PROSPERO: Active Learning for Robust Protein Design Beyond Wild-Type Neighborhoods

Michal Kmicikiewicz[1,2] **Vincent Fortuin**[2,3,4] **Ewa Szczurek**[1,5]

[1]Institute of AI for Health, Helmholtz Munich
[2]School of Computation, Information and Technology, Technical University of Munich
[3]Helmholtz AI, [4]Munich Center for Machine Learning
[5]Faculty of Mathematics, Informatics and Mechanics, University of Warsaw
{michal.kmicikiewicz, vincent.fortuin, ewa.szczurek}@helmholtz-munich.de

Abstract

Designing protein sequences of both high fitness and novelty is a challenging task in data-efficient protein engineering. Exploration beyond wild-type neighborhoods often leads to biologically implausible sequences or relies on surrogate models that lose fidelity in novel regions. Here, we propose PROSPERO, an active learning framework in which a frozen pre-trained generative model is guided by a surrogate updated from oracle feedback. By integrating fitness-relevant residue selection with biologically-constrained Sequential Monte Carlo sampling, our approach enables exploration beyond wild-type neighborhoods while preserving biological plausibility. We show that our framework remains effective even when the surrogate is misspecified. PROSPERO consistently outperforms or matches existing methods across diverse protein engineering tasks, retrieving sequences of both high fitness and novelty.

1 Introduction

Proteins are essential macromolecules that play a central role in virtually all biological processes. The ability to design novel protein sequences with desired functional properties is crucial for a wide range of applications, including drug design, industrial biotechnology, and beyond [1–3]. Despite this promise, optimization of protein sequences remains a grand challenge in computational biology. The protein *fitness landscape* [4], mapping between the space of sequences and *fitness*, their corresponding functional levels, is typically rugged, sparse, and highly non-convex [5, 6]. Moreover, upon proposing a candidate sequence from the combinatorially large search space, evaluation requires querying an expensive black-box objective function. To alleviate this, machine learning models are often used as inexpensive surrogate models that approximate the costly black-box oracle [7–9]. To further facilitate navigation of the landscape, optimization commonly begins from a *wild-type* sequence, preserved through natural evolution and, as such, exhibiting reasonable fitness [6, 10].

Numerous strategies have emerged to traverse protein fitness landscapes. Ren et al. [6] introduced PEX, an evolutionary algorithm that exploits the wild-type neighborhood. While highly effective and favoring biologically plausible sequences thanks to its local focus, this approach limits broader exploration of the fitness landscape, potentially missing advantageous sequences that are inaccessible through local mutations. To overcome this, reinforcement learning (RL) methods [8, 10] and Generative Flow Networks (GFNs) [9] have been employed to target novel regions of the search space. However, these global exploration strategies often encounter issues with surrogate model

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

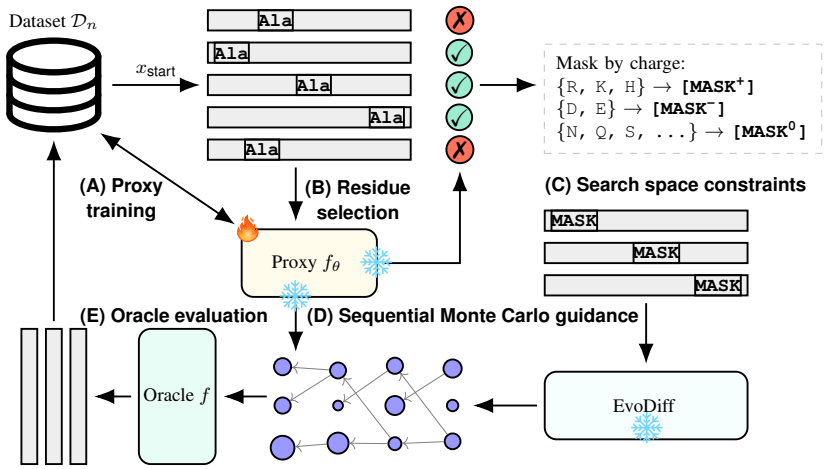

Figure 1: Overview of PROSPERO. Each active learning iteration begins with training a surrogate model on the current dataset (A). The surrogate is then used to identify fitness-relevant residues within the top sequence (B), which are subsequently masked, yielding partially masked sequences (C). EvoDiff, guided by the surrogate, completes these sequences to generate new candidates (D), which are evaluated by the oracle and added to the dataset (E).

misspecification when evaluating sequences substantially different from the surrogate's original training distribution [11]. To address this and balance exploration with robustness, GFN-AL-δCS [12] learns an unmasking policy that reconstructs partially masked sequences. Yet, since masking is applied at random, this approach may modify conserved residues, potentially degrading structural and functional integrity and yielding biologically implausible proteins. Pre-trained generative models offer a compelling alternative, inherently encoding rich biological priors that greatly reduce the risk of generating implausible sequences [13]. However, effectively incorporating such models in iterative optimization workflows presents a challenge, as each iteration would require impractical task-specific fine-tuning on limited oracle-annotated data or low-fidelity surrogate-annotated data [14, 15]. The shortcomings of existing approaches point to the need for a *protein design framework capable of generating high-fitness sequences beyond the wild-type neighborhood, while addressing the surrogate misspecification and loss of biological plausibility* that often arise from exploring such novel regions.

To meet these challenges, we introduce PROSPERO. Our main contributions are:
(i) A robust exploration framework, to our knowledge the first to formulate iterative design of protein sequences as inference-time guidance of a pre-trained generative model by a surrogate updated in an active learning loop. This enables straightforward incorporation of biological priors encoded by the generative model, helping preserve biological plausibility even when surrogate-guided exploration extends beyond wild-type neighborhoods.
(ii) A targeted masking strategy, which focuses edits on fitness-relevant residues while preserving structurally and functionally important sites. In contrast, prior approaches risk disrupting essential residues through random or uninformed masking.
(iii) Biologically-constrained Sequential Monte Carlo (SMC) sampling, offering a novel strategy to incorporate explicit biological priors into inference-time guidance in discrete sequence space. Restricting proposals to amino acids with properties similar to their wild-type counterparts increases the likelihood of retrieving high-fitness sequences in novel regions of the search space, where the surrogate may be misspecified.

We conduct extensive experiments across diverse protein fitness landscapes and demonstrate that PROSPERO consistently approaches the Pareto frontier between candidate sequence fitness and novelty, while preserving biological plausibility. We perform ablation studies under varying degrees of proxy misspecification to assess the contribution of individual components to the overall robustness of our method. Code is available at https://github.com/szczurek-lab/ProSpero.

2 Related Work

Evolutionary Algorithms Evolutionary algorithms are a common approach in protein sequence design [16, 7, 6]. Notably, Sinai et al. [7] proposed a greedy algorithm, AdaLead, which adaptively mutates and recombines high-fitness sequences selected by a threshold-based filter to balance exploration and exploitation. Ren et al. [6] introduced an exploration method, PEX, designed to exploit the local neighborhood of the wild-type by prioritizing variants with fewer mutations.

Machine-Learning-Assisted Directed Evolution (MLDE) To enhance traditional directed evolution [17], MLDE leverages machine learning models to predict sequence fitness and guide the selection of promising mutants for screening [1]. Qin et al. [15] iteratively fine-tune ESM-1b [18] with data annotated by a surrogate model. Tran and Hy [13] mask k-mers of a wild-type sequence and use ESM-2 [19] to propose new candidate sequences. The work of Qiu et al. [20], Qiu and Wei [21] and Wang et al. [22] explores clustering amino acids with similar properties; however, it differs from our approach by assuming a small number of fixed mutational sites.

Reinforcement Learning and Generative Flow Networks (GFNs) DyNaPPO uses an ensemble of surrogate models with varying architectures to train a generative policy that constructs candidate sequences amino acid by amino acid [8]. Rather than acting in the sequence space, LatProtRL [10] learns a generative policy operating in the latent space of ESM-2 [19]. GFNs use a surrogate model to learn a stochastic policy that samples sequences proportionally to their predicted fitness values [9]; however, when the proxy is misspecified, they can perform poorly [11, 12]. To address this, Kim et al. [12] proposed GFN-AL-δCS, a strategy enabling to control a trade-off between novelty and robustness based on uncertainty of the proxy.

Bayesian Optimization (BO) BO is a commonly used framework for optimizing expensive black-box functions and has been widely applied to the design of biological sequences under limited evaluation budgets [23–26]. Among these approaches, Amin et al. [26] optimize antibodies by sampling from a LLM trained on clonal families, using a twisted SMC procedure to incorporate knowledge about previous experimental measurements. A detailed comparison of PROSPERO and Amin et al. [26] can be found in Appendix E.

Generative and Energy-Based Models Frey et al. [27] use Langevin Markov-Chain Monte Carlo to sample from smoothed data distributions for antibody discovery. Kirjner et al. [28] construct a smoothed version of the fitness landscape prior to training a surrogate model, whose gradients are then used to guide the design of new sequences. However, the approach depends on a smoothing-strength hyperparameter, which is highly sensitive to the underlying fitness landscape and difficult to tune in practice. Frameworks introduced by Brookes et al. [14] or Song and Li [29] can propose new candidate sequences by sampling from generative models like VAEs [30]. Ghaffari et al. [31] present a VAE with a fitness-structured latent space, enabling robust optimization despite the sparsity and ruggedness of the underlying fitness landscape.

3 Problem formulation

We aim to discover protein sequences $x_{1:L} \in \mathcal{A}^L$ with high fitness $y \in \mathbb{R}$, where $\mathcal{A}$ denotes the vocabulary of 20 natural amino acids and L represents the length of the sequence. Unless emphasis on sequential structure is needed, we simply write x for brevity. The fitness value y measures a given property of a protein, such as binding affinity or fluorescence intensity. Designed sequences are evaluated by a black-box oracle $f : \mathcal{A}^L \to \mathbb{R}$. Since oracle evaluations—such as wet-lab experiments or costly computational simulations—are expensive and time-consuming, queries are limited to a batch of K sequences per a small number of rounds N. We assume access to an initial dataset $\mathcal{D}_0 = \{(x^{(i)}, y^{(i)})\}_{i=1}^M$ which allows to train a cheap and fast to query surrogate model $f_\theta : \mathcal{A}^L \to \mathbb{R}$, approximating the oracle. Additionally, let $x_{\text{start}} = \arg\max_{x \in \mathcal{D}_0} y$ correspond to the wild-type sequence. The main goal is to design protein sequences with high fitness values assigned by the oracle across N active learning iterations. Desirably, generated sequences should also be biologically plausible, novel, and diverse.

4 Proposed framework

Algorithm 1: Active Learning with PROSPERO

Input: Oracle f, proxy f_θ, initial dataset $\mathcal{D}_0$, active learning rounds N, pre-trained generative model $\mathcal{P}$, oracle budget K, SMC batch size B

1 **for** $n \leftarrow 1$ **to** N **do**

2 Fit f_θ on dataset $\mathcal{D}_{n-1}$:
$\theta \leftarrow \arg\min_\theta \mathbb{E}_{x \sim D_{n-1}} \left[(f(x) - f_\theta(x))^2 \right]$

3 Select starting sequence:
$x_{\text{start}} \leftarrow \arg\max_{x \in \mathcal{D}_{n-1}} f(x)$

4 Get masked variants of x_{start}:
$\left\{ \tilde{x}^{(i)} \right\}_{i=1}^{B} \leftarrow \textsc{TargetedMasking}(x_{\text{start}}, f_\theta)$ `// Algorithm 2`

5 Propose new candidate sequences:
$\left\{ x^{(i)} \right\}_{i=1}^{K} \leftarrow \textsc{ConstrainedSMC}\left(\left\{ \tilde{x}^{(i)} \right\}_{i=1}^{B}, \mathcal{P}, f_\theta \right)$ `// Algorithm 3`

6 Evaluate candidates with the oracle:
$\hat{\mathcal{D}}_n \leftarrow \left\{ (x^{(i)}, f(x^{(i)})) \right\}_{i=1}^{K}$

7 Update dataset:
$\mathcal{D}_n \leftarrow \mathcal{D}_{n-1} \cup \hat{\mathcal{D}}_n$

PROSPERO follows the active learning loop illustrated in Figure 1 and outlined in Algorithm 1, consisting of: (i) training the surrogate model f_θ on the current dataset $\mathcal{D}_{n-1}$ by minimizing the loss $\mathcal{L}(\theta) = \mathbb{E}_{x \sim D_{n-1}} \left[(f(x) - f_\theta(x))^2 \right]$; (ii) identifying and masking fitness-relevant residues in x_{start} with *targeted masking*; (iii) sampling new candidate sequences using *biologically-constrained SMC*; (iv) evaluating candidates with the oracle f and augmenting $\mathcal{D}_{n-1}$. As demonstrated by Antoniuk et al. [32], the use of the active learning loop is expected to expand the support of the surrogate model, thereby providing a more reliable guiding signal to the pre-trained generative model. In the remainder of this section, we describe two core innovations of our framework: the targeted masking and the biologically-constrained SMC (line 4 and line 5 of Algorithm 1, respectively), which represent the key methodological advances of PROSPERO over prior approaches.

4.1 Targeted masking

Our targeted masking strategy is inspired by alanine scanning, a mutagenesis technique used both experimentally and *in silico* to identify functionally important residues by substituting each position with alanine—a neutral amino acid that disrupts side-chain interactions [33–36]. Traditionally, alanine scanning aims to locate critical residues one at a time based on wet-lab experiments or *in silico* structural modeling. In PROSPERO, we propose a batched strategy that operates purely in sequence space to identify positions within x_{start} that are fitness-relevant but at the same time tolerant to mutation. Specifically, we construct S batches of B mutated sequences (denoted collectively as $\{x^{(i)}\}_{i=1}^{B \cdot S}$) by randomly substituting a subset of residues at locations $\mathcal{I}^{(i)} \subset \{1, \ldots, L\}$ with alanine. Each such mutated sequence is scored by the surrogate model f_θ, which returns a predictive mean and uncertainty estimate: $f_\theta(x^{(i)}) = (\mu_\theta(x^{(i)}), \sigma_\theta(x^{(i)}))$. We select the top B sequences according to the Upper Confidence Bound (UCB) [37] acquisition function, identifying substitutions that are not immediately harmful yet exhibit uncertainty suggestive of functional relevance of the affected residues. For each selected sequence, we construct a partially masked sequence $\tilde{x}^{(i)}$ by replacing previously substituted positions with a mask token: $\tilde{x}^{(i)}[j] = \texttt{[MASK]}$ if $j \in \mathcal{I}^{(i)}$, and $\tilde{x}^{(i)}[j] = x_{\text{start}}[j]$ otherwise. The resulting batch $\{\tilde{x}^{(i)}\}_{i=1}^{B}$ is then used as input to the guided generative procedure described next.

