

A Continuous Relaxation for Discrete Bayesian Optimization

Richard Michael¹ Simon Bartels² Miguel González-Duque^{1,3} Yevgen Zainchukovskyy⁴ Jes Frellsen^{4,3}
 Søren Hauberg^{4,3} Wouter Boomsma^{1,3}

Abstract

To optimize efficiently over discrete data and with only few available target observations is a challenge in *Bayesian optimization*. We propose a continuous relaxation of the objective function and show that inference and optimization can be computationally tractable. We consider in particular the optimization domain where very few observations and strict budgets exist; motivated by optimizing protein sequences for expensive to evaluate bio-chemical properties. The advantages of our approach are two-fold: the problem is treated in the continuous setting, and available prior knowledge over sequences can be incorporated directly. More specifically, we utilize available and learned distributions over the problem domain for a weighting of the Hellinger distance which yields a covariance function. We show that the resulting acquisition function can be optimized with both continuous or discrete optimization algorithms and empirically assess our method on two bio-chemical sequence optimization tasks.

1. Introduction

Optimizing discrete sequences with respect to multi-dimensional properties is particularly challenging. If no gradient information is available and evaluation is expensive, one can use *Bayesian optimization* (BO) to solve the problem (Močkus, 1975; Shahriari et al., 2016). The problem becomes even more challenging if our inputs are sequences of discrete elements, only few observations are given, and a strict limit is imposed on the number of possible evaluations. Yet, most Bayesian optimization algorithms assume the optimization problem to be continuous. Furthermore for contemporary BO approaches the number of starting observations and available budget are often only tested in scenarios where these values are quite high (Swersky et al.,

¹DIKU, University of Copenhagen ²IMT, Université de Toulouse III ³Center for Machine Learning in Life Science ⁴DTU. Correspondence to: <{richard.michael ; wb}@di.ku.dk>.

Copyright 2024 by the author(s).

2013; 2020). In this work we address the challenge of only a hand-full of available observations at the start and a strict budget *i.e.* only a few hundred observations are feasible – an *ice-cold* start. We show how to transform the optimization of discrete sequential inputs such that it is in the continuous domain, and how to utilize available probabilistic models over sequences. Specifically, we relax the problem by mapping sequences to distributions and optimize in distribution space, which lets us incorporate prior information directly.

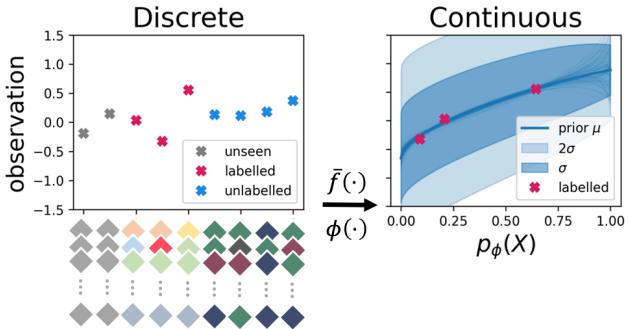


Figure 1: The proposed problem transformation: given a set of discrete element sequences (X (blocks) elements in sequences) with discontinuous observations (left) we continuously relax the objective \bar{f} (right), and assign a \mathcal{GP} prior to it. The probability space over the elements is given by ϕ , a pretrained *a priori* model which parameterizes the distributions over the sequences and elements therein (middle, bottom).

The setup of restricted discrete optimization problems is highly relevant for domains like protein engineering and drug discovery, where sequences of tokens are optimized, *i.e.* a string of amino acids, or small molecule tokens (Biswas et al., 2021; Gao et al., 2022). Proteins are the basic building blocks of life, making their form and function the subject of scientific discovery and engineering. To evaluate protein candidates often requires laboratory experiments making *target function* values expensive to obtain. Since proteins can be described as a sequence of amino

acids from an alphabet of 20 naturally occurring ones, the search space is discrete and even for small proteins with 100 amino acids, the space of 20^{100} possible elements is infeasibly large (Maynard Smith, 1970). At the same time, only a very small subset of these sequences is likely to be useful (Tian & Best, 2017). These observations render both brute-force searches and direct optimization strategies for discovering useful proteins intractable and undesirable. Previous work in optimizing bio-chemical sequences uses vast databases of known sequences to learn a low-dimensional continuous representation and perform the search for new and useful candidates therein (Frazer et al., 2021; Notin et al., 2022; Stanton et al., 2022). It has been observed that in some cases, the Euclidean distance measure in learned latent representations can be a poor proxy for protein similarity (Detlefsen et al., 2022), which indirectly influences Bayesian optimization techniques negatively. This motivates us to develop an algorithm for optimizing discrete sequences directly and validate our method in the domain of protein engineering.

