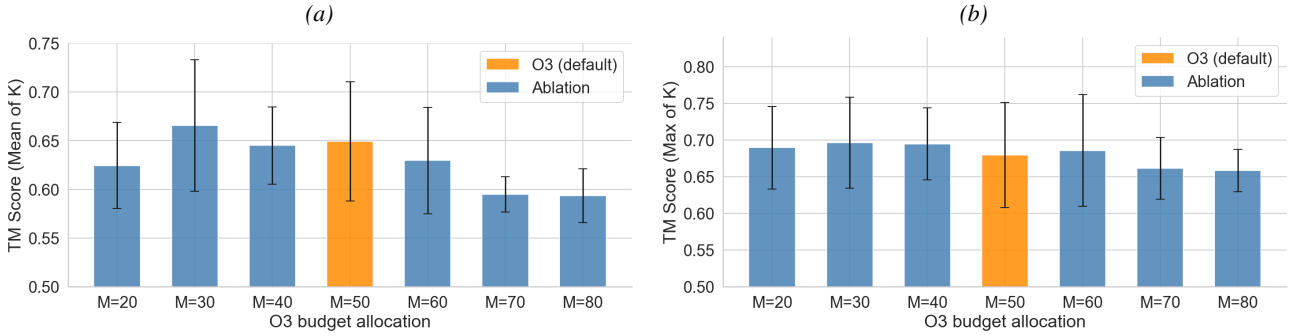

Figure A.4. **O3 Ablation: Budget allocation M.** (a) Mean-of-K TM Score reported for a fixed budget $N = 100$, varying the initial number of samples used to create the $\mathcal{U}$ subspace as per Section 3.1, resulting in varying remaining BO budget, $N - M$ acquisition rounds. (b) Max-of-K counterpart. In all cases we keep the subspace-dimension constant $d = 7$, $K = 10$, and draw $j = 10$ uniform random u samples as initial GP training data. Bars are the mean over 3 random seeds; error bars are ± 1 std. Higher is better. Experiment uses a previous set of hyperparameters.

O3 EFFECT OF GP COVARIANCE FUNCTION

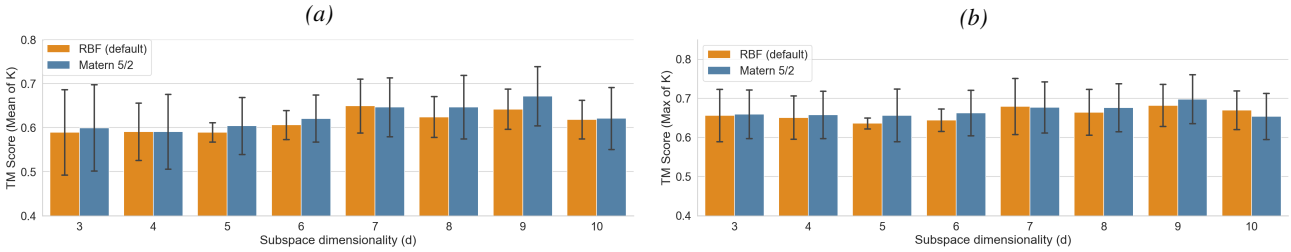

Figure A.5. **O3 Ablation: GP covariance function.** (a) Mean-of-K TM Score across different subspace dimensionality for an RBF kernel vs Matern 5/2 kernel as Gaussian process covariance function. (b) Max-of-K counterpart to (a). In all cases we keep the budget $N = 100$, batch size $K = 10$, with $j = 10$ uniform random u samples, as per Section 3. Bars are the mean over 5 random seeds; error bars are ± 1 std. Higher is better. Experiment uses a previous set of hyperparameters.