4.2 Biologically-constrained Sequential Monte Carlo

To design sequences with high fitness, one aims to sample from the posterior $p(x \mid y) \propto p(y \mid x)\, p(x)$. Since querying the true fitness function given by an oracle f is assumed to be expensive, $p(y \mid x)$ is

typically approximated using a surrogate f_θ. This yields the following target posterior distribution:

$$\gamma(x) = \frac{f_\theta(x) \cdot \mathcal{P}(x)}{Z}, \tag{1}$$

where $\mathcal{P}(x)$ denotes a prior over sequences and Z is a normalization constant. $\mathcal{P}(x)$ can be modeled in various ways; here, we use EvoDiff-OADM [38]. This formulation poses two key challenges: (i) surrogate models may exhibit low fidelity on out-of-distribution sequences, and (ii) direct sampling from $\gamma(x)$ is infeasible due to the intractability of Z. Next, we describe how PROSPERO overcomes these challenges.

Addressing surrogate misspecification with biologically constrained exploration To encourage biologically plausible exploration even under potential surrogate misspecification, we constrain candidate sampling in PROSPERO by leveraging the *charge class* of wild-type residues as an explicit biological prior. This draws inspiration from reduced amino acid alphabets (RAAs), which simplify sequence space by grouping residues with similar physicochemical and functional properties, exploiting the many-to-one relationship between sequence and structure [39, 40]. Biasing exploration toward substitutions within the same class favors alternative sequences that are more likely to preserve wild-type fitness, irrespective of the surrogate quality. We select charge as the grouping criterion, given its both fundamental and universal role in stabilizing protein structure via salt bridges, hydrogen bonding, and electrostatic interactions [41]. Specifically, for each partially masked sequence $\tilde{x}^{(i)}$, we further divide the set of masked positions $\mathcal{I}^{(i)}$ into three disjoint subsets: positive $\mathcal{I}^{(i)}_{(+)} = \{j \in \mathcal{I}^{(i)} \mid x_{\text{start}}[j] \in \{\text{R}, \text{K}, \text{H}\}\}$; negative $\mathcal{I}^{(i)}_{(-)} = \{j \in \mathcal{I}^{(i)} \mid x_{\text{start}}[j] \in \{\text{D}, \text{E}\}\}$; and neutral $\mathcal{I}^{(i)}_{(0)} = \{j \in \mathcal{I}^{(i)} \mid x_{\text{start}}[j] \notin \{\text{R}, \text{K}, \text{H}, \text{D}, \text{E}\}\}$. We formalize constrained sampling as sampling from the conditional distribution $\mathcal{P}_{RAA}(\cdot \mid \cdot)$, defined over normalized logits of the base model $\mathcal{P}$ restricted to amino acids in the same charge class as the wild-type residues at positions $\mathcal{I}^{(i)}$.

Sampling from an intractable distribution using Sequential Monte Carlo To sample from the intractable target distribution $\gamma(x)$, in PROSPERO we perform approximate inference using SMC. Rather than directly sampling from complex, high-dimensional $\gamma(x_{1:L})$ defined over full sequences, SMC decomposes the problem into sequential sampling from a series of simpler, unnormalized intermediate target distributions $\{\tilde{\gamma}_l(x_{1:l})\}_{l=1}^{L}$, relying on a tractable proposal distribution and resampling based on intermediate importance weights (for background, refer to Appendix C). This allows us to sample sequences from the approximate target posterior in a residue-by-residue manner. We start by capitalizing on EvoDiff's order-agnostic nature by defining the sampling permutation order for each $\tilde{x}^{(i)}$ as $\pi^{(i)} = \text{concat}(\{j \notin \mathcal{I}^{(i)}\}, \mathcal{I}^{(i)}_{(-)}, \mathcal{I}^{(i)}_{(+)}, \mathcal{I}^{(i)}_{(0)})$. For simplicity, we assume that all sequences in the batch share the same number of masked positions $|\mathcal{I}| = \max_{i=1}^{B} |\mathcal{I}^{(i)}|$. In practice, for sequences with fewer masked tokens, no new proposals are made once all masked positions have been filled, but these sequences remain in the population and are still included in weighting and resampling (see Algorithm 3). We proceed by performing the following operations at each unmasking step $t = L - |\mathcal{I}| + 1, \ldots, L$:

(i) **Constrained proposal**: for each $\tilde{x}^{(i)}$, we sample an amino acid at position $\pi^{(i)}(t)$ from the constrained base model: $\tilde{x}^{(i)}_{\pi(t)} \sim \mathcal{P}_{RAA}(\tilde{x}^{(i)}_{\pi(t)} \mid \tilde{x}^{(i)}_{\pi(<t)})$.

(ii) **Weighting**: since the surrogate model f_θ operates only on fully unmasked sequences, intermediate targets $\tilde{\gamma}_t(\tilde{x}_{\pi(\leq t)})$ are defined only implicitly and cannot be used directly to compute importance weights w_t. Instead, we approximate w_t by first rolling out the remainder of each sequence with the base model:

$$x^{(i)}_{\text{unroll}} \sim \prod_{s=t+1}^{T} \mathcal{P}_{RAA}(\tilde{x}^{(i)}_{\pi(s)} \mid \tilde{x}^{(i)}_{\pi(<s)}), \tag{2}$$

followed by scoring it with the surrogate model $\hat{y}^{(i)} = \mu_\theta(x^{(i)}_{\text{unroll}}) + k \cdot \sigma_\theta(x^{(i)}_{\text{unroll}})$. As the logits of the base model are constrained to charge-compatible amino acids, samples from $\mathcal{P}_{RAA}$ do not reflect the true prior over sequences. We therefore compute the perplexity of the unconstrained model $\mathcal{P}$, compensating for cases where $\mathcal{P}_{RAA}$ assigns uniformly low

likelihoods across all available choices:

$$\mathrm{Perp}(x^{(i)}_{\text{unroll}}) = \exp\left(-\frac{1}{|\mathcal{I}|}\sum_{s=T-|\mathcal{I}|+1}^{T} \log \mathcal{P}(x^{(i)}_{\text{unroll}_{\pi(s)}} \mid x^{(i)}_{\text{unroll}_{\pi(<s)}})\right). \tag{3}$$

Finally, the unnormalized importance weights for each sequence $\tilde{x}^{(i)}$ are computed as $w_t^{(i)} = \hat{y}^{(i)} / \mathrm{Perp}(x^{(i)}_{\text{unroll}})$, with the perplexity term correcting bias introduced by the constrained sampling.

(iii) **Resampling**: we sample a new population of partially masked sequences based on their normalized weights, effectively discarding sequences improbable under $\tilde{\gamma}_t(\tilde{x}_{\pi(\leq t)})$:

$$\tilde{x}^{(i)} \sim \mathrm{Cat}\left(\{\tilde{x}^{(i)}\}_{i=1}^{B}, \left\{\frac{w_t^{(i)}}{\sum_{j=1}^{B} w_t^{(j)}}\right\}_{i=1}^{B}\right). \tag{4}$$

After all sequences have been fully unmasked, we select the top K candidates for oracle evaluation by ranking both the final population and intermediate rollouts from the last n_{keep} unmasking steps according to their predicted UCB scores $\hat{y}$.

5 Experiments

We show that PROSPERO successfully balances generation of high-fitness sequences with exploration, while maintaining both diversity (Section 5.1) and biological plausibility (Section 5.2). In Section 5.3 and Section 5.4, we demonstrate the robustness of our approach under different forms of surrogate misspecification, namely covariate shift and surrogate noise. Finally, Section 5.5 investigates the contribution of individual components of our framework to overall performance. We evaluate PROSPERO based on the following general setup, which serves as the basis for all subsequent experiments.

Datasets and oracles We evaluate our method on eight diverse protein engineering tasks, details of which can be found in Appendix A.1. For the AAV landscape, we use ground-truth fitness scores provided in FLEXS [7]. For all other tasks, following Ren et al. [6], we use TAPE [42] as the oracle f to simulate wet-lab experiments. Similarly to Kim et al. [12], we replace experimental measurements in each initial dataset with scores assigned by the oracle model.

Baselines We compare our approach against a suite of established methods for iterative biological sequence design, covering diverse algorithmic paradigms: (i) evolutionary algorithms—AdaLead, PEX and CMA-ES [7, 6, 16]; (ii) on-policy reinforcement learning—DyNaPPO and LatProtRL [8, 10]; (iii) GFlowNets—GFN-AL and GFN-AL-δCS [9, 12]; (iv) evolutionary Bayesian Optimization (BO) [7]; (v) probabilistic framework CbAS [14]; and (vi) the machine-learning-assisted directed evolution (MLDE) approach of Tran and Hy [13]. Further details are provided in Appendix A.4.

Implementation details We employ PROSPERO with $N = 10$ active learning rounds, each ending with a batch of $K = 128$ sequences evaluated by the oracle f. To model the proxy f_θ, we follow Sinai et al. [7] and Kim et al. [12], and use an ensemble of three one-dimensional convolutional neural networks. This architecture is shared across all baselines to ensure a fair comparison. Sequence selection is guided by the UCB acquisition function $f_\theta(x) = \mu_\theta(x) + k \cdot \sigma_\theta(x)$, where $k = 1$ for the targeted masking and $k = 0.1$ for the biologically-constrained SMC. The SMC batch size is set to $B = 256$, and during generation we retain rollouts from the last $n_{\text{keep}} = 10$ unmasking steps. In the targeted masking, the number of alanine scans $S = 16$, with 3–10 mutations for shorter sequences ($L \approx 100$) as in the AAV, E4B, and Pab1 landscapes, and 5–15 for longer ones.

Evaluation metrics We evaluate performance of all methods using four primary metrics, with particular emphasis on (i) *maximum fitness*, defined as the highest fitness value achieved among all generated sequences. This reflects the primary objective of discovering high-performing candidates. We also report: (ii) *mean fitness* of the top 100 generated sequences; (iii) *novelty*, measured as the average Hamming distance between the top 100 sequences and the starting sequence x_{start}; and (iv) *diversity*, defined as the average pairwise Hamming distance among the top 100 sequences.

Table 1: Maximum fitness values achieved by each method. Reported values are the mean and standard deviation over 5 runs. Green denotes fitness improvement over wild-type x_{start}. **Bold:** the best overall fitness per each task. Underline: second-best. PROSPERO improves fitness on every task, ranking first on 5 out of 8 and second on the remaining 3.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
CMA-ES	-6.857 ± 0.257	0.037 ± 0.01	-0.429 ± 0.252	0.553 ± 0.038	0.000 ± 0.000	1.972 ± 0.135	0.135 ± 0.178	-1.337 ± 0.021
DyNaPPO	-3.683 ± 0.575	0.067 ± 0.008	3.924 ± 0.883	0.783 ± 0.036	0.009 ± 0.018	3.550 ± 0.012	2.796 ± 0.059	-0.007 ± 0.015
BO	0.168 ± 0.056	0.682 ± 0.369	7.442 ± 0.242	0.814 ± 0.081	0.667 ± 0.024	3.584 ± 0.007	2.883 ± 0.069	0.026 ± 0.003
PEX	**0.248 ± 0.007**	**1.232 ± 0.000**	8.099 ± 0.017	1.499 ± 0.343	0.665 ± 0.022	3.603 ± 0.003	2.991 ± 0.001	0.037 ± 0.001
AdaLead	0.235 ± 0.002	1.228 ± 0.002	8.034 ± 0.036	**1.978 ± 0.188**	0.683 ± 0.037	3.581 ± 0.003	2.985 ± 0.002	0.038 ± 0.001
CbAS	-8.202 ± 0.032	0.019 ± 0.002	-0.569 ± 0.092	0.351 ± 0.043	0.000 ± 0.000	1.858 ± 0.067	-0.056 ± 0.003	-1.492 ± 0.035
GFN-AL	-7.853 ± 0.270	0.027 ± 0.020	0.160 ± 0.228	0.507 ± 0.025	0.000 ± 0.000	2.004 ± 0.022	0.271 ± 0.443	-1.164 ± 0.118
GFN-AL-δCS	0.203 ± 0.005	0.701 ± 0.148	7.930 ± 0.055	1.297 ± 0.337	0.686 ± 0.021	3.589 ± 0.006	2.984 ± 0.002	0.033 ± 0.001
LatProtRL	0.224 ± 0.000	1.229 ± 0.000	7.902 ± 0.086	1.122 ± 0.152	0.593 ± 0.018	3.590 ± 0.003	2.983 ± 0.000	0.020 ± 0.000
MLDE	0.241 ± 0.003	1.229 ± 0.000	7.934 ± 0.077	0.896 ± 0.015	0.555 ± 0.000	3.596 ± 0.003	2.984 ± 0.003	0.038 ± 0.002
PROSPERO	0.246 ± 0.006	1.231 ± 0.002	**8.114 ± 0.037**	1.527 ± 0.254	**0.720 ± 0.027**	**3.617 ± 0.002**	**2.993 ± 0.003**	**0.043 ± 0.002**

5.1 Protein design evaluation

Fitness optimization PROSPERO consistently achieves superior or comparable performance to the baselines in generating candidate sequences with high fitness, irrespective of the underlying fitness landscape (Table 1). Our approach obtains the highest maximum fitness values on 5 out of 8 protein engineering tasks and ranks second on the remaining 3. Notably, among the 11 evaluated methods, only PROSPERO and PEX are able to achieve fitness improvements over wild-type x_{start} across *all* landscapes, highlighting their reliability in diverse optimization scenarios. In contrast, 4 methods fail to achieve improvements on any task. We report further results only for 6 out of 11 methods that managed to improve fitness on at least half of the tasks. Results in Table 2 demonstrate that PROSPERO is consistently able to generate a broad set of high-fitness candidates, achieving the highest mean fitness among the top 100 sequences on 5 out of 8 landscapes and ranking second on 2. Notably, mean fitness fell below that of x_{start} on only a single task. Additionally, Figure 2 shows that our method discovers high-fitness sequences at earlier active learning rounds than competing approaches on half of the evaluated tasks. A detailed comparison of early-round performance across all benchmarks is provided in Appendix D.2.