Our contributions are threefold, we *first* propose a **continuous relaxation of the objective function** in the BO algorithm, and *secondly* show that inference and optimization remain computationally tractable. Specifically by proposing a covariance function that is linear in runtime with respect to input lengths. *Lastly*, we demonstrate the applicability in the domain of *computational biology* by empirically assessing two discrete sequence optimization tasks.

2. Background

2.1. Problem statement

Given are discrete sequences of length $L \in \mathbb{N}$ and an alphabet of A tokens such that, at each position of each sequence, $|A|$ elements are available. This defines our discrete input set $\mathbb{X} := \cup_{l=1}^L A^l$. We are given a *costly* black-box function $f : \mathbb{X} \rightarrow \mathbb{R}$, for which we wish to find the minimizer $\mathbf{x}_* := \arg \min_{\mathbf{x} \in \mathbb{X}} f(\mathbf{x})$ using as few function evaluations as possible.

For the specific case of proteins, A is the set of naturally occurring 20 amino acids, then \mathbb{X} contains all possible protein sequences up to a particular length L . The function f may then be some measurable property of a protein system (metabolic, regulatory, etc.), which is costly to evaluate as it requires us to run a set of wetlab experiments.

2.2. Bayesian optimization

Bayesian optimization typically consist of a surrogate model m for f , and an acquisition function α (Garnett, 2022). The model is updated at each iteration given the observations of all experiments and the acquisition function is numerically optimized on the surrogate to select the next configuration

for the evaluation of f . Typically, m is a Gaussian process (Rasmussen & Williams, 2006), and popular choices for α are *Expected Improvement* (Jones et al., 1998) and the *Upper Confidence Bound* (Srinivas et al., 2012).

Since the input space is discrete, continuous numerical optimization algorithms cannot directly be used to find the optima of α . One possible approach to this issue is to fit a latent variable model and to perform the optimization in latent space (Lu et al., 2018; Gómez-Bombarelli et al., 2018). Since it is unclear whether the Euclidean distance in representation space is a reliable proxy for similarity we choose a different approach, in the form of a continuous relaxation, a constraint probability space, and distance measure of probability vectors – as described in section 3.1.

2.3. Gaussian process regression

Gaussian processes (GPs) are a typical choice for m due to their expressiveness and closed-form inference. Formally, a GP is a collection of random variables, such that every finite subset follows a multivariate normal distribution (Rasmussen & Williams, 2006, p. 13). A GP prior is described by a mean function μ (often set to $\mu(\mathbf{x}) := 0$), and a positive definite covariance function (kernel) $k : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$. Assuming that observations of f are distorted by Gaussian noise, the posterior over the function f conditioned on these observations is again a Gaussian process.

2.4. Related work

Wan et al. (2021) motivate high dimensional discrete BO as a challenge and consider a categorical and mixed search space for optimization (*CASMOPOLITAN*). Stanton et al. (2022) proposed *LaMBO*, a Bayesian optimization routine leveraging a learned lower-dimensional latent space and autoregressive model to optimize discrete sequences (small molecules and protein sequences). A subsequent extension of the algorithm (*LaMBO-2*) still optimizes discrete space with ensemble methods replacing GPs (Gruver et al. (2023)). Further approaches conduct Bayesian optimization on structured inputs using string-based kernels (Moss et al., 2020) or leverage structure of the latent space i.e. for molecules (Gómez-Bombarelli et al., 2018; Tripp et al., 2020).

