Directed Evolution of Proteins via Bayesian Optimization in Embedding Space

1st Matouš Soldát
Department of Computer Science
FEE, Czech Technical University in Prague
Prague, Czech Republic
matous.soldat@uochb.cas.cz

2nd Jiří Kléma
Department of Computer Science
FEE, Czech Technical University in Prague
Prague, Czech Republic
klema@fel.cvut.cz

***Abstract*—Directed evolution is an iterative laboratory process of designing proteins with improved function by iteratively synthesizing new protein variants and evaluating their desired property with expensive and time-consuming biochemical screening. Machine learning methods can help select informative or promising variants for screening to increase their quality and reduce the amount of necessary screening. In this paper, we present a novel method for machine-learning-assisted directed evolution of proteins which combines Bayesian optimization with informative representation of protein variants extracted from a pre-trained protein language model. We demonstrate that the new representation based on the sequence embeddings significantly improves the performance of Bayesian optimization yielding better results with the same number of conducted screening in total. At the same time, our method outperforms the state-of-the-art machine-learning-assisted directed evolution methods with regression objective.**

***Index Terms*—protein engineering, directed evolution, Bayesian optimization, large language models, sequence embedding**

I. Introduction

Protein engineering (PE) is the process of designing proteins with desired properties, such as improved stability, catalytic function, or specific binding affinity [1]. PE can be leveraged in industrial applications, environmental applications, medicine, nanobiotechnology, and other fields [1]. Because the functional properties of proteins are determined by their sequence of amino acids [2], the task of PE translates to finding a sequence of amino acids with the desired properties/function. However there is an infinite number of possible protein sequences and non-functional proteins dominate the sequence space [2], which makes PE a challenging task. One of the most widespread approaches to this issue is Directed Evolution (DE) [3].

DE is an iterative laboratory process of creating new biomolecules of desired properties, which mimics Darwinian evolution in a controlled environment [3]. DE circumvents the problem of the vast protein-sequence space filled with non-functional sequences by iteratively mutating an existing protein (often called the wild-type variant) to improve its function [3]. A DE iteration consists of two main steps: mutagenesis, in which parent molecule(s) are mutated and/or recombined to create a library of variants, and screening/selection, where high-quality variants are identified to form a new generation of parents with improved properties [3]. The quality of a given protein variant in terms of the desired property is reported as a numerical value termed fitness.

The wet lab experiments associated with synthetization and screening of the mutated protein variants are expensive and time-consuming [2]. Because of this, the screening process is a common bottleneck of all DE methods. This motivates the employment of machine learning methods to minimize the amount of conducted screening while maximizing the highest obtained fitness. Instead of discarding the low-fitness variants as in traditional DE, methods of machine-learning-assisted directed evolution (MLDE) incorporate information about all screened variants into a model which predicts a protein's fitness based on its sequence [2]. The model is then used to intelligently select new variants for screening which maximize the predicted fitness and/or minimize uncertainty in the model [2].

In this work, we propose a novel MLDE method, Bayesian Optimization in Embedding Space (BOES), which combines Bayesian optimization (BO) [4] with informative embedding of protein sequences extracted by a pre-trained protein language model (PPLM). BOES exploits a PPLM to extract informative embeddings of all variants in the sequence space and the BO procedure is conducted in the obtained embedding space. In each iteration, a Gaussian process (GP) model is fitted to the already screened variants and the next variant for screening is chosen by maximization of expected improvement (EI). To the best of our knowledge, this paper is the first work that successfully combines BO with a PPLM-extracted embedding. In the following text, we describe the new combination and demonstrate its applicability to efficient protein engineering.

II. Related Work

A wide variety of models have been applied to MLDE including simple linear regression models, decision trees/forests, kernel methods, Gaussian Process models, and deep learning [2]. Existing MLDE methods typically employ regression methods which require an additional exploitation stage since they do not prioritize high-fitness variants during the training [5]–[9]. In contrast, BO corresponds to the objective of MLDE much more closely. As an optimization method, BO aims to maximize the fitness function in each iteration and no additional exploitation stage is required. Furthermore, BO is very data efficient, making it an ideal choice in problems where the evaluation of data points is costly and the objective function is multimodal [4]. Both of these properties are key difficulties in exploring protein fitness landscapes [10], [11].

BO guides the exploration-exploitation trade-off based on the selected acquisition function. Two notable acquisition functions are widely used in PE applications. The upper confidence bound (UCB) acquisition function selects the data point with the largest upper confidence bound for evaluation, prioritizing data points that are predicted to be both optimized and uncertain [12]. The relative importance of the prediction and uncertainty can be manually controlled with a weighting parameter [13]. The second notable acquisition function, expected improvement (EI), selects the data point where the expectation over the possible values of the objective function is predicted to have the largest improvement over the current best observation [4]. Similarly to UCB, this approach also strikes a balance between prioritizing data points predicted to be optimized and unexplored data points where the prediction is uncertain. Both methods have been shown to be efficient in the number of function evaluations required to find the global optimum of multi-modal black-box objective functions [14], [15].

UCB has been used in GP regression with a structure-based metric of similarity to provide a probabilistic description of the landscapes for various properties of proteins and to design a cytochrome P450 variant that is more than 5 °C more thermostable than P450 variants previously optimized by different methods and 14 °C more stable than the most stable parent from which it was made [12]. In [16], GP classification and regression models were trained with UCB on expression and localization data from 218 channelrhodopsin [17] variants. Structural similarity obtained by aligning residue-residue contact maps of each variant and counting the number of identical contacts were used as a metric of sequence similarity. In addition to GP regression with UCB criterion, in [18], the method first samples 20 variants from the sequence space that maximize the Gaussian mutual information. The sampled variants are used to fit the GP before the first iteration of sequential optimization. Lastly, in [13], a GP trained with UCB is compared with other methods that model uncertainty differently or do not model uncertainty at all. [13] highlights GP-based methods as particularly useful and shows a consistently strong performance of the GP model.