Exploration and diversity As showcased in Table 3, PROSPERO attains high-fitness solutions with substantially greater novelty compared to other leading approaches, outperforming them on 6 out of 8 tasks. Remarkably, PROSPERO frequently breaks the conventional Pareto frontier between sequence fitness and novelty, achieving levels of both that remain mutually constraining for the competing methods (Figure 3). In particular, although PEX—the second-best performing in terms of maximum fitness—is designed to exploit the local neighborhood of the wild-type, our method often matches or exceeds its fitness while achieving approximately 2 to 9 times greater novelty. Importantly, the exceptional performance of PROSPERO in both fitness and novelty does not come at the expense of diversity. As shown in Table 9 in Appendix D.1, among leading approaches, only GFN-AL-δCS exceeds our method in diversity. However, it generates lower fitness sequences across all tasks, highlighting PROSPERO's ability to maintain diversity without compromising performance.

5.2 Biological plausibility

Setup We assess biological plausibility on the E4B task, where a large reference dataset of over 80,000 sequences not included in $\mathcal{D}_0$ is available (see Appendix A.1). Following the approach

Table 2: Mean fitness of top 100 sequences generated by leading methods. Reported values are the mean and standard deviation over 5 runs. Green: fitness improvement over wild-type x_{start}. **Bold:** the best overall fitness per each task. Underline: second-best. PROSPERO improves fitness on 7 out of 8 tasks, ranking first on 5 and second on 2.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
PEX	**0.238 ± 0.004**	**1.227 ± 0.002**	7.948 ± 0.046	1.307 ± 0.258	0.620 ± 0.017	3.597 ± 0.003	**2.987 ± 0.001**	0.033 ± 0.001
AdaLead	0.229 ± 0.001	1.201 ± 0.002	7.846 ± 0.040	**1.836 ± 0.266**	0.644 ± 0.031	3.563 ± 0.007	2.976 ± 0.003	0.037 ± 0.001
GFN-AL-δCS	-0.244 ± 0.137	0.192 ± 0.027	7.653 ± 0.136	1.070 ± 0.113	0.648 ± 0.020	3.569 ± 0.009	2.968 ± 0.006	0.024 ± 0.004
LatProtRL	0.217 ± 0.001	1.222 ± 0.000	7.562 ± 0.06	0.888 ± 0.072	0.563 ± 0.009	3.582 ± 0.003	2.975 ± 0.001	0.019 ± 0.000
MLDE	0.231 ± 0.004	1.131 ± 0.021	7.843 ± 0.122	0.877 ± 0.024	0.555 ± 0.000	3.591 ± 0.003	2.975 ± 0.005	0.036 ± 0.002
PROSPERO	0.236 ± 0.007	1.176 ± 0.029	**8.017 ± 0.054**	1.401 ± 0.202	**0.679 ± 0.025**	**3.613 ± 0.002**	**2.987 ± 0.003**	**0.040 ± 0.002**

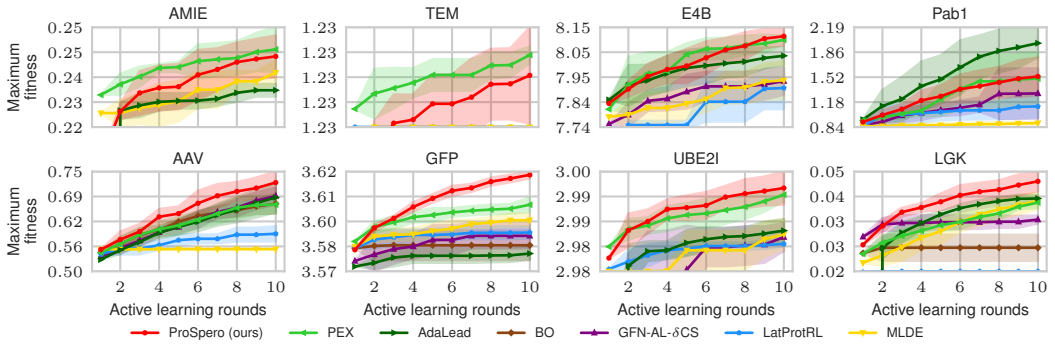

Figure 2: Maximum fitness recovered over 10 active learning rounds. Only methods that improved over x_{start} are shown. Shaded regions indicate standard deviation across 5 runs. PROSPERO retrieves high-fitness sequences in earlier rounds than baselines on 4 out of 8 tasks.

Table 3: Average novelty of top 100 sequences generated by leading methods. Reported values are the mean and standard deviation over 5 runs. **Bold:** the best overall novelty. Underline: second-best. PROSPERO ranks first on 6 out of 8 tasks and second on 1.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
PEX	5.19 ± 1.08	1.79 ± 0.21	3.96 ± 0.54	5.29 ± 0.74	7.29 ± 1.38	6.23 ± 1.34	4.55 ± 0.82	8.45 ± 1.39
AdaLead	4.21 ± 0.89	2.93 ± 0.07	3.92 ± 0.62	8.47 ± 1.58	9.86 ± 1.55	33.79 ± 5.84	3.92 ± 0.47	14.75 ± 8.02
GFN-AL-δCS	8.29 ± 0.42	$\mathbf{10.10 \pm 0.65}$	4.94 ± 1.59	10.29 ± 1.29	9.70 ± 0.33	34.83 ± 3.77	8.01 ± 3.69	63.16 ± 4.24
LatProtRL	1.09 ± 0.04	1.10 ± 0.01	3.11 ± 0.49	3.85 ± 1.19	3.03 ± 0.40	12.73 ± 2.21	1.69 ± 0.05	1.71 ± 0.01
MLDE	19.02 ± 2.69	4.28 ± 0.55	$\mathbf{11.88 \pm 3.49}$	9.67 ± 3.12	4.48 ± 0.25	23.65 ± 5.40	9.60 ± 2.58	50.09 ± 8.08
PROSPERO	$\mathbf{20.99 \pm 3.32}$	3.37 ± 0.53	8.81 ± 0.98	$\mathbf{11.83 \pm 3.52}$	$\mathbf{15.03 \pm 1.59}$	$\mathbf{39.85 \pm 3.95}$	$\mathbf{16.45 \pm 5.11}$	$\mathbf{74.33 \pm 7.75}$

of Surana et al. [11], we define *validity* as the percentage of top 100 generated sequences whose key physicochemical properties fall within the central 99% quantiles of the corresponding property distributions in the reference set. Additionally, for landscapes where x_{start} folds reliably, we further assess structural quality of generated sequences using pTM and pLDDT scores from ESMFold [19], as well as scPerplexity [38] computed after inverse folding with ESM-IF1 [43]. More information about the metrics is provided in Appendix A.5.

Results Figure 4A demonstrates that PROSPERO maintains biological plausibility while proposing sequences that are over twice as novel as those generated by baselines with comparable validity, such as PEX and LatProtRL. Moreover, it achieves this while also attaining higher fitness. Sequences generated by PROSPERO exhibit strong folding confidence, with pTM and pLDDT scores consistently above 70 (Figure 4B). Notably, this performance remains comparable to that of the wild-type even on the LGK landscape, where candidate sequences differ from the wild-type by over 70 amino acids.

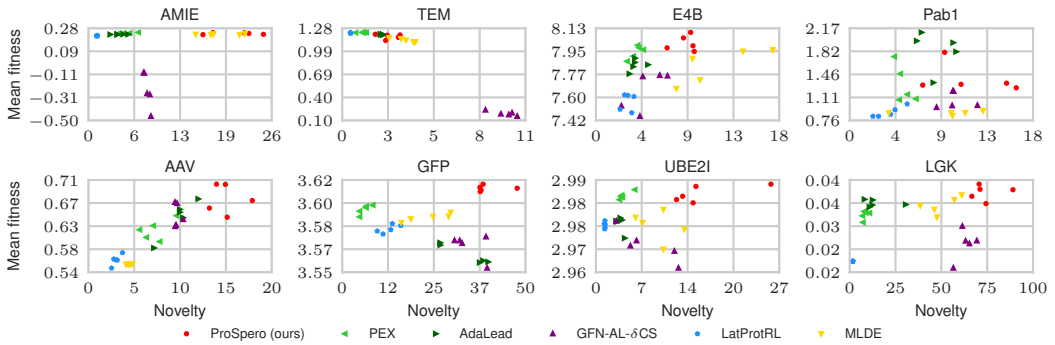

Figure 3: Comparison of fitness-novelty trade-offs among leading methods. Each dot represents the outcome of a single run. PROSPERO achieves both higher fitness and novelty than the baselines.

5.3 Out-of-distribution robustness

Setup For the simulation of distribution shifts, we used UBE2I variants and predicted their pTM scores with ESMFold [19] as the oracle. The surrogate model was trained only on sequences close to the wild-type, and optimization was then initiated from increasingly distant starting sequences x_{start}. We considered three cases: (i) a moderate covariate shift with x_{start} differing by 35 mutations from the wild-type; (ii) a severe covariate shift with x_{start} differing by 75 mutations (approximately half of the sequence length); (iii) low-data regime, where the 35-mutation case was repeated with the surrogate trained only on a small subset of the available data. Full details are provided in Appendix A.2.

Results Across all settings, PROSPERO consistently outperforms competing approaches, generating sequences with the highest pTM scores (Table 4), while simultaneously exploring more novel regions of the sequence space (Appendix D.3). The performance advantage over the baselines was especially pronounced under the severe shift, highlighting the effectiveness of our approach in the most challenging conditions. Even in the low-data regime, our method maintained strong performance, thereby demonstrating robustness to both distribution shifts and data scarcity. Taken together, these results show that PROSPERO remains effective in the challenging OOD settings, commonly faced when exploring sequence space beyond wild-type neighborhoods.

5.4 Noise robustness study

Setup We evaluate the robustness of exploration strategies to surrogate model misspecification by introducing increasingly noisy surrogates on the AAV landscape, where ground-truth fitness is available. Specifically, we replaced the surrogate f_θ with an ensemble of noisy oracles f_ϵ, each defined by adding zero-mean Gaussian noise to the ground-truth oracle and truncating negative outputs to zero, following the perturbation scheme of Sinai et al. [7]. The magnitude of the injected noise was determined by the Signal-to-Noise Ratio (SNR), with the noise scale given by $\sigma_{\text{noise}} = \sqrt{\text{Var}(\mathcal{D}_0) \cdot 10^{-\text{SNR}/10}}$, where $\text{Var}(\mathcal{D}_0)$ denotes the variance of fitness scores in the initial dataset. This setup introduces both stochastic noise and systematic shift, as the truncation flattens low-fitness regions and biases predictions upward, making it a strong test of robustness to surrogate error.

Results As shown in Figure 4C, PROSPERO maintains an advantage over the majority of the baselines even at low SNR levels, demonstrating strong robustness to surrogate noise. The performance gap between our method and competing approaches widens as the noise levels decrease, highlighting PROSPERO's ability to increasingly capitalize on informative signal. Notably, our method and AdaLead exhibit a sharp performance improvement earlier than other methods, suggesting greater robustness to surrogate misspecification and ability to guide exploration toward promising regions of the search space more effectively than competing approaches.

5.5 Ablation

Setup To assess the contribution of individual components in PROSPERO, we conduct ablation studies under the same noisy surrogate setting as in the robustness analysis (Section 5.4). We compare the full method to the following ablations: (i) without SMC, corresponding to sampling from EvoDiff

Table 4: Maximum and mean pTM scores of top 100 sequences generated by leading methods under distribution shifts. Reported values are the mean and standard deviation over 5 runs. **Bold:** the best overall pTM score. Underline: second-best. PROSPERO demonstrates the highest robustness to covariate shifts.

	Moderate shift		Severe shift		Low-data shift	
Method	Max	Mean	Max	Mean	Max	Mean
PEX	0.807 ± 0.023	$\underline{0.760 \pm 0.012}$	0.578 ± 0.014	0.518 ± 0.003	$\underline{0.806 \pm 0.013}$	$\underline{0.752 \pm 0.005}$
AdaLead	0.796 ± 0.013	0.755 ± 0.011	0.593 ± 0.028	0.526 ± 0.007	0.781 ± 0.016	0.742 ± 0.004
GFN-AL-δCS	0.791 ± 0.010	0.729 ± 0.005	0.630 ± 0.024	0.542 ± 0.006	0.782 ± 0.006	0.731 ± 0.006
LatProtRL	0.787 ± 0.013	0.743 ± 0.003	0.560 ± 0.000	0.508 ± 0.003	0.792 ± 0.013	0.743 ± 0.001
MLDE	$\underline{0.810 \pm 0.020}$	0.752 ± 0.004	$\underline{0.652 \pm 0.059}$	$\underline{0.572 \pm 0.035}$	0.782 ± 0.022	0.735 ± 0.025
PROSPERO	$\mathbf{0.822 \pm 0.027}$	$\mathbf{0.777 \pm 0.020}$	$\mathbf{0.672 \pm 0.031}$	$\mathbf{0.599 \pm 0.014}$	$\mathbf{0.808 \pm 0.017}$	$\mathbf{0.763 \pm 0.017}$

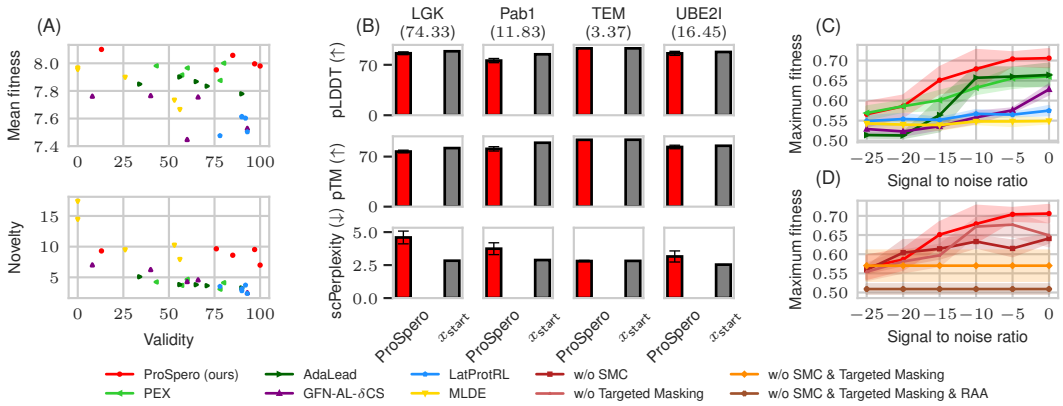

Figure 4: (A) Trade-offs between validity, fitness and novelty across leading methods; each dot represents the outcome of a single run. (B) Structural quality of top 100 sequences generated by PROSPERO across 5 runs compared to x_{start}; average novelty of generated sequences is shown below each task. (C) Performance of leading methods on the AAV landscape under varying levels of surrogate noise. (D) Ablation of PROSPERO components under the same setting as in (C). In both (C) and (D), shaded regions represent the standard deviation across 5 runs. PROSPERO generates highly biologically plausible sequences and remains robust to surrogate misspecification.

with the resampling steps omitted; (ii) without targeted masking, where masked positions are selected at random; (iii) without both SMC and targeted masking; and (iv) without SMC, without targeted masking, and without restricting SMC proposals to charge-compatible amino acids (without RAA).