To the best of our knowledge, no current approach formulates BO continuously with a *covariance function on distributions* over discrete sequence inputs, defining the Gaussian Process prior over the *objective* that extends to a *continuous treatment of the acquisition function*. Furthermore, obtaining a kernel by applying a weighting from distributions has not been done prior to this work (see section 3.2.1).

On a first glimpse, most closely related to our work may be the articles by Garrido-Merchán & Hernández-Lobato (2020) or Daulton et al. (2022). The former proposes a

continuous relaxation for categorical inputs on discrete and mixed spaces, whereas we relax the objective. The latter approach introduces a continuous relaxation in its *probabilistic reparameterization* (PR) of the acquisition function. This is different from our relaxation of the objective and does not consider the constrained probability space, which we introduce in Eq. (1) and (2) respectively and which is required for computational feasibility.

Issues with Gaussian process models over latent space representations

Lu et al. (2018) investigated Bayesian optimization defining a Gaussian process model directly on the latent space of a variational autoencoder (VAE). Regular choices of covariance functions, *i.e.* the Matérn or Squared exponential kernel (Rasmussen & Williams, 2006) rely on a distance measure to assess covariance of two observations. However, this property is not always satisfied in practice, since a learned latent space does not necessarily have a Euclidean measure (Arvanitidis et al., 2021).

A key assumption for kernels based *just* on Euclidean distance measures is that far-apart observations are independent from one another given a particular length-scale. This assumption is not necessarily fulfilled in learned latent representations, here two sequences can be highly related, while being far away in latent space or close in latent space and not at all related (Detlefsen et al., 2022).

Issues with Bayesian optimization’s budget Some contemporary Bayesian optimization routines rely on a significant number of black-box function evaluations (Maus et al., 2023) either at the start or through a large budget of possible evaluations. These approaches are thus limited in prohibitively expensive setting where we can only make a few hundred observations, such as bio-chemical assay experiments (Gao et al., 2022). Given a strict set-up, only few observations will be available at the start and the budget of evaluating the function limited to few expensive observations.

3. Continuously Relaxed Bayesian Optimization

For our main contribution we propose a continuous relaxation of the objective function which at a first glance appears to make the optimization problem more complicated. We show how the problem can be restricted and which crucial ingredients exist to solve the computational challenges induced by the relaxation. Following the proposed steps we can optimize in our problem-space using *a priori* available probability densities for our surrogate function. This requires us to consider a covariance function that acts on probability distributions, which we develop below.

3.1. From discrete to continuous space

We can turn this discrete optimization problem (Section 2.1) into a continuous one by optimizing in the space of probability distributions over \mathbb{X} , minimizing the expected function value of f . This way we obtain a **differentiable function \bar{f} with the same optima** (see proof in Supplementary Appendix A):

$$\bar{f}(\mathbf{p}) := \mathbb{E}_{\mathbf{p}} f(x) = \sum_{x \in \mathbb{X}} f(x) \mathbf{p}_x. \quad (1)$$

where $\mathbf{p} \in \mathbb{P} := \{\mathbf{p} \in [0, 1]^{|\mathbb{X}|} \mid \sum_i p_i = 1\}$ are probability distributions over \mathbb{X} , which are real vectors of length $|\mathbb{X}|$ with elements between 0 and 1 whose components sum to 1. Note that, each element of $x \in \mathbb{X}$ can be represented as $\mathbf{p}_x[x'] := \mathbb{1}_{x'=\mathbf{x}}(x')$ such that $f(x) = \bar{f}(\mathbf{p}_x)$.

Having a continuous objective function with preserved optima, may naively appear sufficient for the successful application of BO. However, our approach is accompanied by a couple of computational challenges, which we will point out in the following.