O3 Effects of Seed selection strategy

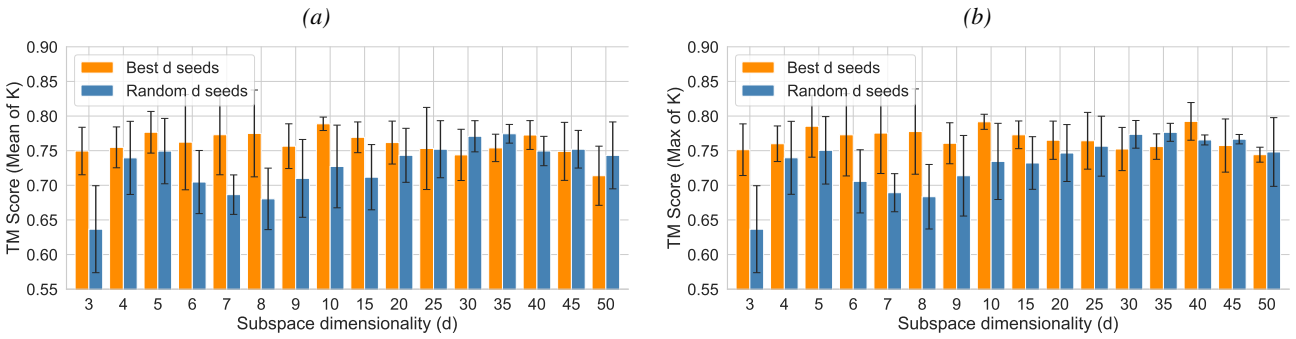

Figure A.6. **O3 Ablation: Choice of seed selection strategy.** (a) Mean-of-K TM Score on 1CLL for different approaches to construct the $\mathcal{U}$ subspace: either the "best d seeds" or "random d seeds" are used, across a sweep of subspace dimensionalities. Using random seeds means M = 0, so number of BO rounds is $N - (d + 2)$, whereas best d seeds has fewer $N - M - 2$. (b) Max-of-K counterpart to (a). In all cases we keep the budget $N = 100$ and batch size $K = 10$. Bars are the mean over 3 random seeds; error bars are ± 1 std. Higher is better.

A.3. Additional FK-Steering Results on PDB: 1CLL

Complementary FK-Steering Metrics

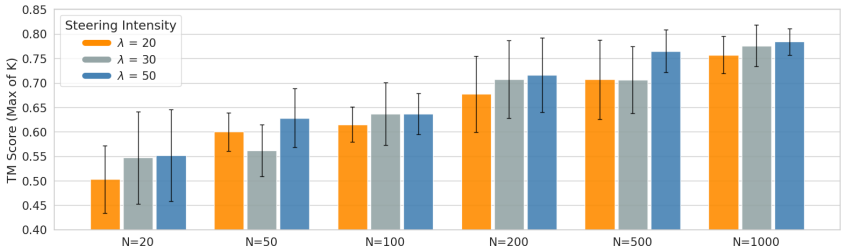

Figure A.7. **Effect of FK-steering λ hyperparameter across different budgets** N. Max-of-K TM-score achieved by FK-steering on 1CLL. Higher λ increases the signal from the rewards and leads to better performance across budgets. Bars are means over 10 random seeds; error bars are ± 1 std.