Results Results of the ablation are depicted in Figure 4D. Notably, PROSPERO with all components intact outperforms all ablated versions, with performance degrading only under extremely low SNR conditions. The advantage of our method over the baselines at low SNR levels, as seen in the noise robustness study in Section 5.4, is most likely supported by PROSPERO's constraint that limits candidate generation to sequences containing residues of the same charge class as their wild-type counterparts. This steers the generation process toward sequences with wild-type fitness regardless of the surrogate quality, providing substantial performance gains. The sharp fitness improvement at higher SNR levels likely reflects the increasing effectiveness of the SMC guidance and the targeted masking as the surrogate signal improves, while at very low SNR levels, guidance appears to slightly hinder the performance. Further ablations are provided in Appendix F.

6 Conclusion

In this paper, we introduced PROSPERO, an active learning framework for iterative protein sequence design based on inference-time guidance of a pre-trained generative model. Our targeted masking strategy enables edits focused on fitness-relevant residues while preserving functionally critical sites. Biologically-constrained SMC sampling allows incorporating biological prior knowledge while traversing fitness landscapes, increasing the likelihood of retrieving high-fitness sequences even under surrogate misspecification. By combining these innovations, PROSPERO enables robust exploration beyond wild-type neighborhoods while maintaining biological plausibility, achieving performance that matches or exceeds state-of-the-art approaches across diverse protein engineering tasks.

Acknowledgments and Disclosure of Funding

We thank Rasmus Møller-Larsen, James Odgers, and Adam Izdebski for their valuable feedback and suggestions that helped us improve the manuscript. This project has received funding from the European Research Council (ERC) under the European Funding Union's Horizon 2020 research and innovation programme (grant agreement No 810115 – DOG-AMP). VF was supported by the Branco Weiss Fellowship.

References

[1] Kevin K. Yang, Zachary Wu, and Frances H. Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694, Aug 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0496-6. URL https://doi.org/10.1038/s41592-019-0496-6.

[2] H A Daniel Lagassé, Aikaterini Alexaki, Vijaya L Simhadri, Nobuko H Katagiri, Wojciech Jankowski, Zuben E Sauna, and Chava Kimchi-Sarfaty. Recent advances in (therapeutic protein) drug development. *F1000Res.*, 6:113, February 2017.

[3] Lynne Regan, Diego Caballero, Michael R Hinrichsen, Alejandro Virrueta, Danielle M Williams, and Corey S O'Hern. Protein design: Past, present, and future. *Biopolymers*, 104(4):334–350, July 2015.

[4] S. Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. *Proceedings of the XI International Congress of Genetics*, 8:209–222, 1932.

[5] J. Arjan G.M. de Visser and Joachim Krug. Empirical fitness landscapes and the predictability of evolution. *Nature Reviews Genetics*, 15(7):480–490, Jul 2014. ISSN 1471-0064. doi: 10.1038/nrg3744. URL https://doi.org/10.1038/nrg3744.

[6] Zhizhou Ren, Jiahan Li, Fan Ding, Yuan Zhou, Jianzhu Ma, and Jian Peng. Proximal exploration for model-guided protein sequence design. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18520–18536. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ren22a.html.

[7] Sam Sinai, Richard Wang, Alexander Whatley, Stewart Slocum, Elina Locane, and Eric Kelsic. Adalead: A simple and robust adaptive greedy search algorithm for sequence design. *arXiv preprint*, 2020.

[8] Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HklxbgBKvr.

[9] Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with GFlowNets. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9786–9801. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/jain22a.html.

[10] Minji Lee, Luiz Felipe Vecchietti, Hyunkyu Jung, Hyun Joo Ro, Meeyoung Cha, and Ho Min Kim. Robust optimization in protein fitness landscapes using reinforcement learning in latent space. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 26976–26990. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/lee24x.html.

[11] Shikha Surana, Nathan Grinsztajn, Timothy Atkinson, Paul Duckworth, and Thomas D Barrett. Overconfident oracles: Limitations of in silico sequence design benchmarking. In *ICML 2024 AI for Science Workshop*, 2024. URL https://openreview.net/forum?id=fPBCnJKXUb.

[12] Hyeonah Kim, Minsu Kim, Taeyoung Yun, Sanghyeok Choi, Emmanuel Bengio, Alex Hernández-García, and Jinkyoo Park. Improved off-policy reinforcement learning in biological sequence design. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=0TY5lhhdZm.

[13] Thanh V. T. Tran and Truong Son Hy. Protein design by directed evolution guided by large language models. *IEEE Transactions on Evolutionary Computation*, 29(2):418–428, 2025. doi: 10.1109/TEVC.2024.3439690.

[14] David Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 773–782. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/brookes19a.html.

[15] Ming Qin, Keyan Ding, Bin Wu, Haihong Yang, Zeyuan Wang, Hongbin Ye, Haoran Yu, Huajun Chen, and Qiang Zhang. *Active Finetuning Protein Language Model: A Budget-Friendly Method for Directed Evolution*. Frontiers in Artificial Intelligence and Applications. IOS Press, 09 2023. ISBN 9781643684369. doi: 10.3233/FAIA230481.

[16] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 06 2001. ISSN 1063-6560. doi: 10.1162/106365601750190398. URL https://doi.org/10.1162/106365601750190398.

[17] Frances H. Arnold. Design by directed evolution. *Accounts of Chemical Research*, 31(3): 125–131, 1998. doi: 10.1021/ar960017f. URL https://doi.org/10.1021/ar960017f.

[18] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2019. doi: 10.1101/622803. URL https://www.biorxiv.org/content/10.1101/622803v4.

[19] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[20] Yuchi Qiu, Jian Hu, and Guo-Wei Wei. Cluster learning-assisted directed evolution. *Nature Computational Science*, 1(12):809–818, Dec 2021. ISSN 2662-8457. doi: 10.1038/s43588-021-00168-y. URL https://doi.org/10.1038/s43588-021-00168-y.

[21] Yuchi Qiu and Guo-Wei Wei. CLADE 2.0: Evolution-driven cluster learning-assisted directed evolution. *J. Chem. Inf. Model.*, 62(19):4629–4641, October 2022.

[22] Yuhao Wang, Qiang Zhang, Ming Qin, Xiang Zhuang, Xiaotong Li, Zhichen Gong, Zeyuan Wang, Yu Zhao, Jianhua Yao, Keyan Ding, and Huajun Chen. Knowledge-aware reinforced language models for protein directed evolution. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=MikandLqtW.

[23] Dinghuai Zhang, Jie Fu, Yoshua Bengio, and Aaron Courville. Unifying likelihood-free inference with black-box optimization and beyond, 2022. URL https://arxiv.org/abs/2110.03372.

[24] Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating bayesian optimization for biological sequence design with denoising autoencoders. *arXiv preprint arXiv:2203.12742*, 2022.

[25] Nate Gruver, Samuel Stanton, Nathan C. Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion, 2023. URL https://arxiv.org/abs/2305.20009.

[26] Alan Nawzad Amin, Nate Gruver, Yilun Kuang, Yucen Lily Li, Hunter Elliott, Calvin McCarter, Aniruddh Raghu, Peyton Greenside, and Andrew Gordon Wilson. Bayesian optimization of antibodies informed by a generative model of evolving sequences. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=E48QvQppIN.

[27] Nathan C. Frey, Dan Berenberg, Karina Zadorozhny, Joseph Kleinhenz, Julien Lafrance-Vanasse, Isidro Hotzel, Yan Wu, Stephen Ra, Richard Bonneau, Kyunghyun Cho, Andreas Loukas, Vladimir Gligorijevic, and Saeed Saremi. Protein discovery with discrete walk-jump sampling. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zMPHKOmQNb.

[28] Andrew Kirjner, Jason Yim, Raman Samusevich, Shahar Bracha, Tommi S. Jaakkola, Regina Barzilay, and Ila R Fiete. Improving protein optimization with smoothed fitness landscapes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=rxlF2Zv8x0.

[29] Zhenqiao Song and Lei Li. Importance weighted expectation-maximization for protein sequence design, 2024. URL https://arxiv.org/abs/2305.00386.

[30] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[31] Saba Ghaffari, Ehsan Saleh, Alex Schwing, Yu-Xiong Wang, Martin D. Burke, and Saurabh Sinha. Robust model-based optimization for challenging fitness landscapes. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=xhEN0kJh4q.

[32] Evan R. Antoniuk, Peggy Li, Nathan Keilbart, Stephen Weitzner, Bhavya Kailkhura, and Anna M. Hiszpanski. Active learning enables extrapolation in molecular generative models, 2025. URL https://arxiv.org/abs/2501.02059.

[33] B C Cunningham and J A Wells. High-resolution epitope mapping of hGH-receptor interactions by alanine-scanning mutagenesis. *Science*, 244(4908):1081–1085, June 1989.

[34] Kim L Morrison and Gregory A Weiss. Combinatorial alanine-scanning. *Current Opinion in Chemical Biology*, 5(3):302–307, 2001. ISSN 1367-5931. doi: https://doi.org/10.1016/S1367-5931(00)00206-4. URL https://www.sciencedirect.com/science/article/pii/S1367593100002064.

[35] Tanja Kortemme, David E Kim, and David Baker. Computational alanine scanning of protein-protein interfaces. *Sci. STKE*, 2004(219):l2, February 2004.

[36] Dennis M Krüger and Holger Gohlke. DrugScorePPI webserver: fast and accurate in silico alanine scanning for scoring protein-protein interactions. *Nucleic Acids Res.*, 38(Web Server issue):W480–6, July 2010.

[37] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 1015–1022, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

[38] Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex X. Lu, Nicolo Fusi, Ava P. Amini, and Kevin K. Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, 2023. doi: 10.1101/2023.09.11.556673. URL https://www.biorxiv.org/content/early/2023/09/12/2023.09.11.556673.

[39] J Wang and W Wang. A computational approach to simplifying the protein folding alphabet. *Nat. Struct. Biol.*, 6(11):1033–1038, November 1999.

[40] D S Riddle, J V Santiago, S T Bray-Hall, N Doshi, V P Grantcharova, Q Yi, and D Baker. Functional rapidly folding proteins from simplified amino acid sequences. *Nat. Struct. Biol.*, 4 (10):805–809, October 1997.

[41] Sandeep Kumar and Ruth Nussinov. Close-range electrostatic interactions in proteins. *Chembiochem*, 3(7):604–617, July 2002.

[42] R Rao, N Bhattacharya, N Thomas, Y Duan, X Chen, J Canny, P Abbeel, and Y S Song. Evaluating protein transfer learning with TAPE. *Adv Neural Inf Process Syst*, 32:9689–9701, 2019.

[43] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. *ICML*, 2022. doi: 10.1101/2022.04.10.487779. URL https://www.biorxiv.org/content/early/2022/04/10/2022.04.10.487779.

[44] Emily E. Wrenbeck, Laura R. Azouz, and Timothy A. Whitehead. Single-mutation fitness landscapes for an enzyme on multiple substrates reveal specificity is globally encoded. *Nature Communications*, 8(1):15695, Jun 2017. ISSN 2041-1723. doi: 10.1038/ncomms15695. URL https://doi.org/10.1038/ncomms15695.

[45] Elad Firnberg, Jason W. Labonte, Jeffrey J. Gray, and Marc Ostermeier. A comprehensive, high-resolution map of a gene's fitness landscape. *Molecular Biology and Evolution*, 31 (6):1581–1592, 02 2014. ISSN 0737-4038. doi: 10.1093/molbev/msu081. URL https://doi.org/10.1093/molbev/msu081.

[46] Lea M Starita, Jonathan N Pruneda, Russell S Lo, Douglas M Fowler, Helen J Kim, Joseph B Hiatt, Jay Shendure, Peter S Brzovic, Stanley Fields, and Rachel E Klevit. Activity-enhancing mutations in an e3 ubiquitin ligase identified by high-throughput mutagenesis. *Proceedings of the National Academy of Sciences of the United States of America*, 110(14):E1263—72, April 2013. ISSN 0027-8424. doi: 10.1073/pnas.1303309110. URL https://europepmc.org/articles/PMC3619334.

[47] Daniel Melamed, David L Young, Caitlin E Gamble, Christina R Miller, and Stanley Fields. Deep mutational scanning of an RRM domain of the saccharomyces cerevisiae poly(a)-binding protein. *RNA*, 19(11):1537–1551, November 2013.

[48] Pierce J Ogden, Eric D Kelsic, Sam Sinai, and George M Church. Comprehensive AAV capsid fitness landscape reveals a viral gene and enables machine-guided design. *Science*, 366(6469): 1139–1143, November 2019.

[49] Karen S Sarkisyan, Dmitry A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitry N Ivankov, Nina G Bozhanova, Mikhail S Baranov, Onuralp Soylemez, Natalya S Bogatyreva, Peter K Vlasov, Evgeny S Egorov, Maria D Logacheva, Alexey S Kondrashov, Dmitry M Chudakov, Ekaterina V Putintseva, Ilgar Z Mamedov, Dan S Tawfik, Konstantin A Lukyanov, and Fyodor A Kondrashov. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, May 2016.

[50] Jochen Weile, Song Sun, Atina G Cote, Jennifer Knapp, Marta Verby, Joseph C Mellor, Yingzhou Wu, Carles Pons, Cassandra Wong, Natascha van Lieshout, Fan Yang, Murat Tasan, Guihong Tan, Shan Yang, Douglas M Fowler, Robert Nussbaum, Jesse D Bloom, Marc Vidal, David E Hill, Patrick Aloy, and Frederick P Roth. A framework for exhaustively mapping functional missense variants. *Molecular Systems Biology*, 13(12):957, 2017. doi: https://doi.org/10.15252/msb.20177908. URL https://www.embopress.org/doi/abs/10.15252/msb.20177908.