Representation Even for small proteins sequences of length less than 100, any $\mathbf{p} \in \mathbb{P}$ has more entries than there are atoms in the observable universe (Maynard Smith, 1970). The space \mathbb{P} will be too large to work with and we will have to restrict it. **In the following, we will consider \mathbb{P}_f , the space of factorizing distributions of length L .**

$$\mathbb{P}_f := \left\{ \mathbf{p} \in [0, 1]^{L \times A} \mid \mathbf{p} \geq 0, \forall l : \sum_{a=1}^A \mathbf{p}_{l,a} = 1 \right\} \quad (2)$$

Inference Having a Gaussian process model over f naturally induces a model over \bar{f} , by using the kernel function $k'(\mathbf{p}, \mathbf{q}) := \sum_{x, x' \in \mathbb{X}} \mathbf{p}[x] k(x, x') \mathbf{q}[x']$. The evaluation of this canonical kernel function is intractable though, even if we consider a restriction of \mathbb{P} . Approaches for this issue are presented in Section 3.2.

Optimization Having established a model m over \bar{f} we must also address the problem of how to optimize a Bayesian optimization’s acquisition function α_m . Even though this acquisition function is continuous, the fact that proposed inputs must remain probability distributions prevents us from freely using any optimizer¹. This topic is discussed in Section 3.3.

3.2. The model

A Gaussian process prior over f induces a Gaussian process belief over \bar{f} , yet the computation of the posterior over \bar{f}

¹The vector components have to be positive and sum to one.

is intractable – even if we restrict ourselves to \mathbb{P}_f . The **key idea is to place a Gaussian process prior directly over \bar{f}** instead of using the induced prior from f . Doing so will allow us to do computationally tractable inference. We note that in general it is *not* the case that $\bar{f}(\mathbf{p}) = \sum_x p(x)\bar{f}(\mathbb{1}_x)$, however we can address this in practice. The main challenge is to find a kernel function which can exploit the structural properties of \mathbb{P}_f ; for example distances informed by *prior* probability densities, and subsequently discard regions of low-probability.

3.2.1. THE WEIGHTED HELLINGER KERNEL

One such kernel $k(\cdot, \cdot) : \mathbb{P}_f \times \mathbb{P}_f \rightarrow \mathbb{R}$ is based on the *Hellinger* distance r (Hellinger, 1909):

$$r(\mathbf{p}, \mathbf{q}) := \sqrt{\frac{1}{2} \sum_{x \in \mathbb{X}} \left(\sqrt{p(x)} - \sqrt{q(x)} \right)^2}, \quad (3)$$

$$k(\mathbf{p}, \mathbf{q}) := \theta \exp(-\lambda r(\mathbf{p}, \mathbf{q})). \quad (4)$$

We know r to be negative definite (Harandi et al., 2015), and therefore k is a *positive definite* kernel for all $\sigma, \lambda > 0$ (Feragen et al., 2015). Since we restrict \mathbf{p}, \mathbf{q} to be elements of \mathbb{P}_f , we can evaluate in $k(\mathbf{p}, \mathbf{q})$ in $\mathcal{O}(L \times A)$ time by rewriting Eq. (3) as:

$$r^2(\mathbf{p}, \mathbf{q}) = 1 - \prod_{l=1}^L \sum_{a=1}^A \sqrt{\mathbf{p}[l, a] \mathbf{q}[l, a]}.$$

For the proof see Proposition 3 in the supplementary.

This kernel will not be useful to guide optimization which becomes apparent when considering the Hellinger distance for any distinct sequences \mathbf{x}, \mathbf{x}' : $r(\mathbb{1}_{\mathbf{x}}, \mathbb{1}_{\mathbf{x}'}) = 1$. However, very often there exists a *prior ranking over elements of \mathbb{X}* in form of a probability distribution. For example, sequence alignment algorithms used to search biological databases often return a hidden Markov model, or a variational autoencoder can be used to compute likelihoods (Durbin et al., 1998; Riesselman et al., 2018; Frazer et al., 2021). Such models are trained in an unsupervised way and can correlate sufficiently with f (Frazer et al., 2021). To make use of this prior knowledge, we propose to weigh the Hellinger distance using the given ranking. For every positive weighting $w : \mathbb{X} \rightarrow \mathbb{R}_+$, we define the weighted Hellinger distance as

$$r_w(\mathbf{p}, \mathbf{q}) := \frac{1}{2} \sum_{x \in \mathbb{X}} w(x) \left(\sqrt{p(x)} - \sqrt{q(x)} \right)^2.$$

The proof that r_w still gives rise to a kernel function can be found in the supplementary material (Proposition 2). When considering distinct sequences \mathbf{x}, \mathbf{x}' , the weighted Hellinger distance evaluates to $r_w(\mathbb{1}_{\mathbf{x}}, \mathbb{1}_{\mathbf{x}'}) = \frac{1}{2}w(\mathbf{x}) + w(\mathbf{x}')$. This means that sequences which have low weighting are considered similar whereas sequences with high weighting are considered more independent.