A GP model with the EI acquisition function has been shown to outperform traditional DE methods in an *in silico* experiment [19]. The proposed method selects variants for evaluation in batches of 19 and uses the squared exponential kernel with Euclidean distances computed from one-hot encoding of the variants at mutated positions. The recent optimization framework for protein DE, termed ODBO [20], combines GP and EI acquisition function with a novel low-dimensional, function-value-based protein encoding strategy and prescreening outlier detection. A protein variant is represented by a feature vector, where each amino acid from the sequence is replaced by the mean or maximum value of the fitness measurements of all variants with the amino acid at that position. Then, in each iteration, the vector representations are inputted into the prescreening via *Extreme Gradient Boosting Outlier Detection* (XGBOD) [21] which filters out potential low fitness samples before the BO step. [20] argues that the novel representation creates a smoother local variable for regression while the prescreening aims to perform more efficient acquisitions in each iteration.

Different protein sequence representations have been applied in BO-based MLDE methods and the advantage of informative representation has been previously demonstrated [20]. The representations learned by PPLMs are known to carry useful information about the function of the variants [22]–[25], which enables the definition of a sensible metric of distance between variants. Furthermore, a key advantage of the PPLM-extracted embedding space over different informative input spaces is that no variants need to be screened for the construction of the input space, saving screening costs. However, PPLM-extracted embeddings have been previously thought to be incompatible with a GP model and BO because of their high dimensionality [26]. In this work, we show that if we limit the number of hyperparameters of the GP model by reducing the number of effective dimensions, BO can be employed in the embedding space to great effect.

III. Problem Formulation

The task of MLDE is formulated as black-box optimization with expensive objective function evaluation. The objective can be formalized as finding the word $\hat{\boldsymbol{x}}$ from the set of all words $\mathcal{X}$ over an alphabet consisting of the twenty common amino acids [27] which maximizes the objective function $f : \mathcal{X} \to \mathbb{R}$,

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathcal{X}}{\arg\max}\, f(\boldsymbol{x}) \tag{1}$$

The objective function f represents the costly screening experiments. For a variant specified by word $\boldsymbol{x} \in \mathcal{X}$, f returns the fitness of the variant. In full generality, the set $\mathcal{X}$ is infinite. In MLDE literature, the problem is often simplified by considering only substitutions of the wild-type protein, limiting the set to l^{20} variants, where l is the protein's length. The problem is usually simplified further, by limiting the number of mutation positions n to very few positions selected by an informed oracle as largely influential to the protein's function, resulting in n^{20} variants. This is also the case in our in-silico experiments, where we use two datasets, each with $n = 4$ pre-selected mutation positions.

IV. Proposed Method

BOES employs BO with a GP model to select variants for screening in an MLDE procedure by maximizing the expected improvement (EI). BO with EI objective function is ideal for MLDE application because it corresponds perfectly to the objective of MLDE in each iteration. That is, each new variant for screening is chosen to maximize the expectation of improvement in the best-so-far screened fitness. This ensures the optimal use of resources after each iteration and eliminates the need for a predefined screening budget, which is necessary when model regression methods are employed.

Before running the BO procedure, BOES uses a PPLM to extract informative sequence embeddings of all variants. The GP model is provided with an embedding function $g : \mathcal{X} \rightarrow \mathbb{E}$ and models the fitness landscape in the m-dimensional sequence embedding space $\mathbb{E} := \mathbb{R}^m$, where variants with similar embeddings are expected to have similar properties.

The BOES algorithm is described in Alg. 1. The MLDE procedure starts with only the wild-type protein in the set of observations $D_1 = \{(\boldsymbol{x}_{wt}, y_{wt})\}$. In each iteration of BO, the GP is fitted to the current set of observations (already screened variants), the EI acquisition function is evaluated at each data point (each variant) and the variant with maximal EI is selected, screened, and added to the set of observations with the observed fitness value.

Algorithm 1 BOES

```
Input: All variants $\mathcal{X}$
Output: Best screened variant $(\boldsymbol{x}, y)$
Initialize dataset $\mathcal{D}_1 \leftarrow \{(\boldsymbol{x}_{wt}, f(\boldsymbol{x}_{wt}))\}$ with the wild-type protein
Fit the model $GP_1$ given $\mathcal{D}_1$
for $n = 1, 2, \ldots, k$ do
    Select new variant for screening by optimizing EI
        $\boldsymbol{x}_{n+1} \leftarrow \arg\max_{\boldsymbol{x} \in \mathcal{X}} EI(\boldsymbol{x}; GP_n)$
    Screen it $\mathcal{D}_{n+1} \leftarrow \mathcal{D}_n \cup \{(\boldsymbol{x}_{n+1}, f(\boldsymbol{x}_{n+1}))\}$
    Fit the model $GP_{n+1}$ given $\mathcal{D}_{n+1}$
end for
return best variant $(\boldsymbol{x}, y) \leftarrow \arg\max_{(\boldsymbol{x},y) \in \mathcal{D}_{k+1}} y$
```

EI	...	Expected Improvement acquisition function.
GP_n	...	Gaussian process model fitted to dataset D_n.
$f : \mathcal{X} \rightarrow \mathbb{R}$	...	Screening, assigns fitness to a variant.