[51] Justin R Klesmith, John-Paul Bacik, Ryszard Michalczyk, and Timothy A Whitehead. Comprehensive sequence-flux mapping of a levoglucosan utilization pathway in e. coli. *ACS Synth. Biol.*, 4(11):1235–1243, November 2015.

[52] Farhan Damani, David H Brookes, Theodore Sternlieb, Cameron Webster, Stephen Malina, Rishi Jajoo, Kathy Lin, and Sam Sinai. Beyond the training set: an intuitive method for detecting distribution shift in model-based optimization, 2023. URL https://arxiv.org/abs/2311.05363.

[53] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[54] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time, 2017. URL https://arxiv.org/abs/1610.10099.

[55] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140:107–113, 1993. doi: 10.1049/ip-f-2.1993.0015. URL https://digital-library.theiet.org/doi/abs/10.1049/ip-f-2.1993.0015.

[56] Christian A. Naesseth, Fredrik Lindsten, and Thomas B. Schön. Elements of sequential monte carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392, 2019.

[57] Arnaud Doucet, Nando Freitas, Kevin Murphy, and Stuart Russell. Sequential monte carlo methods in practice. 01 2013. doi: 10.1007/978-1-4757-3437-9_24.

[58] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt.

[59] Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XsgHl54yO7.

[60] Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=DmT862YAieY.

[61] Cheuk Kit Lee, Paul Jeha, Jes Frellsen, Pietro Lio, Michael Samuel Albergo, and Francisco Vargas. Debiasing guidance for discrete diffusion with sequential monte carlo, 2025. URL https://arxiv.org/abs/2502.06079.

[62] Luhuan Wu, Brian L. Trippe, Christian A Naesseth, John Patrick Cunningham, and David Blei. Practical and asymptotically exact conditional sampling in diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=eWKqr1zcRv.

[63] Brian L. Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi S. Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6TxBxqNME1Y.

[64] Filip Ekström Kelvinius and Fredrik Lindsten. Discriminator guidance for autoregressive diffusion models. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 3403–3411. PMLR, 02–04 May 2024. URL https://proceedings.mlr.press/v238/ekstrom-kelvinius24a.html.

[65] Dongjun Kim, Yeongmin Kim, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Refining generative process with discriminator guidance in score-based diffusion models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 16567–16598. PMLR, 2023. URL https://proceedings.mlr.press/v202/kim23i.html.

[66] Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Lm8T39vLDTE.

[67] Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding, 2024. URL https://arxiv.org/abs/2408.08252.

[68] Masatoshi Uehara, Xingyu Su, Yulai Zhao, Xiner Li, Aviv Regev, Shuiwang Ji, Sergey Levine, and Tommaso Biancalani. Reward-guided iterative refinement in diffusion models at test-time with applications to protein and DNA design. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=9qzpNSTUYp.

[69] Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, 37(2), April 2009. ISSN 0090-5364. doi: 10.1214/07-aos574. URL http://dx.doi.org/10.1214/07-AOS574.

Appendix

A Experimental details **18**

A.1 Protein design tasks 18

A.2 Out-of-distribution robustness 18

A.3 Surrogate training 19

A.4 Baselines implementation 19

A.5 Evaluation metrics 20

A.6 Evodiff 20

B Discussion **21**

B.1 Limitations 21

B.2 Future work 21

B.3 Broader impact 21

C Background **21**

D Full results **22**

D.1 Protein design 22

D.2 Early-round protein design 23

D.3 Out-of-distribution robustness 24

E Extended related work **24**

F Further ablations **25**

G Algorithms **26**

A Experimental details

A.1 Protein design tasks

We evaluated PROSPERO across eight diverse protein fitness landscapes. Among these, AAV and GFP were created by Kim et al. [12] (Apache-2.0 license), while the remaining datasets were collected by Tran and Hy [13] (GPL-3.0 license). For AAV and GFP, we used oracles available in FLEXS [7] (Apache-2.0 license), while for the remaining datasets oracles provided by Ren et al. [6] (Apache-2.0 license).

(i) **Aliphatic Amide Hydrolase (AMIE).** The objective is to optimize amidase sequences for high enzymatic activity [44]. The initial dataset $\mathcal{D}_0$ includes 6417 sequences of length $L = 341$, making the search space span across 20^{341} possible variants. The fitness of the starting sequence is $f(x_{\text{start}}) = 0.224$, while the average distance between the starting sequence and $\mathcal{D}_0$ is $\text{Novelty}(\mathcal{D}_0, x_{\text{start}}) = 2$.

(ii) **TEM-1 β-Lactamase (TEM).** The goal is to identify TEM-1 β-lactamase variants with improved thermodynamic stability [45]. $\mathcal{D}_0$ consists of 5199 sequences with $L = 286$, $f(x_{\text{start}}) = 1.229$, and $\text{Novelty}(\mathcal{D}_0, x_{\text{start}}) = 2$.

(iii) **Ubiquitination Factor Ube4b (E4B).** The goal is to enhance the activity of the E4B ubiquitination enzyme [46]. The dataset consists of 91,032 sequences with $L = 102$, from which we randomly select 10,000 for the initial dataset $\mathcal{D}_0$. The fitness of the starting sequence $f(x_{\text{start}}) = 7.743$, with $\text{Novelty}(\mathcal{D}_0, x_{\text{start}}) = 5.42$.

(iv) **Poly(A)-binding Protein (Pab1).** The aim is to improve the binding fitness of Pab1 variants in the RNA recognition motif region [47]. The dataset contains 36,389 mutants of length $L = 75$. Similarly to E4B, we restrict $\mathcal{D}_0$ to 10,000 randomly selected sequences. The fitness of the starting sequence $f(x_{\text{start}}) = 0.843$, with $\text{Novelty}(\mathcal{D}_0, x_{\text{start}}) = 3.95$.

(v) **Adeno-associated Viruses (AAV).** The objective is to discover VP1 protein sequence fragments(positions 450–540) with improved gene therapy efficiency [48]. The dataset $\mathcal{D}_0$, of size 15,307, was created by Kim et al. [12] through random mutations of the wild-type and scoring with the oracle. Here, $L = 90$, $f(x_{\text{start}}) = 0.500$, and $\text{Novelty}(\mathcal{D}_0, x_{\text{start}}) = 5.05$.

(vi) **Green Fluorescent Proteins (GFP).** The goal is to identify protein sequences with high log-fluorescence intensity [49]. $\mathcal{D}_0$ includes 10200 sequences mutated by Kim et al. [12], as with AAV. In this case, $L = 238$, $f(x_{\text{start}}) = 3.572$, and $\text{Novelty}(\mathcal{D}_0, x_{\text{start}}) = 42.87$.

(vii) **SUMO E2 conjugase (UBE2I).** The aim here is to optimize variants of SUMO E2 conjugase for functional mapping applications [50]. $\mathcal{D}_0$ consists of 3022 sequences with $L = 159$, $f(x_{\text{start}}) = 2.978$, and $\text{Novelty}(\mathcal{D}_0, x_{\text{start}}) = 2$.

(viii) **Levoglucosan Kinase (LGK).** The objective is to optimize levoglucosan kinase variants for improved enzymatic activity [51]. $\mathcal{D}_0$ contains 7633 sequences with $L = 439$, $f(x_{\text{start}}) = 0.020$, and $\text{Novelty}(\mathcal{D}_0, x_{\text{start}}) = 2$.

A.2 Out-of-distribution robustness

In this experiment, we used candidate sequences generated by different exploration methods during UBE2I optimization task in Section 5.1. We employed ESMFold [19] as the oracle to predict pTM scores, as protein structure prediction models provide highly reliable feedback, better suited for assessing performance in challenging OOD settings [52]. From all UBE2I candidates, we selected those within a Hamming distance ≤ 5 from the wild-type to construct the initial dataset $\mathcal{D}_0$, consisting of 2624 sequence–pTM pairs, and with the average pTM score of 0.860 ± 0.029. The surrogate model f_θ was trained on $\mathcal{D}_0$, and optimization was performed for 4 iterations starting from sequences increasingly distant from the wild-type. We considered three cases:

(i) **Moderate covariate shift.** Optimization was initiated from a starting sequence x_{start} differing by 35 mutations from the wild-type, and with an initial pTM of approximately 0.70.

(ii) **Severe covariate shift.** Optimization was initiated from a starting sequence x_{start} differing by 75 mutations from the wild-type, and with an initial pTM of approximately 0.50.

(iii) **Low-data regime shift.** The moderate shift experiment was repeated with the surrogate trained on only 200 randomly selected points from $\mathcal{D}_0$, resulting in a reduced training set with an average pTM score of 0.860 ± 0.023.

Across all three cases, the exploration algorithms were run with their configurations corresponding to shorter sequences, as detailed in Section 5 for PROSPERO and in Appendix A.4 for the baselines.

A.3 Surrogate training

Following Kim et al. [12], we trained surrogate models using the Adam optimizer [53], with both the learning rate and L2 penalty set to 0.0001, and a batch size of 256. The maximum number of proxy updates was set to 3000, but we employed early stopping with 10% of the dataset reserved for validation, terminating training if the validation loss failed to improve for 10 consecutive iterations.

A.4 Baselines implementation

For the following baselines, we employed the open-source implementations provided by the FLEXS benchmark [7], available at `https://github.com/samsinai/FLEXS/tree/master` under the Apache-2.0 license.

(i) **AdaLead** [7]: We followed the default hyperparameter settings provided by the authors. Specifically, we used a recombination rate of 0.2, a mutation rate of $1/L$, where L is the sequence length, and a threshold $\tau = 0.05$.

(ii) **DyNaPPO** [8]: We altered the implementation of DyNaPPO from FLEXS following the approach of Kim et al. [12]. Specifically, we replaced originally proposed proxy architectures with the same one-dimensional CNN ensembles used for other methods.

(iii) **CbAS** [14]: We implemented CbAS using a VAE [30] as the generator, retraining it at each cycle using top 20% of sequences weighted by the density ratio between the ground-truth-conditioned distribution and current sampling distribution.

(iv) **BO** [7]: We select starting sequences via Thompson sampling and use UCB to select local mutations based on surrogate model's predicted mean and uncertainty.

(v) **CMA-ES** [16]: Following Sinai et al. [7], we convert the continuous outputs from CMA-ES to one-hot representations by taking the argmax at each sequence position.

We adapted the implementation of **MLDE** [13] from `https://github.com/HySonLab/Directed_Evolution` under the GPL-3.0 license. We run 10 surrogate-based optimization steps with a population size of 128 and a beam size of 4. The random-to-importance masking ratio was set to 0.6:0.4, and we used ESM-2 [19] with 35 million parameters for unmasking.

For comparisons with **LatProtRL** [10], we used the code available at `https://github.com/haewonc/LatProtRL` under the MIT license. We employed a pre-trained ESM-2 [19] for both the encoder and decoder components of VED. For tasks involving shorter sequences (AAV, E4B, Pab1) we set: (i) VED latent dimension $R = 16$; (ii) action perturbation magnitude $\delta = 0.1$; (iii) episode length $T_{\text{ep}} = 4$; (iv) constrained decoding term $m_{\text{decode}} = 8$. For tasks involving longer sequences (GFP, AMIE, TEM, UBE2I, LGK), we used: (i) $R = 32$; (ii) $\delta = 0.3$; (iii) $T_{\text{ep}} = 6$; (iv) $m_{\text{decode}} = 18$.

In our GFlowNet setup, we employed the implementation available at `https://github.com/hyeonahkimm/delta_cs` under the Apache-2.0 license [12]. For the conservative strategy **GFN-AL-δCS** proposed by Kim et al. [12], we used an adaptive δ with maximum masking radius set to 0.05 and rank-based proxy training with reweighting factor $k = 0.01$. For AAV, E4B and Pab1 we set the scaling factor $\lambda = 0.1$, whereas for GFP, AMIE, TEM, UBE2I and LGK $\lambda = 1$. For comparisons with **GFN-AL** [9] we modified the above configuration by removing rank-based proxy training and using a fixed masking radius of 1.

We implemented **PEX** [6] using the code in `https://github.com/HeliXonProtein/proximal-exploration/tree/main` under the Apache-2.0 license, with the default setting of 2 random mutations and a frontier neighbor size of 5.

A.5 Evaluation metrics

Protein fitness optimization metrics Let $\mathcal{D}_{\text{best}} = \{(x^{(i)}, f(x^{(i)}))\}_{i=1}^{100}$ denote the set of the 100 highest-ranking sequence-fitness pairs generated across N active learning rounds. The evaluation of exploration algorithms in our experiments was based on the following metrics:

(i) **Maximum fitness.** The primary evaluation criterion, representing the ability of an exploration algorithm to recover highly functional protein sequences:

$$\text{MaxFitness}(\mathcal{D}_{\text{best}}) = \max_{x \in \mathcal{D}_{\text{best}}} f(x). \tag{5}$$

(ii) **Mean fitness.** The mean fitness values of the top 100 candidate sequences:

$$\text{MeanFitness}(\mathcal{D}_{\text{best}}) = \frac{1}{|\mathcal{D}_{\text{best}}|} \sum_{x \in \mathcal{D}_{\text{best}}} f(x). \tag{6}$$

(iii) **Novelty.** The average Hamming distance between top 100 candidates and a starting sequence, characterizing the extent of divergence from the wild-type protein:

$$\text{Novelty}(\mathcal{D}_{\text{best}}, x_{\text{start}}) = \frac{1}{|\mathcal{D}_{\text{best}}|} \sum_{x \in \mathcal{D}_{\text{best}}} d(x, x_{\text{start}}). \tag{7}$$

(iv) **Diversity.** Defined as the mean pairwise Hamming distance between the top 100 candidate sequences, reflecting the exploration algorithm's ability to explore diverse regions of the fitness landscape:

$$\text{Diversity}(\mathcal{D}_{\text{best}}) = \frac{1}{|\mathcal{D}_{\text{best}}|(|\mathcal{D}_{\text{best}}| - 1)} \sum_{\substack{x, x' \in \mathcal{D}_{\text{best}} \\ x \neq x'}} d(x, x'), \tag{8}$$

Biological plausibility measures To evaluate biological plausibility of candidate sequences generated by various exploration algorithms we used the following measures:

(i) **Validity.** Defined following Surana et al. [11] as a diagnostic measure to assess whether high fitness scores correspond to biologically plausible sequences. Specifically, it checks whether physicochemical properties of the top 100 candidates all fall within the 0.5th to 99.5th percentile range of the reference property distribution. This provides high-confidence indication of whether elevated fitness scores reflect genuine biological plausibility or rather result from surrogate and oracle misspecification. The considered properties were:

- molecular weight
- aromaticity
- isoelectric point
- grand average of hydropathy (GRAVY)
- instability index

(ii) **pLDDT** and **pTM**. Both pLDDT and pTM are structure confidence scores predicted by ESMFold, scaled between 0 and 100 [19]. pLDDT measures the local per-residue confidence in structural accuracy, while pTM reflects the predicted global topology confidence. In both cases, values greater than 70 are indicative of high-confidence predictions.