The kernel is particularly suitable to model functions that have a cut-off or where the majority of inputs yields a particular value. If the weighting is 0 for sequences that have the same function-value, they correlate perfectly under this kernel, allowing us to disregard this vast space with one function evaluation in our BO routine.

When choosing w it is important to allow for efficient evaluation² of r_w . In the following we will consider two options: i) hidden Markov models (Durbin et al., 1998) and ii) variational autoencoders (VAEs) (Kingma & Welling, 2019). Fig. 2 shows a visualization of how the combination of Hellinger kernel and decoder induces a more complex, non-Euclidean similarity measure on the latent space.

3.2.2. MODELING DETAILS

This seemingly simplistic kernel gives rise to expressive Gaussian process models by recalling that products of kernels define a valid kernel (Rasmussen & Williams, 2006, p. 95). More specifically, for a given latent variable model we propose to use $k'(x_A, x_B) := \prod_{n=1}^N k_{P(X|Z_n)}(x_A, x_B)$ with Z_n being subsets of the latent space and the same hyper-parameters θ and λ for each kernel. This is a product kernel over the available samples. We defer specifying our choices for Z_n to Section 4.

Regarding the hyper-parameters, we follow Jones et al. (1998) and set a constant prior mean function with $\mu := \frac{\mathbf{1}^\top \mathbf{K}^{-1} \mathbf{y}}{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}$ and the amplitude of the kernel to $\theta := \frac{1}{N-1} (\mathbf{y} - \mu)^\top \mathbf{K}^{-1} (\mathbf{y} - \mu)$. The other parameters, which are λ and the observation noise σ^2 , we set by maximizing the evidence $\log p(\mathbf{y} | \lambda, \sigma^2)$.

3.3. Optimizing the acquisition function

As a consequence of the continuous relaxation, the acquisition function also acts on the probability distributions. Essentially, there are three ways to find local optima: (i) discrete optimization algorithms, (ii) using a continuous parameterization of \mathbb{P}_f , (iii) manifold optimization on \mathbb{P}_f , and (iv) any combination of the former options.

Note that our acquisition function still allows us to rate sequences individually by treating them as indicator functions. Therefore, any discrete optimization algorithm applies, which we use later to compare with existing discrete optimizers.

Another option for optimization arises if we have a continuous parameterization for \mathbb{P}_f , for example given through the decoder of a VAE $\text{Dec} : \mathbb{R}^d \rightarrow \mathbb{P}_f$. In that case, we can apply standard continuous optimization algorithms to $\beta : \mathbb{R}^d \rightarrow \mathbb{R}, \mathbf{z} \mapsto \alpha(\text{Dec}[\mathbf{z}])$.

²Note, w can be of different form than elements of \mathbb{P}_f .