FK-Steering K vs N/K tradeoff

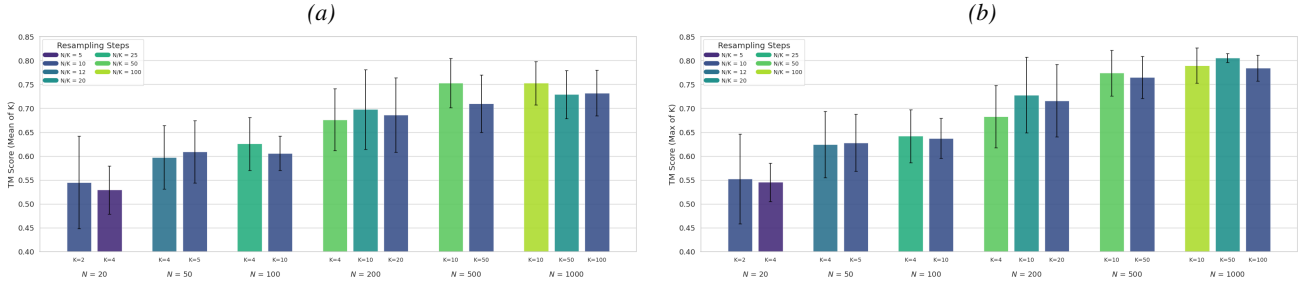

Figure A.8. **FK-steering K vs N/K tradeoff for a set budget.** (a) Mean-of-K TM Score for different approaches to allocate K particles and N/K resampling steps per set budget of N oracle calls. (b) Max-of-K counterpart to (a). For a given budget N, increasing the number of resampling steps N/K at the cost of particle population K generally improves performance in the higher budget groups. In all cases $\lambda = 50$. Bars are the mean over 10 random seeds; error bars are ± 1 std. Higher is better.

FK-STEERING REWARD TRAJECTORIES

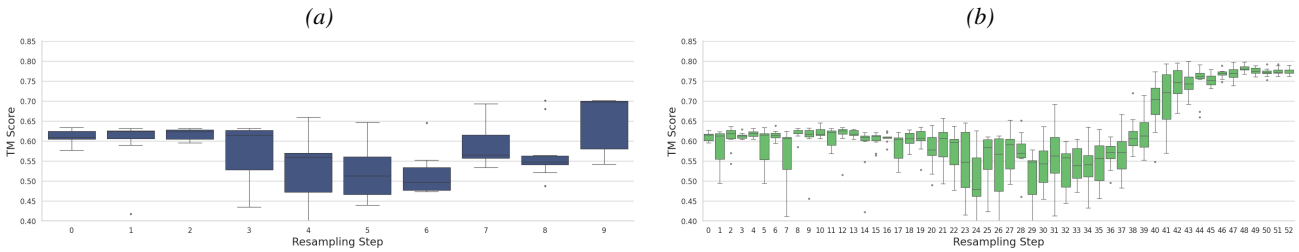

Figure A.9. **FK-steering reward trajectory for $K = 10$ particles with different number of resampling steps N/K.** (a) $N/K = 10$ resampling steps with a budget of $N = 100$ oracle calls. (b) $N/K = 50$ resampling steps with a budget of $N = 500$ oracle calls. In both cases $\lambda = 50$. Colors preserve N/K colorscheme from Fig. A.8. Resampling more frequently at higher N/K propagates the reward signal better and reduces reward variance in the final resampling steps.

A.4. Additional DPO Results on PDB: 1CLL

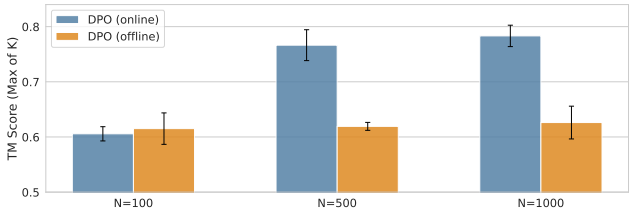

Figure A.10. **Online vs. offline DPO across oracle budgets.** Max-of-K TM-score achieved by online and offline DPO on 1CLL, for oracle budgets $N \in \{100, 500, 1000\}$. Bar are means over 5 random seeds; error bars show ± 1 std. Online DPO improves consistently with N, while offline DPO remains flat across budgets.

A.5. Compute resources used in PDB: 1CLL Experiments

Table A.2. **Compute resources used by the Boltz-2 guidance PDB: 1CLL experiment reported in this paper.** 'Evals/run' counts oracle (TM-score) evaluations, each of which corresponds to one full generation through Boltz-2. The FK-steering rows aggregate over the 3 values of λ swept per budget.