(iii) **self-consistency Perplexity (scPerplexity)**. scPerplexity [38] quantifies how well a generated sequence can be recovered from its predicted structure. Specifically, it is defined as the negative log-likelihood of the original sequence conditioned on the structure predicted by a folding model. Lower scPerplexity values indicate that the sequence is more plausible under the inverse folding model given its predicted structure. For sequence folding we used ESMFold [19]; for inverse folding we used ESM-IF1 [43]

A.6 Evodiff

To model the prior over protein sequences in PROSPERO we used EvoDiff-OADM with 38 million parameters, introduced by Alamdari et al. [38] and available under the MIT license.

B Discussion

B.1 Limitations

PROSPERO utilizes EvoDiff as its backbone, a model built upon a ByteNet-style CNN architecture [54]. Without enforcing reproducibility the approach remains computationally efficient and exhibits reasonable runtime. However, ensuring deterministic runs with CNNs typically leads to substantially longer runtimes, representing a practical limitation. Specifically, we conducted all experiments on a NVIDIA Tesla V100 32GB GPU, with the total runtime across all tasks being approximately 3 hours under non-deterministic setting and around 30 hours when enforcing reproducibility. For the noise robustness study in Section 5.4, the total runtime across all signal-to-noise ratio levels took approximately 30 minutes under non-deterministic configuration and around 12 hours in the reproducible setting. We note, however, that (i) reproducibility is not a strict requirement for practitioners, and (ii) even under deterministic configuration, the computational cost remains negligible compared to the burden of wet-lab experiments—which PROSPERO is designed to help alleviate.

PROSPERO performs well in maintaining diversity of generated sequences, as shown in Section 5.1. However, approximate inference using Sequential Monte Carlo carries an inherent risk of reduced diversity, which could potentially arise depending on design choices and remains a possible limitation of the approach.

B.2 Future work

An interesting direction for future work is the development of adaptive strategies for reducing the amino acid alphabet. While our charge-based grouping offers broad applicability, more tailored schemes could further enhance performance on specific proteins. Another promising extension would be to apply PROSPERO in a lab-in-the-loop setting, using experimental validation as the oracle and integrating structure-based alanine scanning into targeted masking to more effectively identify critical residues.

B.3 Broader impact

PROSPERO can advance protein engineering for therapeutics, enzymes, and sustainable materials by improving data efficiency and reducing experimental burden associated with wet-lab screening. However, as with any general-purpose protein design tool, there is a risk of misuse for designing harmful proteins or contributing to biosecurity concerns.

C Background

Sequential Monte Carlo (SMC). Sequential Monte Carlo (SMC) is a class of approximate inference methods for sampling from complex, high-dimensional target distributions $\gamma(x_{1:T})$, where $x_{1:T}$ denotes a sequence of variables or partial states [55–57]. Rather than directly sampling from $\gamma(x_{1:T})$, SMC simplifies the inference by constructing a sequence of unnormalized intermediate target distributions $\{\tilde{\gamma}_t(x_{1:t})\}_{t=1}^T$, that progressively approximate the target. At each step t, a collection of weighted samples (i.e., particles) is propagated by sampling from proposal distributions $\{q_t(x_t \mid x_{1:t-1})\}_{t=2}^T$, and corrected using importance weights $\{w_t(x_{1:t})\}_{t=1}^T$, to account for discrepancies between the proposal and the intermediate target. At the initial step $t = 1$, N particles are sampled independently from the proposal distribution $q_1(x_1)$, and initial importance weights $w_1^{(n)}$ are assigned for each particle n:

$$x_1^{(n)} \sim q_1(x_1), \quad w_1^{(n)} = \frac{\tilde{\gamma}_1(x_1^{(n)})}{q_1(x_1^{(n)})}. \tag{9}$$

Subsequently, each step $t = 2, \ldots, T$ consists of the following three operations [56]:

(i) Optional resampling:

$$x_{1:t-1}^{(n)} \sim \mathrm{Cat}\left(\left\{ x_{1:t-1}^{(n)} \right\}_{n=1}^N, \left\{ \frac{w_{t-1}^{(n)}}{\sum_{m=1}^N w_{t-1}^{(m)}} \right\}_{i=1}^N \right). \tag{10}$$

(ii) Proposing:

$$x_t^{(n)} \sim q_t(x_t^{(n)} \mid x_{1:t-1}^{(n)}). \tag{11}$$

(iii) Weighting:

$$w_t^{(n)} = \frac{\tilde{\gamma}_t(x_{1:t}^{(n)})}{\tilde{\gamma}_{t-1}(x_{1:t-1}^{(n)}) q(x_t^{(n)} \mid x_{1:t-1}^{(n)})} \tag{12}$$

D Full results

D.1 Protein design

Table 5: Mean fitness of top 100 sequences generated by each method. Reported values are the mean and standard deviation over 5 runs.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
CMA-ES	-8.317 ± 0.029	0.013 ± 0.000	-1.009 ± 0.029	0.232 ± 0.012	0.000 ± 0.000	1.593 ± 0.008	-0.072 ± 0.004	-1.538 ± 0.008
DynaPPO	-6.493 ± 0.155	0.027 ± 0.002	0.574 ± 0.148	0.481 ± 0.013	0.000 ± 0.000	2.064 ± 0.068	1.600 ± 0.101	-1.020 ± 0.045
BO	-0.849 ± 0.474	0.606 ± 0.352	5.909 ± 0.785	0.510 ± 0.047	0.618 ± 0.010	3.538 ± 0.036	2.695 ± 0.148	-0.017 ± 0.020
PEX	0.238 ± 0.004	1.227 ± 0.002	7.948 ± 0.046	1.307 ± 0.258	0.620 ± 0.017	3.597 ± 0.003	2.987 ± 0.001	0.033 ± 0.001
AdaLead	0.229 ± 0.001	1.201 ± 0.002	7.846 ± 0.040	1.836 ± 0.266	0.644 ± 0.031	3.563 ± 0.007	2.976 ± 0.003	0.037 ± 0.001
CbAS	-8.361 ± 0.025	0.010 ± 0.001	-0.820 ± 0.068	0.162 ± 0.082	0.000 ± 0.000	1.666 ± 0.021	-0.072 ± 0.003	-1.659 ± 0.023
GFN-AL	-8.268 ± 0.010	0.015 ± 0.001	-0.415 ± 0.091	0.276 ± 0.036	0.000 ± 0.000	1.776 ± 0.009	0.172 ± 0.396	-1.345 ± 0.037
GFN-AL-δCS	-0.244 ± 0.137	0.192 ± 0.027	7.653 ± 0.136	1.070 ± 0.113	0.648 ± 0.020	3.569 ± 0.009	2.968 ± 0.006	0.024 ± 0.004
LatProtRL	0.217 ± 0.001	1.222 ± 0.000	7.562 ± 0.06	0.888 ± 0.072	0.563 ± 0.009	3.582 ± 0.003	2.975 ± 0.001	0.019 ± 0.000
MLDE	0.231 ± 0.004	1.131 ± 0.021	7.843 ± 0.122	0.877 ± 0.024	0.555 ± 0.000	3.591 ± 0.003	2.975 ± 0.005	0.036 ± 0.002
ProSpero	0.236 ± 0.007	1.176 ± 0.029	8.017 ± 0.054	1.401 ± 0.202	0.679 ± 0.025	3.613 ± 0.002	2.987 ± 0.003	0.040 ± 0.002

Table 6: Median fitness of top 100 sequences generated by each method. Reported values are the mean and standard deviation over 5 runs.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
CMA-ES	-8.392 ± 0.006	0.011 ± 0.000	-1.046 ± 0.027	0.206 ± 0.013	0.000 ± 0.000	1.578 ± 0.006	-0.079 ± 0.002	-1.563 ± 0.008
DyNaPPO	-6.731 ± 0.125	0.025 ± 0.001	0.333 ± 0.167	0.465 ± 0.012	0.000 ± 0.000	1.769 ± 0.033	1.577 ± 0.108	-1.198 ± 0.032
BO	-0.919 ± 0.591	0.600 ± 0.352	5.779 ± 0.851	0.489 ± 0.043	0.613 ± 0.009	3.543 ± 0.034	2.681 ± 0.169	-0.009 ± 0.028
PEX	0.237 ± 0.004	1.228 ± 0.000	7.937 ± 0.050	1.298 ± 0.257	0.616 ± 0.017	3.597 ± 0.003	2.986 ± 0.001	0.033 ± 0.001
AdaLead	0.228 ± 0.001	1.198 ± 0.001	7.832 ± 0.042	1.830 ± 0.271	0.641 ± 0.031	3.561 ± 0.007	2.976 ± 0.003	0.037 ± 0.001
CbAS	-8.371 ± 0.023	0.009 ± 0.001	-0.841 ± 0.065	0.152 ± 0.084	0.000 ± 0.000	1.655 ± 0.021	-0.073 ± 0.004	-1.670 ± 0.024
GFN-AL	-8.287 ± 0.007	0.014 ± 0.001	-0.458 ± 0.083	0.257 ± 0.043	0.000 ± 0.000	1.758 ± 0.009	0.169 ± 0.396	-1.353 ± 0.045
GFN-AL-δCS	-0.184 ± 0.150	0.145 ± 0.026	7.633 ± 0.143	1.064 ± 0.113	0.645 ± 0.020	3.568 ± 0.009	2.967 ± 0.006	0.024 ± 0.005
LatProtRL	0.218 ± 0.001	1.222 ± 0.000	7.523 ± 0.062	0.876 ± 0.070	0.560 ± 0.010	3.582 ± 0.003	2.975 ± 0.001	0.018 ± 0.000
MLDE	0.231 ± 0.004	1.117 ± 0.038	7.834 ± 0.130	0.874 ± 0.027	0.555 ± 0.000	3.591 ± 0.003	2.975 ± 0.006	0.036 ± 0.002
ProSpero	0.235 ± 0.007	1.187 ± 0.041	8.013 ± 0.055	1.392 ± 0.200	0.676 ± 0.024	3.613 ± 0.002	2.987 ± 0.003	0.040 ± 0.002

Table 7: Average diversity between top 100 sequences generated by each method. Reported values are the mean and standard deviation over 5 runs.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
CMA-ES	247.97 ± 1.96	203.40 ± 4.60	75.23 ± 0.83	49.51 ± 1.67	60.34 ± 0.89	163.73 ± 2.44	108.01 ± 1.72	316.96 ± 5.31
DynaPPO	116.31 ± 0.82	98.68 ± 1.83	28.14 ± 0.31	22.00 ± 0.83	27.96 ± 0.44	79.06 ± 1.63	49.53 ± 0.81	157.11 ± 7.56
BO	21.46 ± 3.10	29.16 ± 24.73	15.65 ± 3.80	35.82 ± 4.80	6.92 ± 0.29	34.92 ± 3.13	32.33 ± 7.00	76.97 ± 57.70
PEX	7.06 ± 0.61	3.35 ± 0.32	4.65 ± 0.36	4.88 ± 0.87	7.00 ± 1.05	6.66 ± 0.86	6.24 ± 0.69	8.17 ± 1.47
AdaLead	6.37 ± 0.93	3.96 ± 0.06	5.82 ± 0.41	3.83 ± 0.72	8.09 ± 2.81	26.58 ± 12.68	6.04 ± 0.62	6.59 ± 1.17
CbAS	236.10 ± 25.20	232.03 ± 5.38	76.31 ± 8.05	53.47 ± 4.69	54.21 ± 5.59	145.39 ± 48.19	125.98 ± 6.49	349.88 ± 33.82
GFN-AL	323.77 ± 0.07	270.79 ± 0.25	96.13 ± 0.66	70.61 ± 0.16	79.70 ± 7.08	225.07 ± 0.12	78.26 ± 72.16	423.68 ± 0.70
GFN-AL-δCS	14.68 ± 0.79	17.97 ± 1.27	5.55 ± 0.96	5.68 ± 1.60	8.33 ± 1.14	24.20 ± 14.47	8.12 ± 3.73	65.68 ± 10.56
LatProtRL	1.10 ± 0.07	1.27 ± 0.01	4.02 ± 0.41	3.84 ± 0.34	3.29 ± 0.29	19.97 ± 3.07	2.24 ± 0.09	1.70 ± 0.00
MLDE	10.10 ± 1.23	6.87 ± 0.96	4.65 ± 1.24	2.35 ± 1.40	1.71 ± 0.22	8.80 ± 1.62	8.86 ± 1.61	7.52 ± 2.06
ProSpero	12.20 ± 1.15	5.10 ± 0.58	4.12 ± 0.28	3.55 ± 0.26	5.11 ± 0.58	9.90 ± 2.01	9.54 ± 2.35	17.25 ± 5.64