V. Implementation

The ESM-1b model [22] was chosen as the embedding extractor for its ability to produce informative embeddings [22] and the widespread use of the ESM family of PPLMs in MLDE-related literature [6], [9], [28]. A plethora of other PPLMs exist [25], [29]–[31], which could be applied to BOES in the future. The goal of this work is to demonstrate that the combination of PPLMs and BO is a feasible and promising direction for MLDE.

A fundamental problem of employing BO in the embedding space of a PPLM, and the probable reason why this approach has not been successfully employed before, is that BO struggles with high dimensional input spaces [4], [20]. This is problematic because PPLM embeddings tend to have a size in orders of 10^2 to 10^3, depending on the architecture of the language model. This means that we are trying to run BO in an input space with potentially thousands of dimensions. To solve this issue, BOES defines the GP model with a custom implementation of the Matérn 3/2 kernel $k : \mathbb{E} \times \mathbb{E} \rightarrow \mathbb{R}$ and Euclidean distance $d : \mathbb{E} \times \mathbb{E} \rightarrow \mathbb{R}$,

$$k(\boldsymbol{e}, \boldsymbol{e}') = \exp(-\sqrt{3}d(\boldsymbol{e}, \boldsymbol{e}'))(1 + \sqrt{3}d(\boldsymbol{e}, \boldsymbol{e}')) \tag{2}$$

$$d(\boldsymbol{e}, \boldsymbol{e}') = \sqrt{(\boldsymbol{e} - \boldsymbol{e}')^T(\mathbf{I} \cdot \theta^2)(\boldsymbol{e} - \boldsymbol{e}')} \tag{3}$$

to limit the effective number of dimensions to one, so that the surrogate model only fits one length scale hyperparameter θ instead of fitting an individual length scale for each dimension of the embedding $\boldsymbol{e} = g(\boldsymbol{x}) \in \mathbb{E}$ extracted from sequence $\boldsymbol{x}$.

For the prior distribution of the singular length scale, a normal distribution with a mean of zero and a standard deviation σ of $\frac{\sqrt{1280}}{3}$, truncated (and normalized) to the interval $[0; \infty)$, was used. σ was chosen such that the diagonal across the high-dimensional embedding space corresponds approximately to 3σ. Since the embedding space of the used model, ESM-1b, has 1280 dimensions and the absolute values of the elements in the protein embeddings rarely exceed 1 (0.3 % of the elements from all GB1 embeddings have absolute values higher than 1), the size of the diagonal is roughly $\sqrt{1280}$.

The GP model is defined with zero prior mean function $\mu_0 : \mathcal{X} \rightarrow 0$. Zero variance σ^2 is used for noise, effectively removing noise from the model, because the experiments are conducted on a noiseless dataset. The BO procedure is implemented with the BOSS.jl package [32]. The model is fitted with maximum likelihood estimation by the NEWUOA algorithm [33] with 20 starts in a multi-start setting and lower bound on the trust region radius $\rho_{end} = 10^{-4}$. The zero noise variance σ^2 is replaced with a very small positive value to ensure numerical stability of the model. To avoid wasting the screening budget on already screened variants, the value of the acquisition function computed for each already screened variant is replaced by zero before the next variant for screening is chosen. This ensures that the screened variants cannot be chosen again unless the acquisition function value of all variants in the sequence space is also zero, which is practically impossible.

Code for the implemented MLDE procedures and DE simulation baselines, as well as the used datasets, are available at https://github.com/soldatmat/PELLM. The MLDE procedures were implemented in a unified modular framework for in silico DE, which is made available separately as the DESilico.jl package [34].

VI. Results

This section presents the experimental settings used for evaluation of the proposed method including datasets, means of evaluation, baseline DE simulations, and comparison to other implemented methods and SOTA MLDE methods.

A. Data

Experiments were carried out in silico on two datasets. Each dataset maps the fitness landscape of a different wild-type protein. The datasets consist of variant-fitness pairs of nearly all possible variants of the wild-type protein mutated at 4 positions. The fitness of each unmeasured variant is assumed to be zero in all conducted experiments since the unmeasured variants are considered meaningless to biologists [6], [9]. The mutation positions were selected as largely influential to the structure and function of the protein.

GB1 dataset [10] is a dataset of variants of the protein G domain B1, mutated at four positions with non-linear epistasis (V39, D40, G41, V54). Fitness values of GB1 variants represent the binding ability to the antibody IgG-Fc and range from 0.0 to 8.76. Fitness value of 1.0 corresponds to the binding ability of the wild-type protein.

PhoQ dataset [35] consists of variants of protein kinase PhoQ obtained by mutating the wild-type sequence at four positions critical to the function of the protein (A284, V285, S288, T289). The fitness values refer to the phosphatase or kinase activity of different PhoQ variants and range from 0.0 to 133.59. Fitness of the wild-type protein kinase PhoQ is 3.29.

B. Performance on the Wild-type Protein

To evaluate the performance of BOES, we compared its performance to SOTA regression-based MLDE methods, which all minimize the prediction error of the model during the DE procedure. The comparison in Table I includes results of SOTA MLDE methods reported in [9]. A concise description of the included methods is adapted from [9]:

- MLDE [5] trains an ensemble of shallow neural networks as fitness predictors on randomly sampled variants.
- ftMLDE, focused training MLDE [6], is a strategy for running MLDE with training sets designed to avoid holes. The comparison includes ftMLDE with two sampling strategies, EVmutation [36] and MSA-transformer [37].
- CLADE [7] trains a fitness predictor with high-fitness mutants obtained through a hierarchical clustering sampling method.
- CLADE 2.0 [8] selects the high-fitness mutants with a scoring function that employs an ensemble of methods including a PPLM.
- AFP-DE [9] uses a PPLM to sample variants and extract sequence embeddings. Iteratively trains a fitness predictor with the sampled variants and finetunes the PPLM with variants with high predicted fitness.