Experiment	Model / pipeline	Hardware	Evals/run	# runs	Per-run wall-clock	Total wall-clock
O3 (Willis et al., 2025)	BOLTZ-2 (PF-ODE)	H100	6 budgets up to N=1000	5 per budget	$\sim$ 0.1 h	$\sim$ 3 h
			N=20	5	2.7 m	13 m
			N=50	5	2.8 m	14 m
			N=100	5	3.1 m	16 m
			N=200	5	2.9 m	15 m
			N=500	5	3.4 m	17 m
			N=1000	5	18 m	1 h 31 m
Best K-of-N	BOLTZ-2	H100	6 budgets up to N=1000	5 per budget	$\sim$ 0.1 h	$\sim$ 3 h
			N=20	5	2.7 m	13 m
			N=50	5	2.8 m	14 m
			N=100	5	3.1 m	16 m
			N=200	5	2.9 m	15 m
			N=500	5	3.4 m	17 m
			N=1000	5	18 m	1 h 31 m
FK-steering (Singhal et al., 2025)	BOLTZ-2	H100	6 budgets, 3 λ values each			
			N=20, 3 λ vals	10	3.3 m	1 h 39 m
			N=50, 3 λ vals	10	3.8 m	1 h 54 m
			N=100, 3 λ vals	10	4.1 m	2 h 3 m
			N=200, 3 λ vals	10	5 m	2 h 30 m
			N=500, 3 λ vals	10	6.9 m	3 h 27 m
			N=1000, 3 λ vals	10	11 m	5 h 30 m
Online DPO (Rafailov et al., 2023; Wallace et al., 2023)	BOLTZ-2 (fine-tuned)	H100	6 budgets up to N=1000			
			N=20	4	4 m	16 m
			N=50	4	6 m	24 m
			N=100	5	11 m	55 m
			N=200	3	17 m	51 m
			N=500	5	44 m	3 h 40 m
			N=1000	4	1 h 31 m	6 h 04 m
Offline DPO (Rafailov et al., 2023; Wallace et al., 2023)	BOLTZ-2 (fine-tuned)	H100	6 budgets up to N=1000			
			N=20	4	4 m	16 m
			N=50	5	7 m	35 m
			N=100	4	18 m	1 h 12 m
			N=200	5	56 m	4 h 40 m
			N=500	4	5 h 40 m	22 h 40 m
			N=1000	3	21 h 57 m	65 h 51 m

B. Experimental Setup and Results on PDB: 9EEH

B.1. MolProbity Oracle

TM-score (Section 4) requires a ground-truth structure, which is usually unavailable in the design settings this paper targets. We therefore also evaluate against a reference-free oracle that scores the physical plausibility of a generated structure, built on MolProbity (Chen et al., 2010; Williams et al., 2018). The oracle runs six checks. Backbone dihedral quality comes from a Ramachandran analysis and Cβ deviation count via `mmtbx`. Steric quality is the clashscore, the number of serious all-atom overlaps (`probe` "bad overlap" contacts with a gap below -0.4 Å) per 1000 atoms, computed after adding and optimising all hydrogens with `reduce`. Covalent geometry is scored as RMS Z-scores of backbone bond lengths and bond angles against the Engh-Huber reference values, and peptide planarity contributes counts of cis-nonPro and twisted ω angles. These terms are combined into a single log-weighted penalty following the MolProbity score formulation, which we flip and normalise to a validity score in $[0, 1]$, with higher values indicating fewer physical violations.

B.2. PDB: 9EEH Results

We repeat the experiment of Section 5.1 on E. coli aspartate transcarbamoylase (PDB: 9EEH) (Miller et al., 2026), scored with the reference-free MolProbity oracle described above. There are three key differences between this setup and the one with the TM-score oracle on the calmodulin protein (PDB: 1CLL). First, 9EEH was deposited to the PDB after the training cutoff of Boltz-2, thus it is likely an out-of-distribution task for the base model, whereas 1CLL is not. Second, 9EEH is a 7,232-atom complex, which is much larger than 1CLL with 1,184 atoms. Third, the MolProbity score measures physical plausibility rather than similarity to a reference structure, and its effective range is a lot narrower, with most structures typically scoring between 0.5 and 0.7 under our MolProbity implementation.