Table 8: Average novelty of top 100 sequences generated by each method. Reported values are the mean and standard deviation over 5 runs.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
CMA-ES	169.18 ± 1.96	136.79 ± 4.91	51.75 ± 0.96	32.37 ± 1.81	40.17 ± 0.98	107.96 ± 2.41	71.26 ± 1.62	212.41 ± 5.63
DynaPPO	64.78 ± 0.55	54.97 ± 1.13	15.40 ± 0.20	12.13 ± 0.48	15.40 ± 0.28	43.88 ± 0.99	27.39 ± 0.51	91.65 ± 12.56
BO	19.27 ± 1.94	35.37 ± 36.86	14.69 ± 2.38	48.94 ± 16.18	7.19 ± 0.68	41.55 ± 9.56	35.58 ± 12.95	90.62 ± 62.22
PEX	5.19 ± 1.08	1.79 ± 0.21	3.96 ± 0.54	5.29 ± 0.74	7.29 ± 1.38	6.23 ± 1.34	4.55 ± 0.82	8.45 ± 1.39
AdaLead	4.21 ± 0.89	2.93 ± 0.07	3.92 ± 0.62	8.47 ± 1.58	9.86 ± 1.55	33.79 ± 5.84	3.92 ± 0.47	14.75 ± 8.02
CbAS	323.72 ± 1.16	271.22 ± 1.43	96.41 ± 0.98	71.56 ± 1.08	86.56 ± 0.48	225.15 ± 1.94	150.19 ± 0.36	423.30 ± 1.96
GFN-AL	323.19 ± 0.28	272.81 ± 0.22	97.49 ± 0.28	71.40 ± 0.04	84.06 ± 1.17	226.60 ± 0.16	152.03 ± 0.60	425.16 ± 0.62
GFN-AL-δCS	8.29 ± 0.42	10.10 ± 0.65	4.94 ± 1.59	10.29 ± 1.29	9.70 ± 0.33	34.83 ± 3.77	8.01 ± 3.69	63.16 ± 4.24
LatProtRL	1.09 ± 0.04	1.10 ± 0.01	3.11 ± 0.49	3.85 ± 1.19	3.03 ± 0.40	12.73 ± 2.21	1.69 ± 0.05	1.71 ± 0.01
MLDE	19.02 ± 2.69	4.28 ± 0.55	11.88 ± 3.49	9.67 ± 3.12	4.48 ± 0.25	23.65 ± 5.40	9.60 ± 2.58	50.09 ± 8.08
PROSPERO	20.99 ± 3.32	3.37 ± 0.53	8.81 ± 0.98	11.83 ± 3.52	15.03 ± 1.59	39.85 ± 3.95	16.45 ± 5.11	74.33 ± 7.75

Table 9: Average diversity between top 100 sequences generated by leading methods. Reported values are the mean and standard deviation over 5 runs. **Bold:** the best overall diversity. Underline: second-best. PROSPERO maintains a viable level of sequence diversity.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
PEX	7.06 ± 0.61	3.35 ± 0.32	4.65 ± 0.36	4.88 ± 0.87	7.00 ± 1.05	6.66 ± 0.86	6.24 ± 0.69	8.17 ± 1.47
AdaLead	6.37 ± 0.93	3.96 ± 0.06	**5.82 ± 0.41**	3.83 ± 0.72	8.09 ± 2.81	**26.58 ± 12.68**	6.04 ± 0.62	6.59 ± 1.17
GFN-AL-δCS	**14.68 ± 0.79**	**17.97 ± 1.27**	5.55 ± 0.96	**5.68 ± 1.60**	**8.33 ± 1.14**	24.20 ± 14.47	8.12 ± 3.73	**65.68 ± 10.56**
LatProtRL	1.10 ± 0.07	1.27 ± 0.01	4.02 ± 0.41	3.84 ± 0.34	3.29 ± 0.29	19.97 ± 3.07	2.24 ± 0.09	1.70 ± 0.00
MLDE	10.10 ± 1.23	6.87 ± 0.96	4.65 ± 1.24	2.35 ± 1.40	1.71 ± 0.22	8.80 ± 1.62	8.86 ± 1.61	7.52 ± 2.06
PROSPERO	12.20 ± 1.15	5.10 ± 0.58	4.12 ± 0.28	3.55 ± 0.26	5.11 ± 0.58	9.90 ± 2.01	**9.54 ± 2.35**	17.25 ± 5.64

D.2 Early-round protein design

Table 10: Maximum fitness of top 100 sequences generated by leading methods limited to 4 rounds. Reported values are the mean and standard deviation over 5 runs. **Bold:** the best overall fitness. Underline: second-best.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
PEX	$\mathbf{0.242 \pm 0.001}$	$\mathbf{1.231 \pm 0.001}$	7.971 ± 0.078	1.064 ± 0.071	0.604 ± 0.018	3.597 ± 0.002	2.987 ± 0.002	0.030 ± 0.001
AdaLead	0.232 ± 0.003	1.227 ± 0.004	7.962 ± 0.071	$\mathbf{1.397 \pm 0.329}$	0.596 ± 0.014	3.580 ± 0.003	2.982 ± 0.002	0.032 ± 0.002
GFN-AL-δCS	0.160 ± 0.048	0.563 ± 0.119	7.859 ± 0.047	1.035 ± 0.094	0.596 ± 0.010	3.584 ± 0.005	2.972 ± 0.013	0.032 ± 0.002
LatProtRL	0.224 ± 0.000	1.229 ± 0.000	7.751 ± 0.016	1.031 ± 0.129	0.565 ± 0.011	3.589 ± 0.003	2.982 ± 0.001	0.020 ± 0.000
MLDE	0.231 ± 0.006	1.229 ± 0.000	7.821 ± 0.063	0.866 ± 0.024	0.555 ± 0.000	3.589 ± 0.003	2.978 ± 0.000	0.028 ± 0.003
PROSPERO	0.236 ± 0.007	1.229 ± 0.001	$\mathbf{7.978 \pm 0.055}$	1.202 ± 0.129	$\mathbf{0.635 \pm 0.019}$	$\mathbf{3.602 \pm 0.002}$	$\mathbf{2.989 \pm 0.002}$	$\mathbf{0.036 \pm 0.001}$

Table 11: Mean fitness of top 100 sequences generated by leading methods limited to 4 rounds. Reported values are the mean and standard deviation over 5 runs. **Bold:** the best overall fitness. Underline: second-best.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
PEX	$\mathbf{0.230 \pm 0.001}$	1.176 ± 0.008	7.686 ± 0.035	0.891 ± 0.036	0.553 ± 0.012	3.589 ± 0.002	$\mathbf{2.982 \pm 0.001}$	0.026 ± 0.001
AdaLead	0.224 ± 0.002	1.185 ± 0.011	7.682 ± 0.038	$\mathbf{1.118 \pm 0.222}$	0.554 ± 0.010	3.557 ± 0.007	2.964 ± 0.007	0.029 ± 0.002
GFN-AL-δCS	-0.936 ± 0.411	0.108 ± 0.017	7.271 ± 0.232	0.865 ± 0.065	0.549 ± 0.006	3.562 ± 0.007	2.772 ± 0.141	0.017 ± 0.005
LatProtRL	0.200 ± 0.001	$\mathbf{1.213 \pm 0.000}$	6.794 ± 0.125	0.743 ± 0.044	0.525 ± 0.004	3.571 ± 0.005	2.960 ± 0.006	0.018 ± 0.000
MLDE	0.212 ± 0.011	1.060 ± 0.024	7.538 ± 0.109	0.786 ± 0.059	0.551 ± 0.000	3.585 ± 0.003	2.911 ± 0.019	0.025 ± 0.004
PROSPERO	0.221 ± 0.010	1.014 ± 0.104	$\mathbf{7.781 \pm 0.048}$	1.009 ± 0.072	$\mathbf{0.576 \pm 0.017}$	$\mathbf{3.595 \pm 0.002}$	2.976 ± 0.006	$\mathbf{0.033 \pm 0.002}$

Table 12: Average diversity between top 100 sequences generated by leading methods limited to 4 rounds. Reported values are the mean and standard deviation over 5 runs. **Bold:** the best overall diversity. Underline: second-best.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
PEX	5.02 ± 0.50	3.07 ± 0.22	3.55 ± 0.36	3.93 ± 0.46	5.27 ± 0.76	5.85 ± 0.99	4.61 ± 0.63	7.25 ± 1.29
AdaLead	4.34 ± 0.34	4.04 ± 0.07	4.19 ± 0.26	3.12 ± 0.50	$\mathbf{7.00 \pm 0.43}$	$\mathbf{32.04 \pm 10.76}$	5.38 ± 0.77	4.36 ± 1.29
GFN-AL-δCS	$\mathbf{21.22 \pm 5.12}$	$\mathbf{20.39 \pm 0.75}$	5.21 ± 1.47	$\mathbf{5.49 \pm 0.85}$	6.53 ± 0.52	21.13 ± 13.93	$\mathbf{13.14 \pm 3.75}$	$\mathbf{62.52 \pm 9.11}$
LatProtRL	1.60 ± 0.04	1.81 ± 0.01	4.24 ± 0.14	4.02 ± 0.55	3.34 ± 0.12	27.93 ± 6.33	2.32 ± 0.01	1.79 ± 0.00
MLDE	10.99 ± 1.45	6.93 ± 0.62	$\mathbf{5.53 \pm 0.75}$	3.98 ± 0.98	1.63 ± 0.40	8.06 ± 1.27	7.36 ± 1.88	8.29 ± 1.82
PROSPERO	11.84 ± 1.82	6.09 ± 1.19	4.11 ± 0.48	3.65 ± 0.25	4.88 ± 0.45	9.18 ± 0.88	9.08 ± 0.78	21.18 ± 2.40

Table 13: Average novelty of top 100 sequences generated by leading methods limited to 4 rounds. Reported values are the mean and standard deviation over 5 runs. **Bold:** the best overall novelty. Underline: second-best.

Method	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
PEX	2.83 ± 0.24	1.57 ± 0.12	1.97 ± 0.27	2.33 ± 0.41	3.68 ± 0.17	4.01 ± 0.78	2.67 ± 0.48	4.45 ± 1.00
AdaLead	2.77 ± 0.44	3.01 ± 0.05	2.41 ± 0.19	4.12 ± 0.89	4.82 ± 0.36	$\mathbf{34.94 \pm 4.19}$	3.41 ± 0.54	8.30 ± 8.37
GFN-AL-δCS	11.68 ± 2.71	$\mathbf{11.45 \pm 0.40}$	3.52 ± 1.71	4.99 ± 0.81	4.52 ± 0.26	30.61 ± 3.44	$\mathbf{13.14 \pm 1.98}$	$\mathbf{45.70 \pm 3.31}$
LatProtRL	1.36 ± 0.034	1.61 ± 0.01	3.19 ± 0.18	2.87 ± 0.45	1.88 ± 0.06	16.44 ± 3.56	1.75 ± 0.07	1.85 ± 0.01
MLDE	$\mathbf{11.85 \pm 2.46}$	4.71 ± 0.50	$\mathbf{7.11 \pm 1.21}$	5.76 ± 0.67	4.10 ± 0.39	13.49 ± 1.48	6.36 ± 0.67	18.90 ± 5.66
ProSpero	10.67 ± 1.62	3.40 ± 0.58	3.95 ± 0.88	$\mathbf{6.16 \pm 2.36}$	$\mathbf{6.60 \pm 0.78}$	17.16 ± 3.23	9.02 ± 3.48	37.09 ± 4.32

D.3 Out-of-distribution robustness

Table 14: Results under moderate covariate shift. Reported values are the mean and standard deviation over 5 runs. **Bold:** the best overall value. Underline: second-best.

Method	Maximum pTM	Mean pTM	Diversity	Novelty
PEX	0.807 ± 0.023	0.760 ± 0.012	6.14 ± 0.89	4.45 ± 0.38
AdaLead	0.796 ± 0.013	0.755 ± 0.011	8.83 ± 2.54	8.36 ± 2.97
GFN-AL-δCS	0.791 ± 0.010	0.729 ± 0.005	$\mathbf{16.92 \pm 0.88}$	9.56 ± 0.60
LatProtRL	0.787 ± 0.013	0.743 ± 0.003	6.32 ± 0.32	5.90 ± 0.53
MLDE	0.810 ± 0.020	0.752 ± 0.004	9.89 ± 1.11	$\mathbf{20.88 \pm 2.98}$
ProSpero	$\mathbf{0.822 \pm 0.027}$	$\mathbf{0.777 \pm 0.020}$	11.50 ± 1.62	17.74 ± 3.20

Table 15: Results under severe covariate shift. Reported values are the mean and standard deviation over 5 runs. **Bold:** the best overall value. Underline: second-best.

Method	Maximum pTM	Mean pTM	Diversity	Novelty
PEX	0.578 ± 0.014	0.518 ± 0.003	3.40 ± 0.07	1.72 ± 0.04
AdaLead	0.593 ± 0.028	0.526 ± 0.007	14.26 ± 1.91	7.66 ± 1.08
GFN-AL-δCS	0.630 ± 0.024	0.542 ± 0.006	$\mathbf{24.13 \pm 1.47}$	14.63 ± 1.16
LatProtRL	0.560 ± 0.000	0.508 ± 0.003	2.24 ± 0.14	1.78 ± 0.16
MLDE	0.652 ± 0.059	0.572 ± 0.035	13.10 ± 1.18	21.68 ± 3.85
ProSpero	$\mathbf{0.672 \pm 0.031}$	$\mathbf{0.599 \pm 0.014}$	14.51 ± 1.99	$\mathbf{22.03 \pm 1.69}$

Table 16: Results under low-data covariate shift. Reported values are the mean and standard deviation over 5 runs. **Bold:** the best overall value. Underline: second-best.

Method	Maximum pTM	Mean pTM	Diversity	Novelty
PEX	0.806 ± 0.013	0.752 ± 0.005	6.77 ± 0.45	4.39 ± 0.51
AdaLead	0.781 ± 0.016	0.742 ± 0.004	7.99 ± 1.39	5.95 ± 2.04
GFN-AL-δCS	0.782 ± 0.006	0.731 ± 0.006	$\mathbf{15.82 \pm 1.77}$	9.46 ± 1.69
LatProtRL	0.792 ± 0.013	0.743 ± 0.001	6.25 ± 0.23	5.57 ± 0.20
MLDE	0.782 ± 0.022	0.735 ± 0.025	9.39 ± 2.88	$\mathbf{16.97 \pm 3.78}$
ProSpero	$\mathbf{0.808 \pm 0.017}$	$\mathbf{0.763 \pm 0.017}$	11.25 ± 2.51	15.87 ± 1.17

E Extended related work

Inference-time guidance methods Several methods have been proposed to steer generative models during inference. Dhariwal and Nichol [58] introduced classifier guidance, where the sampling is biased toward desired properties by adjusting the generative score with the gradient of an auxiliary classifier. However, this approach is not directly applicable in the discrete data domain, where

gradients with respect to inputs are not well-defined. To address this, Nisonoff et al. [59] developed a framework that enables classifier guidance in discrete diffusion and flow matching models. Their approach leverages a continuous-time Markov chain formulation of the forward process and corresponding reverse-time generative process [60], where only one coordinate changes at each transition, making exact guidance tractable.