Furthermore, simulation of a single mutation walk (SMW) [5] is included in the comparison to serve as a baseline with no use of ML methods. Implementation details of the SMW simulation can be found in [38]. Lastly, one conceptually different optimization MLDE method is included in the comparison. Neighborhood Search Directed Evolution (NSDE) [38] performs a greedy graph search in a neighborhood graph constructed from the variants' sequence embeddings extracted by a PPLM.

The regression-based methods are tested with a screening budget of 80 variants and two different splits between the part of the budget used for training and the rest of the budget left to screen variants with high predicted fitness. SMW and the two optimization methods do not split the resources, so Table I contains just a single result for these methods.

Dataset	GB1		PhoQ	
Screening budget	$(24+56)$	$(48+32)$	$(24+56)$	$(48+32)$
SMW	3.90		18.44	
MLDE	3.93	4.43	6.55	13.23
ftMLDE (EVmut.)	4.99	5.27	22.04	8.68
ftMLDE (trans.)	4.98	5.31	17.77	26.18
CLADE	4.88	3.92	21.51	25.65
CLADE 2.0	4.36	6.01	24.45	21.51
AFP-DE	6.20	6.20	24.98	28.19
NSDE	4.54		20.71	
BOES	**7.28**		**37.94**	

TABLE I: Comparison of BOES with SOTA regression-based methods: maximum fitness obtained with 80 screened variants starting from the wild-type protein is reported. Regression-based methods split the screening budget between training and exploitation (24 + 56 or 48 + 32). Optimization methods screen all 80 variants during the optimization procedure.

Table I shows a clear dominance of the proposed BOES method. The results confirm that the optimization approach to MLDE can be more efficient than methods with a regression objective.

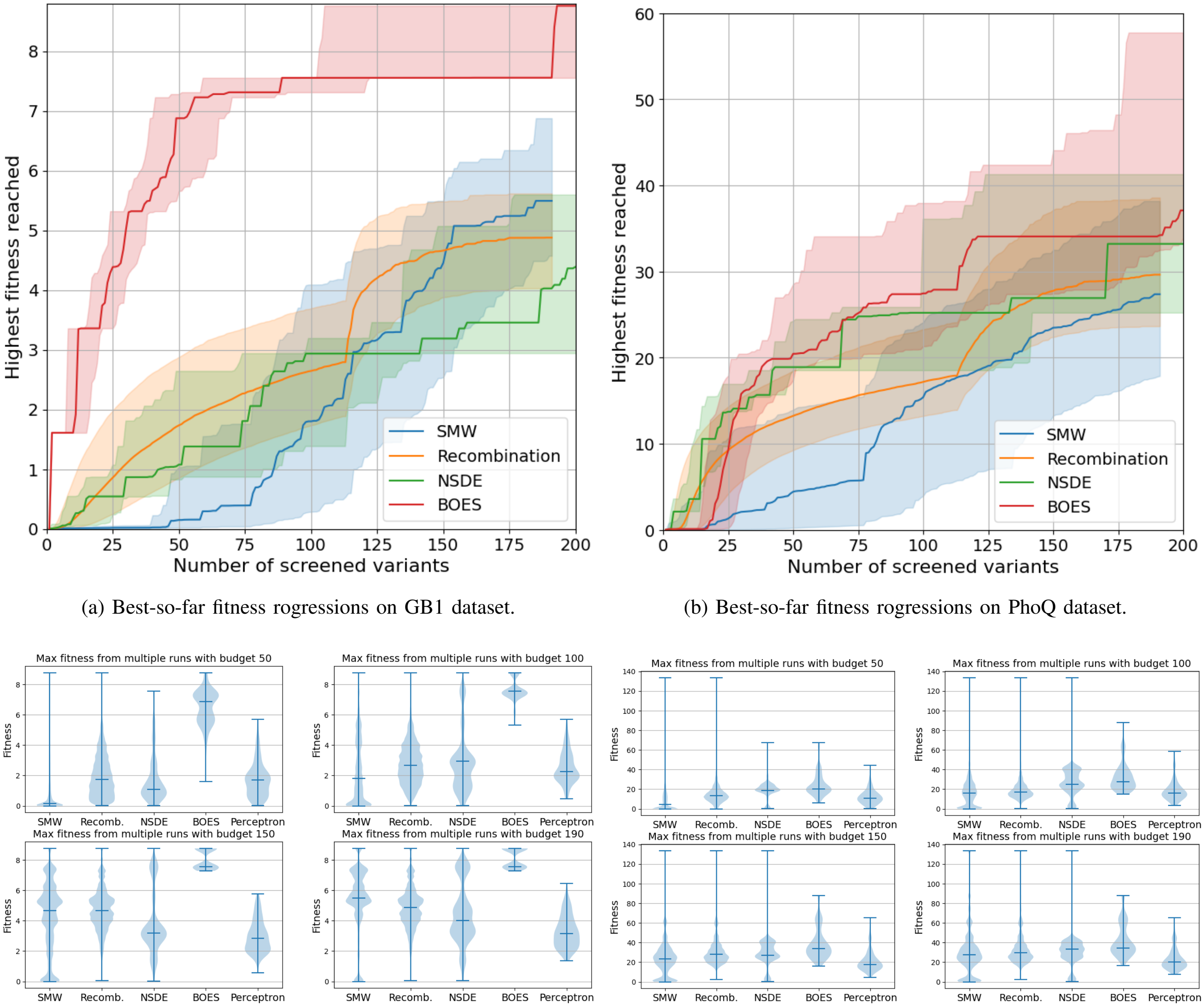
C. Robustness to the Starting Protein

While the performance on the wild-type protein corresponds to the practical use of MLDE algorithms, making conclusions about the methods' performance based on a single run, albeit on two different datasets, would be ill-advised. The results obtained from such a limited evaluation can be strongly skewed by the properties of the specific dataset. Especially local-search methods, like the SMW baseline or the NSDE method based on a KNN graph, could potentially show wildly different efficiency based on the relative position of the starting variant, the global optimum, and any local optima located between them in the sequence space.