Fig. B.1 reports the mean-of-K MolProbity score. O3 scores highest at all but the smallest budget, where it matches Best K-of-N. Best K-of-N has a relatively constant performance across budgets, as expected under fixed K/N ratio. DPO improves with N, although it performs significantly worse than Best K-of-N at low budgets. FK-steering performs poorly throughout and does not improve with budget, in contrast to 1CLL where it does better than Best K-of-N for budgets above $N = 100$. In Fig. B.2 under max-of-K, Best K-of-N is strongest at large budgets and both it and FK-steering improve with N, while O3 is unchanged.

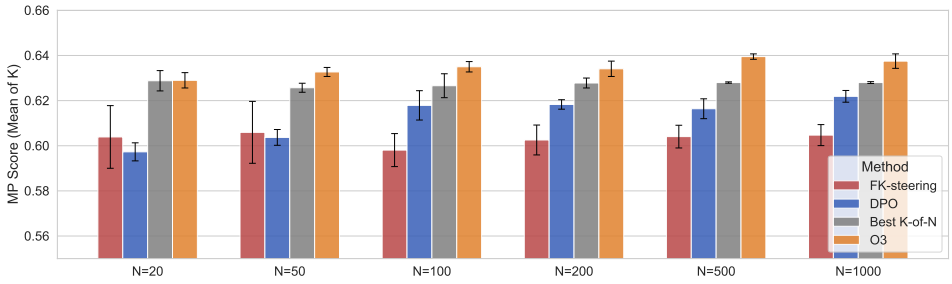

Figure B.1. **Mean-of-K MolProbity score across oracle budgets.** Comparison of FK-steering, DPO (in the online setting), Best K-of-N, and O3 on 9EEH across the six (N, K) configurations of Section 4. Bars are the mean over 3 random seeds; error bars are ± 1 std. Higher is better.

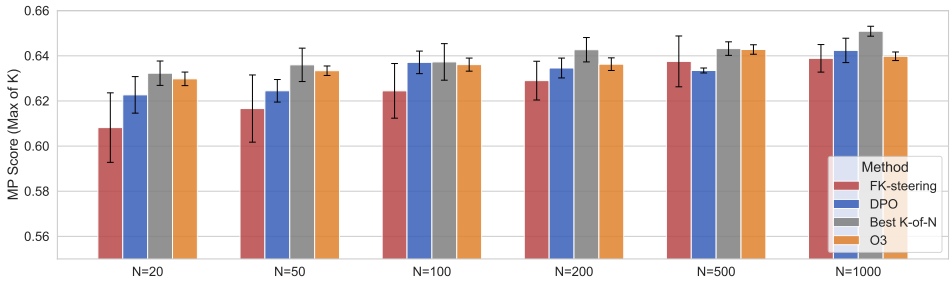

Figure B.2. **Max-of-K MolProbity score across oracle budgets.** Comparison of FK-steering, DPO, Best K-of-N, and O3 on 9EEH across the six (N, K) configurations of Section 4, reporting the maximum MolProbity score across the returned batch of K candidate structures. Bars are the mean over 3 random seeds; error bars are ± 1 std. Higher is better.

Under max-of-K, O3 performs worse than Best K-of-N, and worse than all methods at the highest budget. We hypothesise that this is because it is the only method that requires sampling Boltz-2 with the probability-flow ODE while the other three methods use the stochastic Boltz-2 sampler which improves the output diversity. Under mean-of-K this yields worse performance overall because the lower-scoring samples drag down the average, however it improves the max-of-K scores, where only the best sample matters and increasing N gives better scores even without any guidance signal (i.e. Best K-of-N baseline).

FK-steering is the only method that scores intermediate denoised predictions, and it is the weakest method under the MolProbity oracle. TM-score varies smoothly with structural accuracy: the same global fold already scores above 0.5, and refinement moves the score toward 1, so intermediate predictions receive informative scores. MolProbity instead measures local geometry, in which residual noise dominates until the final denoising steps. Since the quality of the denoised predictions is much worse at the intermediate steps than at the final ones, this leaves noise-sensitive intermediate MolProbity scores nearly uninformative. Moreover, since FK-steering relies on a positive noise scale for particle diversity, in Boltz-2 resampling is concentrated in the first three quarters of the denoising trajectory, which exacerbates the issues with signal quality from the MolProbity oracle at intermediate steps.