Guidance can also be realized through SMC-based approaches, applicable in both discrete and continuous domains [61–63]. These methods steer generation by maintaining a population of particles that represent partial trajectories and resampling them according to their likelihood under a target distribution. Ekström Kelvinius and Lindsten [64] extend discriminator guidance [65], originally developed for score-based diffusion models, to Autoregressive Diffusion Models (ARDMs) [66] (such as leveraged in our work EvoDiff-OADM [38]), and further employ SMC to correct for discriminator errors at intermediate sampling steps. Li et al. [67] take a similar approach that resembles SMC in the use of importance sampling, but instead of resampling across the entire batch of particles, they generate and reweight multiple candidates from each individual sample at the previous step. Building on this idea, Uehara et al. [68] combine it with a noising policy, iteratively alternating between re-noising and reward-guided denoising to progressively refine samples.

Similarities to CloneBO [26] PROSPERO and CloneBO by Amin et al. [26] share similarities in guiding a generative model at inference time using SMC. However, the approaches differ meaningfully in both scope and mechanism. First, in CloneBO, the generative model (CloneLM) has been trained specifically for the optimization task by fitting to a distribution of clonal families. In contrast, our method uses a general-purpose, task-agnostic pre-trained generative model, enabling effortless optimization regardless of the protein family. As for the differences in the use of SMC, in CloneBO the authors compute intermediate importance weights directly, as a likelihood ratio between the base CloneLM and a twisted variant incorporating high value sequences (i.e. sequences with experimental measurements) in the conditioned clonal family. In PROSPERO, we approximate the intermediate importance weights using the surrogate model, which serves as a tractable proxy for the true likelihood. Moreover, our biologically-constrained SMC restricts proposals to charge-compatible amino acids, making certain residues impossible to sample. In contrast, CloneBO does not enforce such constraints; although twisting reduces the probability of sampling undesirable residues, it does not eliminate them entirely.

Connections between SMC and pseudo-marginal MCMC Pseudo-marginal MCMC [69] and SMC differ in the way they approximate the target distribution. In pseudo-marginal MCMC, within a single Markov chain, a single collection of samples is generated whose marginal stationary distribution is exactly the target distribution. The approximation improves with the number of steps T and is exact in the limit of inifite T. In contrast, SMC maintains a population of N samples that evolve over a sequence of T intermediate distributions. The empirical distribution formed by these samples converges to the target distribution in the limit of infinite N.

F Further ablations

Influence of the number of starting sequences on candidate generation We investigated how varying the number of best starting sequences x_{start} affects the proposed candidates by conducting experiments on the LGK landscape, which requires generating the longest sequences ($L = 439$). The results in Table 17 demonstrate that fewer starting points drive deeper, more directed exploration, resulting in higher novelty and fitness but lower diversity. In contrast, more starting points promote broader, more diffuse exploration, increasing diversity but limiting how far any single trajectory moves from the wild-type across subsequent optimization rounds.

Influence of reducing the amino acid alphabet To further analyze the isolated effect of constraining the proposals to charge-compatible amino acids, we directly compare the performance of the full PROSPERO with its ablated counterpart lacking this restriction (without RAA). We followed the setup detailed in Section 5.5. The results in Table 18 highlight the relevance of this feature, showing consistent improvements across all signal-to-noise ratio levels.

Influence of the charge-class permutation order The sampling permutation order in biologically-constrained SMC was chosen to prioritize charge classes with fewer valid options (negatively charged

residues first, followed by positively charged, and finally neutral), as resolving the most constrained decisions early should help prevent suboptimal completions later in the sequence. To assess this, we conducted an ablation comparing PROSPERO with its standard permutation order (ascending) to both a reversed (descending) order and a random order. The results presented in Table 19 show that the reasoning behind this choice appears correct, though the performance benefits are modest.

Table 17: Results on the LGK landscape for varying numbers of starting sequences. Reported values are then mean and standard deviation over 5 runs. The best overall values are highlighted in **bold**.

x_{start}	Maximum fitness	Mean fitness	Diversity	Novelty
1	$\mathbf{0.043 \pm 0.002}$	$\mathbf{0.040 \pm 0.002}$	17.25 ± 5.64	$\mathbf{74.33 \pm 7.75}$
4	0.041 ± 0.001	0.039 ± 0.001	19.30 ± 4.58	70.20 ± 6.05
16	0.038 ± 0.002	0.037 ± 0.002	19.09 ± 5.80	64.86 ± 5.44
64	0.037 ± 0.002	0.034 ± 0.001	29.13 ± 6.97	56.38 ± 7.06
128	0.033 ± 0.002	0.030 ± 0.002	$\mathbf{33.89 \pm 9.54}$	50.45 ± 5.52

Table 18: Results on the AAV landscape across different signal-to-noise ratio levels (SNR). Reported values are then mean and standard deviation over 5 runs. The best overall values are highlighted in **bold**.

SNR level	**-25**	**-20**	**-15**	**-10**	**-5**	**0**
PROSPERO	**0.566 ± 0.030**	**0.586 ± 0.026**	**0.651 ± 0.032**	**0.679 ± 0.047**	**0.704 ± 0.016**	**0.706 ± 0.022**
PROSPERO w/o RAA	0.507 ± 0.025	0.555 ± 0.040	0.588 ± 0.029	0.605 ± 0.032	0.653 ± 0.042	0.666 ± 0.029

Table 19: Results on all the landscapes with different permutation orderings. Reported values are then mean and standard deviation over 5 runs. The best overall values are highlighted in **bold**.

Ordering	AMIE	TEM	E4B	Pab1	AAV	GFP	UBE2I	LGK
Ascending	**0.246 ± 0.006**	1.231 ± 0.002	8.114 ± 0.037	**1.527 ± 0.254**	**0.720 ± 0.027**	**3.617 ± 0.003**	**2.993 ± 0.003**	**0.043 ± 0.002**
Descending	0.244 ± 0.005	**1.232 ± 0.003**	8.139 ± 0.037	1.363 ± 0.141	0.706 ± 0.035	3.614 ± 0.003	2.991 ± 0.003	0.041 ± 0.003
Random	0.243 ± 0.005	**1.232 ± 0.002**	**8.164 ± 0.015**	1.338 ± 0.113	0.708 ± 0.029	3.614 ± 0.003	**2.993 ± 0.002**	0.042 ± 0.003

G Algorithms

Algorithm 2: Targeted Masking

Input: Starting sequence x_{start}, proxy f_θ, SMC batch size B, scans S, min substitutions $n_{\min}$, max substitutions $n_{\max}$, exploitation-exploration coefficient k

Output: Masked sequences $\{\tilde{x}^{(i)}\}_{i=1}^{B}$, substitution locations $\{\mathcal{I}^{(i)}\}_{i=1}^{B}$

for $i \leftarrow 1$ **to** $B \times S$ **do**
  $n_{\text{sub}}^{(i)} \sim \mathcal{U}_{[n_{\min}, n_{\max}]}$
  $\mathcal{I}^{(i)} \sim \text{UniformSubset}([1, |x_{\text{start}}|],\ n_{\text{sub}}^{(i)})$
  $x^{(i)} \leftarrow x_{\text{start}}$, where $x^{(i)}[j] \leftarrow \texttt{A}$ for $j \in \mathcal{I}^{(i)}$
$\mathcal{J} \leftarrow \arg\max^{B}_{x \in \{x^{(i)}\}_{i=1}^{B \times S}} \mu_\theta(x) + k \cdot \sigma_\theta(x)$
for $x^{(i)} \in \mathcal{J}$ **do**
  $\tilde{x}^{(i)} \leftarrow x^{(i)}$, where $\tilde{x}^{(i)}[j] \leftarrow$ `[MASK]` for $j \in \mathcal{I}^{(i)}$
return $\{\tilde{x}^{(i)}\}_{i=1}^{B}$, $\{\mathcal{I}^{(i)}\}_{i=1}^{B}$

Algorithm 3: ConstrainedSMC

Input: Partially masked sequences $\{\tilde{x}^{(i)}\}_{i=1}^{B}$, mask locations $\{\mathcal{I}^{(i)}\}_{i=1}^{B}$, pre-trained generative model $\mathcal{P}$, proxy f_θ, oracle budget K, exploitation-exploration coefficient k, kept rollouts threshold n_{keep}

Output: Candidate sequences $\{x^{(i)}\}_{i=1}^{K}$

RolloutBuffer $\leftarrow \{\}$
for $i \leftarrow 1$ **to** B **do**
  $\pi^{(i)} = \text{concat}(\underline{\mathcal{I}}^{(i)}, \mathcal{I}_{(-)}^{(i)}, \mathcal{I}_{(+)}^{(i)}, \mathcal{I}_{(0)}^{(i)})$
$T \leftarrow \arg\max_{\mathcal{I} \in \{\mathcal{I}^{(i)}\}_{i=1}^{B}} |\mathcal{I}|$
for $t \leftarrow 1$ **to** T **do**
  for $\tilde{x}^{(i)} \in \{\tilde{x}^{(i)}\}_{i=1}^{B}$ **do**
    if $t = 1$ **then**
      $LL^{(i)} \leftarrow 0$
    if $t \leq |\mathcal{I}^{(i)}|$ **then**
      $\tilde{x}^{(i)}_{\pi(t+|\underline{\mathcal{I}}^{(i)}|)} \sim \mathcal{P}_{\text{RAA}}(\tilde{x}^{(i)}_{\pi(t+|\underline{\mathcal{I}}^{(i)}|)} \mid \tilde{x}^{(i)}_{\pi(<t+|\underline{\mathcal{I}}^{(i)}|)})$
      $LL^{(i)} \leftarrow LL^{(i)} + \log \mathcal{P}(\tilde{x}^{(i)}_{\pi(t+|\underline{\mathcal{I}}^{(i)}|)} \mid \tilde{x}^{(i)}_{\pi(<t+|\underline{\mathcal{I}}^{(i)}|)})$
      $(x^{(i)}_{\text{unroll}}, LL^{(i)}_{\text{unroll}}) \leftarrow \textsc{Rollout}(\tilde{x}^{(i)}, \pi^{(i)}, LL^{(i)}, \mathcal{P}, s = t+1)$ `// Algorithm 4`
      $\hat{y}^{(i)} \leftarrow \mu_\theta(x^{(i)}_{\text{unroll}}) + k \cdot \sigma_\theta(x^{(i)}_{\text{unroll}})$
      $\text{invPPL}^{(i)} \leftarrow \exp\left(\frac{LL^{(i)}_{\text{unroll}}}{|\mathcal{I}^{(i)}|}\right)$
      if $T - t < n_{keep}$ **then**
        RolloutBuffer $\leftarrow$ RolloutBuffer $\cup \{(x^{(i)}_{\text{unroll}}, \hat{y}^{(i)})\}$
  $w \leftarrow \left(\frac{\hat{y}^{(i)} \cdot \text{invPPL}^{(i)}}{\sum_{j=1}^{B} \hat{y}_j \cdot \text{invPPL}_j}\right)_{i=1}^{B}$
  for $i \leftarrow 1$ **to** B **do**
    Resample:
    $idx^{(i)} \sim \text{Cat}(w)$
    $\tilde{x}^{(i)} \leftarrow \tilde{x}[idx^{(i)}]$
    $\pi^{(i)} \leftarrow \pi[idx^{(i)}]$
    $LL^{(i)} \leftarrow LL[idx^{(i)}]$
return $\{x^{(i)}\}_{i=1}^{K} \leftarrow \arg\max^{K}_{(x,\hat{y}) \in RolloutBuffer} \hat{y}$

Algorithm 4: Rollout

Input: Partially masked sequences $\{\tilde{x}^{(i)}\}_{i=1}^{B}$, sampling permutations $\{\pi^{(i)}\}_{i=1}^{B}$, log-likelihoods $\{LL^{(i)}\}_{i=1}^{B}$, pre-trained generative model $\mathcal{P}$, unmasking step s

Output: Unrolled sequences $\{x^{(i)}_{\text{unroll}}\}_{i=1}^{B}$, log-likelihoods $\{LL^{(i)}_{\text{unroll}}\}_{i=1}^{B}$

$T \leftarrow \arg\max_{\mathcal{I} \in \{\mathcal{I}^{(i)}\}_{i=1}^{B}} |\mathcal{I}|$
for $t \leftarrow s$ **to** T **do**
  for $\tilde{x}^{(i)} \in \{\tilde{x}^{(i)}\}_{i=1}^{B}$ **do**
    if $t \leq |\mathcal{I}^{(i)}|$ **then**
      $\tilde{x}^{(i)}_{\pi(t+|\underline{\mathcal{I}}^{(i)}|)} \sim \mathcal{P}_{\text{RAA}}(\tilde{x}^{(i)}_{\pi(t+|\underline{\mathcal{I}}^{(i)}|)} \mid \tilde{x}^{(i)}_{\pi(<t+|\underline{\mathcal{I}}^{(i)}|)})$
      $LL^{(i)} \leftarrow LL^{(i)} + \log \mathcal{P}(\tilde{x}^{(i)}_{\pi(t+|\underline{\mathcal{I}}^{(i)}|)} \mid \tilde{x}^{(i)}_{\pi(<t+|\underline{\mathcal{I}}^{(i)}|)})$
$\{x^{(i)}_{\text{unroll}}\}_{i=1}^{B} \leftarrow \{\tilde{x}^{(i)}\}_{i=1}^{B}$
$\{LL^{(i)}_{\text{unroll}}\}_{i=1}^{B} \leftarrow \{LL^{(i)}\}_{i=1}^{B}$
return $\{x^{(i)}_{unroll}\}_{i=1}^{B}, \{LL^{(i)}_{unroll}\}_{i=1}^{B}$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction reflect the paper's contributions and scope. Section 5.1 presents our method's protein fitness optimization capabilities; Section 5.2 demonstrates the biological plausibility of the generated sequences; Section 5.4 shows robustness to surrogate misspecification.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of our work are discussed in Section B.1.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not include any theoretical results or formal proofs in this work.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We detail the approach in Section 4, provide the algorithms in Section G and include code to reproduce our results in Section 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We include code to reproduce our results in Section 1 and provide details on the implementation of the baselines in Section A.4.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We detail our experimental setup in Section 5 and describe the surrogate model training procedure in Section A.3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report standard deviation for all figures and tables in Section 5, as noted in each caption.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We detail the compute resources used, including GPU type and runtimes, in Section B.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work conforms with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impacts of our work in Section B.3.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This work does not involve the release of pre-trained models or datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All asset owners are properly credited, and licenses explicitly mentioned and respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We include documented code implementing our method in Section 1, which we will release under the GPL-3.0 license.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This work does not involve LLMs as an important, original, or non-standard component of the core methodology.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.