To ensure that the methods' hyperparameters are not overfitted to the path from the wild-type variant to the global optimum, a test of robustness to the starting protein was conducted. This test also gives helpful insight into the variance in performance of the DE methods. Among the tested methods were the BOES method, the aforementioned NSDE method [38], a perceptron-training method based on the AFP-DE procedure [9] with a different *exploration* stage implementation, and two simulations of classical DE methods without the use of ML: SMW and Recombination [5]. Implementation details of the tested methods are included in [38].

The evaluation was carried out by running each method repeatedly with a different starting variant for 200 to 160,000 runs, based on the computational demand of each method. The first quartile, median, and third quartile values of the highest achieved fitness by each method are reported in Fig. 1a and Fig. 1b for the GB1 and PhoQ dataset, respectively. Additionally, distributions of the highest achieved fitness at 50, 100, 150, and 190 screened variants are reported in Fig. 1c and Fig. 1d.The starting variant is also counted towards the number of screened variants.

The median fitness curve of the BOES method in Fig. 1a shows that BOES usually finds the globally optimal variant



(a) Best-so-far fitness rogressions on GB1 dataset.

(b) Best-so-far fitness rogressions on PhoQ dataset.

(c) Distributions of the highest achieved fitness on GB1 dataset.

(d) Distributions of the highest achieved fitness on PhoQ dataset.

Fig. 1: Resulting fitness over multiple runs from sampled starting variants. (a, b) Best-so-far fitness progressions. Bold line represents median, highlighted areas correspond to interquartile range. (c, d) Distributions of the highest achieved fitness with different screening budgets.

in under 200 screened variants even when starting from a different, non-functional variant in the GB1 sequence space. The violin plots in Fig. 1c and Fig. 1d illustrate the superiority of our method further and show, that with increasing screening budget, BOES gives an increasing lower bound on the probable resulting fitness, whereas the other evaluated methods can strongly underperform more often.

D. Ablation Study

Advantage of Embedding Space To evaluate the effect of the informative input space on the performance of the BO procedure, we compare our results to a method proposed in [19] which, in its most simple form, performs BO with a GP model and EI acquisition function directly in the protein sequence space. The distance between two variants is computed simply from one-hot encoding of the amino acids at mutated positions. This approach corresponds to the proposed BOES method in terms of the employed model and acquisition function but uses a straightforward definition of the input space and kernel function in place of the embedding space. This makes the method an ideal candidate for evaluation of the effect of the embedding space on the performance of BOES. The average results of the SMW baseline, BOES method, and the aforementioned method with one-hot encoding, denoted as *GP+EI*, are included in Table II. It is important to note that in the *GP+EI* method from [19], the model is initially trained on

GB1 dataset	Avg maximum fitness	Screening budget
SMW	5.35 ± 2.14	191
GP+EI	7.28	$20 + 191$
BOES	$\mathbf{8.14} \pm 0.62$	191

TABLE II: Comparison of BOES with BO conducted in the original protein sequence space: Mean of the maximum obtained fitness from multiple runs is reported. Results of the implemented methods are accompanied by standard deviation.

GB1 dataset	Avg maximum fitness	Screening budget
NaiveBO + GP	6.40 ± 0.79	$40 + 50$
TuRBO	$\mathbf{6.57} + 1.02$	$40 + 50$
BOES	6.47 ± 1.15	**50**

TABLE III: Comparison of BOES to BO conducted with a different informative sequence representation: Mean of the maximum obtained fitness from multiple runs is reported with the standard deviation.

20 randomly selected variants before the first iteration of BO, which are not counted towards the screening budget, skewing the comparison in its favor. One last note-worthy distinction between BOES and the method reported in [19] is that, unlike BOES, this method selects new variants for screening in batches of 19. Comparison in Table II decidedly confirms a positive effect of employing BO in the embedding space over the original sequence space with a one-hot encoding-based kernel function.

Other Informative Input Spaces The conducted comparison to a BO-based method defined on the original sequence space proves the positive effect of the innovative input space qualitatively. However, a comparison to another state-of-the-art BO-based method with a different, yet also informative, input space can help assess the performance of the proposed method quantitatively. The ODBO framework [20] employs BO for DE in combination with a novel encoding of amino acids based on the fitness of observed variants with the specific amino acids at the specified mutation position. In Table III, results of the BOES method with a screening budget of 50 variants are compared to a classical BO procedure with a GP model and the positional amino-acid encoding of variants from [20] and to a trust region BO procedure (TuRBO) [39] with the same model and encoding.

A critical difference between the two methods of sequence-space representation is that the positional amino-acid encoding proposed in [20] requires an initial dataset of screened variants in which each amino acid appears at each mutation site at least two times, while the PPLM-extracted embedding space used in BOES requires no screened variants for its construction. The original paper presents a solution to this obstacle in the form of an initial sampling algorithm, which for the GB1 dataset constructs an initial set of 40 variants. This means that while each of the BO procedures compared in Table III is provided with a screening budget of 50 variants, the construction of the encoding that precedes the two procedures from [20] requires an additional 40 screening experiments, which the BOES method saves in comparison.