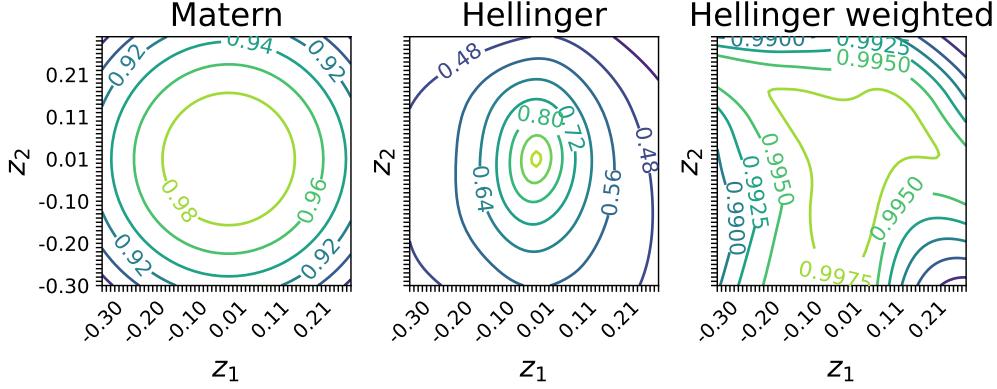


Figure 2: Visual comparison of the (weighted) Hellinger distance kernel and a Matérn 5/2. We are using a two-dimensional version of the decoder proposed by Brookes et al. (2019). For the Matérn kernel we are visualizing $k(0, [z_1, z_2])$ whereas for the Hellinger kernel we show $k(P(x|0), P(x'|[z_1, z_2]))$. With the Hellinger kernel, the decoder induces a much more complex, non-Euclidean similarity measure on the latent space. Note that, the Hellinger kernels are non-stationary in the latent space—Fig. 7 in the supplementary displays the same visualization for a different reference point.

In the ideal case, the optimization stops with an atomic distribution³ such that the decision which sequence to evaluate is unambiguous. Even if this is not the case, the optimization helps to narrow down the choice of candidates. To decide on a final candidate, we may choose the most likely sequence or we can sample from the optimized distribution and use the same acquisition function to score the sampled candidates. The latter option makes our approach applicable for a batched optimization setting.

A third option to optimize the acquisition function, we mention here in passing, since its implications are out of the scope of this work. The space \mathbb{P}_f is the product simplex, a manifold. Hence, we can apply any manifold optimization algorithm directly on the acquisition function (Boumal, 2014).

3.4. The CoRel algorithms

Algorithm 1 shows how CoRel can be used in the general Bayesian optimization loop given a continuous parameterization of \mathbb{P}_f , here the decoder of a variational autoencoder. See Algorithm 3 and Algorithm 4 in the supplementary material for the conceptual difference when using discrete optimization or direct continuous optimization.

Given is a set of starting sequences with observations, the acquisition function on the \mathbb{P}_f space, and a pretrained model with which we derive a probability vector for each sequence. We fit a GP posterior predictive optimizing the kernel parameters. The acquisition function is queried for the maximizing point on the parameterized space via the predictive GP to find the maximizing probability vector. From the probabil-

ity vector we can obtain a sequence or a set of sequences within a given budget. The black-box function is evaluated on each sequence and the observations as well as the inputs are added to the dataset. These steps are repeated until the budget is exhausted.

Algorithm 1 CoRel using parameterized optimization

Input: acquisition $a : \mathbb{P}_f \rightarrow \mathbb{R}$, black-box $f : \mathbb{X} \rightarrow \mathbb{R}$, dataset $\mathcal{D}_1 = \{X, y\}$, pretrained LVM $\phi : \mathbb{R}^D \rightarrow \mathbb{P}_f$

for $t \in 1, \dots, t_{\max}$ **do**

- $m \leftarrow \text{trainModel}((\mathbb{1}_{x_i}, y_i)_{i=1}^t)$
- $z_* \leftarrow \arg \max_z a(\phi(z), m)$
- $p_* \leftarrow \phi(z_*)$
- $x \leftarrow \text{getSequenceFromDistribution}(p_*)$
- $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{x, f(x)\}$

end for

Algorithm 2 getSequenceFromDistribution

Input: distribution P , acquisition function a , budget b

Output: x_*

$x_* = \arg \max_x P(x)$

$y_* = a(\mathbb{1}_{x_*})$

for $t \in 1, \dots, b$ **do**

- $x \sim P$
- $y = a(\mathbb{1}_x)$
- if** $y > y_*$ **then**

 - $y_* = y, x_* = x$

- end if**

end for

Now that we have a complete formulation of the proposed Bayesian optimization procedure we assess the results empirically.

³A deterministic distribution in which one outcome has a likelihood of one and all others zero.