Finally, the differences between methods on this target are small, and the error bars over three seeds overlap at several budgets. Three of the four methods behave much as they do on 1CLL — O3 scores highest, DPO improves with N, and Best K-of-N is relatively constant — with FK-steering the exception, for the reason above. We therefore read these orderings as consistent trends across budgets rather than as individually meaningful differences, and note that guidance is less effective on this target than on 1CLL with TM-score.

B.3. Additional O3 Details and Results on PDB: 9EEH

O3 Experimental Details

Table B.1. Experimental configurations for O3 protein structure optimisation (Boltz-2 generator, MolProbity oracle on 9EEH). N: total Oracle budget; K: top-K batch selection; M: initial samples filtered to create subspace $\mathcal{U}$; d: number of seed latents used to create the $d-1$ dimensional $\mathcal{U}$; $d+2$: initial points used to fit the Gaussian process in $\mathcal{U}$; n_{rounds}: number of BO acquisition rounds is set to $N-M-2$ for filtered seed-latents (and $N-(d+2)$ for random seed latents); Wall-clock time of experiments run using a H100 GPU with 80Gb RAM.

Task	N	K	M	d	n_{rounds}	wall-clock time (mins)
`n20_k2`	20	2	10	5	8	9.1 ($\pm$0.1)
`n50_k5`	50	5	25	5	23	19.2 ($\pm$0.3)
`n100_k10`	100	10	50	20	48	40.0 ($\pm$ 0.7)
`n200_k20`	200	20	100	20	98	75.9 ($\pm$2.0)
`n500_k50`	500	50	250	30	248	224.2 ($\pm$9.0)
`n1000_k100`	1000	100	400	50	598	368.0 ($\pm$7.4)

O3 Effects of Seed selection strategy

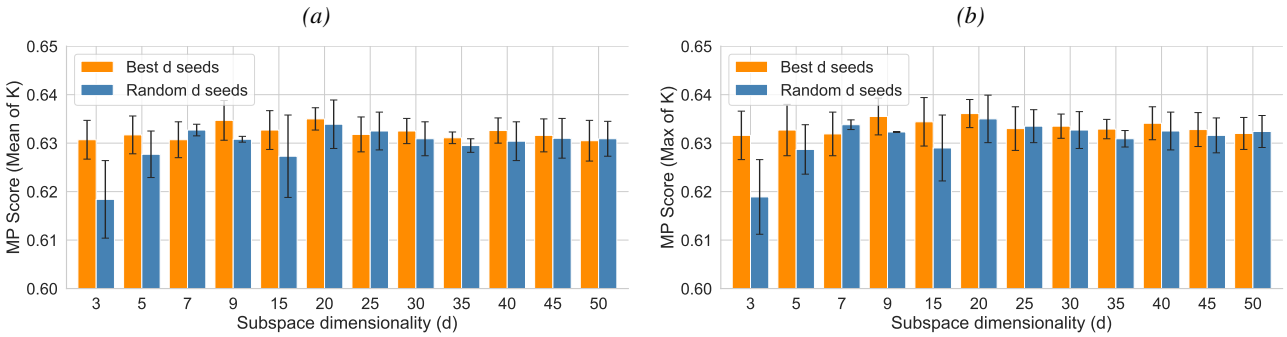

Figure B.3. **O3 Ablation: Choice of seed selection strategy.** (a) Mean-of-K MolProbity Score on 9EEH for different approaches to construct the $\mathcal{U}$ subspace: either the "best d seeds" or "random d seeds" are used, across a sweep of subspace dimensionalities. Using random seeds means M = 0, so the number of BO rounds is $N-(d+2)$, whereas best d seeds has fewer $N-M-2$. (b) Max-of-K counterpart to (a). In all cases, we keep the budget $N=100$ and batch size $K=10$. Bars are the mean over 3 random seeds; error bars are ± 1 std. Higher is better.