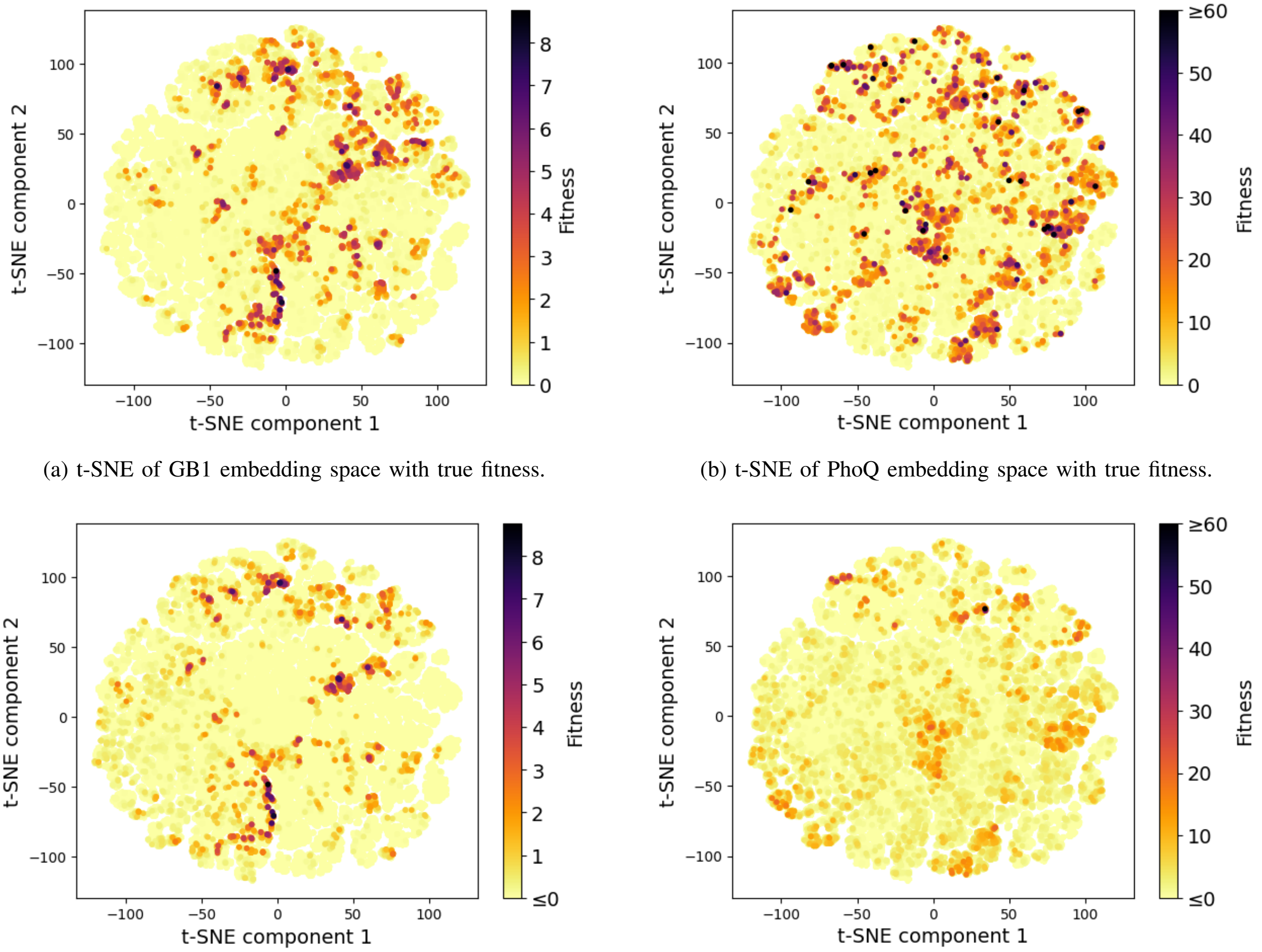
Results in Table III reveal that all three compared BO-based methods produce proteins of extremely similar quality with BOES outperforming the other classical BO method, labeled NaiveBO, and the trust region variant slightly outperforming BOES. It should be noted that a trust region variant of BOES could also be implemented, which would most probably improve the original BOES method's performance. Similarly, the authors of the compared BO method [20] propose two additional improvements: prescreening outlier detection via XGBOD [21] and employing a BO procedure robust to outliers [40]. The variant of the authors' method with these improvements outperforms BOES, but the improvements could also be combined with BOES. Adding the prescreening outlier detection step requires a set of already screened variants. To circumvent this, the outlier detection could be enabled after a certain number of BOES iterations. Additionally, the XGBOD method could be replaced with an unsupervised outlier detection method in the initial iterations of BOES. A version of BOES with the additional improvements from [20] can be expected to yield similar results to the full version of ODBO [20] while saving screening costs on the construction of sequence representation.

E. Visualizing the Embedding Space

The BOES method operates on a PPLM-extracted sequence embedding space instead of using the raw sequences of amino acids. It is crucial that the embedding space provides a sensible metric of similarity between variants as well as encodes useful information about the variants' properties. To ensure that this assumption holds, the embedding space was visualized with dimensionality reduction methods.

First, a joint principal component analysis (PCA) was conducted on sequence embeddings of all variants from both datasets (GB1 and PhoQ) as a sanity check. The PCA confirmed that the two datasets are easily separable. The first principal component alone accounts for 97.4 % of variance in the joint distribution and separates the two datasets into two clear clusters.