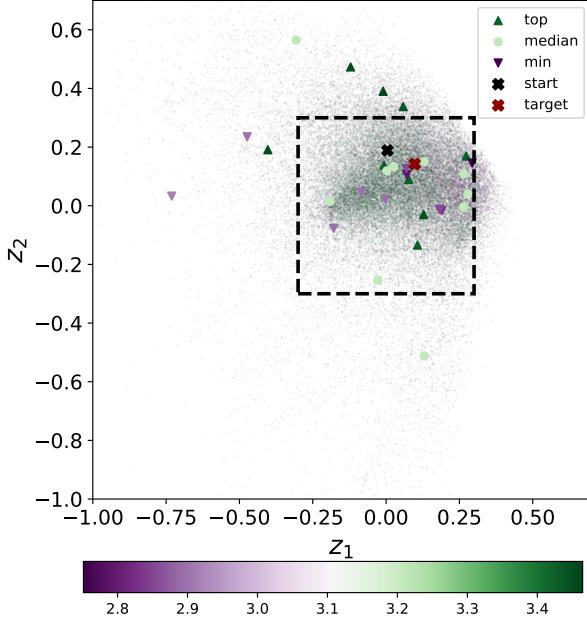


Figure 3: Two dimensional VAE latent space adapted from (Brookes et al., 2019). We encode the GFP corpus of experimentally evaluated sequences (dots). Available for optimization are only oracle evaluations - see markers for oracle predictions (top \blacktriangle , median \bullet , and lowest \blacktriangledown 10 observations each). Start is the reference wild-type and target the maximally fluorescent candidate.

4. Empirical Results

Two problems serve as an empirical benchmark of our method: (i) Stanton et al. (2022) optimized several *Red Fluorescent Proteins* (RFP) with *LamBO* and (ii) Brookes et al. (2019) defined a model to optimize *enhanced Green Fluorescent Protein* (eGFP) with *CBas*. We optimize both problems with CoRel in the *ice-cold* setup, where only the candidates are available that strictly define the problem; specifically 6 RFP and 3 GFP sequences. This means that only the observations for initial candidate sequences are available which are to be optimized and no other measurements can be obtained initially. A setup particularly relevant for drug-discovery where initial experimental observations can be prohibitively expensive (Dubach et al., 2017).

In order to systematically test the protein sequences for their properties under different experimental setups we provide poli – a dedicated protein optimization library to query black box functions with discrete sequence inputs.

4.1. Continuous optimization

Optimizing with a Latent Variable Model (LVM) is arguably the most appealing approach for CoRel as a continuous optimizer. In this section we investigate the GFP

problem for which a continuous parameterization of \mathbb{P}_f already exists as a pretrained latent decoder (Brookes et al., 2019) (see Supplementary Appendix E for details). Figure 3 shows an adaptation of the GFP VAE as a learned two-dimensional embedding, for which only an oracle function can be queried (see markers) and experimental observations are unavailable. The CBas oracle function evaluations serve as surrogate for the true GFP fluorescence values. To qualitatively inspect our method we evaluate the covariance function values of the (2D) latent space in an area around the reference sequence. Fig. 2 shows the evaluated Hellinger (Eq. (3)) and weighted Hellinger kernel, which are *not* equidistant in latent space (like the Matérn) respective a reference point while the decoder indeed informed the covariance function values. These values change as anticipated when computed with respect to a different reference point (see Supplementary Fig. 7). In particular higher covariance values are assigned in a density around the reference points and the respective decoding probabilities.

When conducting optimization we observe larger objective values within the allotted budget (100 queries) compared to a random acquisition while the model appears to prioritize extreme values in the problem-space (see Supplementary Fig. 11).

4.2. Discrete optimization

Optimizing in the discrete proposed sequence setting is what contemporary optimization protocols often translate to if the generated proposals are directly in the sequence space. Therefore we show that our approach is still applicable to the discrete setting. For this we optimize the RFP problem with respect to multiple properties: stability and surface area accessibility (SASA). We use LamBO as a SoTA reference in this optimization setting and compare also against random sequence mutations. Again we focus in particular on a strict setup where only the Pareto front is given (6 reference RFP sequences) and the number of oracle evaluations is limited to 180 queries specifically. To optimize for multiple tasks in the BO algorithm we use the expected hypervolume improvement (EHVI) as an acquisition function (Daulton et al., 2020).