Next, PCA was conducted for each dataset separately to visually confirm whether expected features of the sequence space, like local maxima and distinguishable areas with low/high fitness, are present in the embedding space. Results of PCA in both of the datasets revealed that the first two principal components together explain roughly 40 % of the variance. That is a very large portion, considering that the ESM-1b embedding space has 1280 dimensions. The PCA analyses of the standalone datasets both showed one large area with functional variants. To assess whether the embedding space is capable of capturing local maxima in fitness landscapes, the embedding space of each dataset was visualized with the t-SNE method, which emphasizes maintaining low distances between close data points, preserving local clusters. The t-SNE visualization is plotted in Fig. 2a for the GB1 dataset and in Fig. 2b for the PhoQ dataset. Both figures confirm the presence of local clusters of high-fitness variants.



(a) t-SNE of GB1 embedding space with true fitness.

(b) t-SNE of PhoQ embedding space with true fitness.

(c) t-SNE of GB1 embedding space with predicted fitness.

(d) t-SNE of PhoQ embedding space with predicted fitness.

Fig. 2: Visualisation of (a, c) GB1 and (b, d) PhoQ embedding space extracted with ESM-1b PPLM with (a, b) true fitness and (c, d) fitness predicted by GP model trained on 384 screened variants in BOES run from the wild type protein.

F. Modelling the Embedding Space

As a secondary result, the fitness landscape modeled by the BOES method was visualized alongside the initial t-SNE plots in Fig. 2. Fig. 2c and Fig. 2d show fitness predicted by a GP model trained on 384 screened variants by the BOES method when initiated with the wild-type protein of the GB1 and PhoQ datasets, respectively. The visualization of predicted fitness shows that BOES was able to identify multiple local clusters of high-fitness variants in both datasets. Especially the result on the GB1 dataset in Fig. 2c reveals that almost all of the major clusters were identified by BOES. High normalized discounted cumulative gain (NDCG) values (GB1: 0.88, PhoQ: 0.79) confirm that BOES effectively models the fitness landscape to rank high-fitness variants, which is essential in MLDE.

VII. Conclusion

In this paper, we have presented a novel method of machine-learning-assisted directed evolution (MLDE), termed Bayesian optimization in embedding space (BOES). Feasibility of the proposed method was confirmed in silico on two datasets. Our method outperforms SOTA MLDE methods with a regression objective. Moreover, the informative representation of the input space based on the sequence embeddings extracted by a pre-trained protein language model (PPLM) has been shown to significantly improve the performance of Bayesian optimization (BO) over optimization in the original protein sequence space. The BOES method produces proteins of comparable quality to other state-of-the-art BO-based methods that employ different informative representations of the input space while significantly reducing screening costs since there is no screening needed within the construction of the sequence representation. This improvement can result in saved resources on experimental costs or more resources for additional iterations of DE, yielding better results with the same amount of conducted screening in total. For future development, we suggest combining the innovative input space representation proposed in this paper with the improvements to the standard BO procedure suggested in [20]. We order the suggestions based on their effect on the performance of the ODBO method [20]. Firstly, conducting prescreening outlier detection via *Extreme Gradient Boosting Outlier Detection* [21] in later iterations. Secondly, implementing a BO procedure robust to outliers based on [40] and finally, employing trust region BO [39].

Acknowledgment

Special thanks go to Šimon Soldát for useful discussions and insight into Bayesian optimization.

This research work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS23/184/OHK3/3T/13. The access to the computational infrastructure of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" is also gratefully acknowledged.

References

[1] Q. Ali, A. Sultan, A. Azhar, N. Kanwal, F. Ali *et al.*, "Protein engineering: A brief overview methodologies and applications," *Life Science Journal*, vol. 13, no. 12, 2016.

[2] K. K. Yang, Z. Wu, and F. H. Arnold, "Machine-learning-guided directed evolution for protein engineering," *Nature methods*, vol. 16, no. 8, pp. 687–694, 2019.

[3] Y. Wang, P. Xue, M. Cao, T. Yu, S. T. Lane *et al.*, "Directed evolution: methodologies and applications," *Chemical reviews*, vol. 121, no. 20, pp. 12 384–12 444, 2021.

[4] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[5] Z. Wu, S. J. Kan, R. D. Lewis, B. J. Wittmann, and F. H. Arnold, "Machine learning-assisted directed protein evolution with combinatorial libraries," *Proceedings of the National Academy of Sciences*, vol. 116, no. 18, pp. 8852–8858, 2019.

[6] B. J. Wittmann, Y. Yue, and F. H. Arnold, "Informed training set design enables efficient machine learning-assisted directed protein evolution," *Cell systems*, vol. 12, no. 11, pp. 1026–1045, 2021.

[7] Y. Qiu, J. Hu, and G.-W. Wei, "Cluster learning-assisted directed evolution," *Nature computational science*, vol. 1, no. 12, pp. 809–818, 2021.

[8] Y. Qiu and G.-W. Wei, "Clade 2.0: evolution-driven cluster learning-assisted directed evolution," *Journal of chemical information and modeling*, vol. 62, no. 19, pp. 4629–4641, 2022.

[9] M. Qin, K. Ding, B. Wu, Z. Li, H. Yang *et al.*, "Active finetuning protein language model: A budget-friendly method for directed evolution," in *ECAI 2023*. IOS Press, 2023, pp. 1914–1921.

[10] N. C. Wu, L. Dai, C. A. Olson, J. O. Lloyd-Smith, and R. Sun, "Adaptation in protein fitness landscapes is facilitated by indirect paths," *Elife*, vol. 5, p. e16965, 2016.

[11] D. Simoncini, S. Barbe, T. Schiex, and S. Verel, "Fitness landscape analysis around the optimum in computational protein design," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 355–362.

[12] P. A. Romero, A. Krause, and F. H. Arnold, "Navigating the protein fitness landscape with gaussian processes," *Proceedings of the National Academy of Sciences*, vol. 110, no. 3, pp. E193–E201, 2013.

[13] B. Hie, B. D. Bryson, and B. Berger, "Leveraging uncertainty in machine learning accelerates biological discovery and design," *Cell systems*, vol. 11, no. 5, pp. 461–477, 2020.

[14] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.

[15] A. D. Bull, "Convergence rates of efficient global optimization algorithms." *Journal of Machine Learning Research*, vol. 12, no. 10, 2011.

[16] C. N. Bedbrook, K. K. Yang, A. J. Rice, V. Gradinaru, and F. H. Arnold, "Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization," *PLoS computational biology*, vol. 13, no. 10, p. e1005786, 2017.

[17] K. Deisseroth and P. Hegemann, "The form and function of channelrhodopsin," *Science*, vol. 357, no. 6356, p. eaan5544, 2017.

[18] J. C. Greenhalgh, S. A. Fahlberg, B. F. Pfleger, and P. A. Romero, "Machine learning-guided acyl-acp reductase engineering for improved in vivo fatty alcohol production," *Nature communications*, vol. 12, no. 1, p. 5825, 2021.

[19] T. S. Frisby and C. J. Langmead, "Fold family-regularized bayesian optimization for directed protein evolution," in *20th International Workshop on Algorithms in Bioinformatics (WABI 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[20] L. Cheng, Z. Yang, C. Hsieh, B. Liao, and S. Zhang, "Odbo: Bayesian optimization with search space prescreening for directed protein evolution," *arXiv preprint arXiv:2205.09548*, 2022.

[21] Y. Zhao and M. K. Hryniewicki, "Xgbod: improving supervised outlier detection with unsupervised representation learning," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[22] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin *et al.*, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, p. e2016239118, 2021.

[23] J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher *et al.*, "Bertology meets biology: Interpreting attention in protein language models," *arXiv preprint arXiv:2006.15222*, 2020.

[24] R. Rao, J. Meier, T. Sercu, S. Ovchinnikov, and A. Rives, "Transformer protein language models are unsupervised structure learners," *Biorxiv*, pp. 2020–12, 2020.

[25] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang *et al.*, "Prottrans: Toward understanding the language of life through self-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 10, pp. 7112–7127, 2021.

[26] J. Yang, R. G. Lal, J. C. Bowden, R. Astudillo, M. A. Hameedi *et al.*, "Active learning-assisted directed evolution," *bioRxiv*, pp. 2024–07, 2024.

[27] M. J. Lopez and S. S. Mohiuddin, "Biochemistry, essential amino acids," 2020.

[28] B. L. Hie, K. K. Yang, and P. S. Kim, "Evolutionary velocity with protein language models predicts evolutionary dynamics of diverse proteins," *Cell Systems*, vol. 13, no. 4, pp. 274–285, 2022.

[29] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu *et al.*, "Language models of protein sequences at the scale of evolution enable accurate structure prediction," *bioRxiv*, 2022.

[30] D. Hesslow, N. Zanichelli, P. Notin, I. Poli, and D. Marks, "Rita: a study on scaling up generative protein sequence models," *arXiv preprint arXiv:2205.05789*, 2022.

[31] N. Ferruz, S. Schmidt, and B. Höcker, "Protgpt2 is a deep unsupervised language model for protein design," *Nature communications*, vol. 13, no. 1, p. 4348, 2022.

[32] Š. Soldát, "BOSS.jl (Bayesian Optimization with Semiparametric Surrogate)." [Online]. Available: https://github.com/soldasim/BOSS.jl

[33] M. J. Powell, "The newuoa software for unconstrained optimization without derivatives," *Large-scale nonlinear optimization*, pp. 255–297, 2006.

[34] M. Soldát, "DESilico.jl (Directed Evolution in Silico)." [Online]. Available: https://github.com/soldatmat/DESilico.jl

[35] A. I. Podgornaia and M. T. Laub, "Pervasive degeneracy and epistasis in a protein-protein interface," *Science*, vol. 347, no. 6222, pp. 673–677, 2015.

[36] T. A. Hopf, J. B. Ingraham, F. J. Poelwijk, C. P. Schärfe, M. Springer *et al.*, "Mutation effects predicted from sequence co-variation," *Nature biotechnology*, vol. 35, no. 2, pp. 128–135, 2017.

[37] R. Rao, J. Liu, R. Verkuil, J. Meier, J. F. Canny *et al.*, "Msa transformer," *bioRxiv*, 2021. [Online]. Available: https://www.biorxiv.org/content/10.1101/2021.02.12.430858v1

[38] M. Soldát, "Protein engineering with large language models," Master's thesis, České vysoké učení technické v Praze. Vypočetní a informační centrum., 2024.

[39] D. Eriksson, M. Pearce, J. Gardner, R. D. Turner, and M. Poloczek, "Scalable global optimization via local bayesian optimization," *Advances in neural information processing systems*, vol. 32, 2019.

[40] R. Martinez-Cantin, K. Tee, and M. McCourt, "Practical bayesian optimization in the presence of outliers," in *International conference on artificial intelligence and statistics*. PMLR, 2018, pp. 1722–1731.