To build the \mathbb{P}_f -space we rely *only* on a Hidden Markov Model (HMM), obtained by the standard HMMER algorithm (Eddy, 2011). We use this as ϕ model to parameterize our distributions. The choice for an HMM is natural, as it is built in the process of querying for related sequences to the starting candidates, a step that would be required when setting up the initial RFP problem-set (Stanton et al., 2022)⁴.

Fig. 5 shows that running CoRel obtains a larger relative

⁴The LamBO RFP data includes a wide range of additional RFP sequences (see Supplementary Appendix E).

hypervolume compared to SoTA method *LamBO*, and a simple random mutation baseline which modifies 2 amino acids selected at random from elements in the Pareto front and retains the best results for subsequent iterations. This results in a larger Pareto front of the respective protein candidates (see Fig. 6).

Our results also show that we achieve on-par performance in the reference setup where we have just over 500 of initial sequences available (see Supplementary Fig. 9). Under those conditions we outperform existing methods with respect to the achieved (relative) hyper-volume within the given budget - see Fig. 5. However we note that if multiple starting candidates are available (*i.e.* $N = 50$) a larger Pareto-front is optimized and *LamBO* does outperform our method.

5. Discussion

We focus on the surrogate model rather than the acquisition function Advances in Bayesian optimization can either be done by building a suitable surrogate function to model the unknown true f or by investigating the acquisition function. Our contribution is apart from the problem transformation primarily the surrogate model via the covariance function. CoRel focuses on the properties of the surrogate function - in particular, a GP which incorporates a ranking over the continuously relaxed inputs. This yields a surrogate model predicting points of interest to observe, at the cost of capturing the underlying function landscape. While the acquisition function is an interesting topic of research in its own right it is out of the scope of this work and we rely on the established results in the field (Jones et al., 1998).

Relying on established prior models can alleviate model complexity Our setup of continuously relaxing and using a pre-computed model such as HMMs parameterizing our constrained probability space is different compared to contemporary latent space optimizations. Given that distributions or a parameterization of elements of a sequence are available for a particular problem-domain that have been shown to work reliably, they can be used directly in the CoRel approach. This alleviates to some extent to account for a plethora of potential decisions which are required for latent-space BO approaches: which encodings to use, which surrogate model to train given the trained latent representation, the sampling and reweighing of base and proposal sequences, etc. .

Simple prior models may impact performance The results show that we perform SoTA in particular in the cold-start setting, restricted domain - relevant for BO on biochemical sequences. The performance in less restrictive conditions can be improved by tuning the models which parameterize the densities. The contributions of this paper

are mainly the problem transformation and its application to sequence optimization. Improving upon the pretrained models for which the weighted Hellinger kernel can be applied will be left for future work.

6. Conclusion

We have shown Bayesian optimization by continuous relaxation with the proposed CoRel approach, namely a continuous relaxation of the objective and a Gaussian process acting on sequences as probability distributions. We are able to do optimization with the introduced weighted Hellinger kernel after having shown that it is indeed a kernel. We have demonstrated the applicability in the bio-chemical domain and compared empirically with contemporary methods by optimizing various protein sequences. Here we have demonstrated feasibility and performance particularly in significantly more challenging formulations of the optimization problems.

7. Software and Data

We provide the code for CoRel on GitHub as well as the *poli* package on, both under the MIT license.

Acknowledgements

This work was in part funded by the Danish Data Science Academy, which is funded by the Novo Nordisk Foundation (NNF21SA0069429) and VILLUM FONDEN (40516). Further funding includes the MLLS Center (Basic Machine Learning Research in Life Science NNF20OC0062606), the Danish Ministry of Education and Science, Digital Pilot Hub, Skylab Digital, and the Pioneer Centre for AI (DRNF grant number P1).

References

- Arvanitidis, G., Hansen, L. K., and Hauberg, S. Latent Space Oddity: on the Curvature of Deep Generative Models, December 2021. URL <http://arxiv.org/abs/1710.11379>. arXiv:1710.11379 [stat].
- Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M., and Church, G. M. Low-N protein engineering with data-efficient deep learning. *Nature Methods*, 18(4):389–396, April 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01100-y. URL <https://www.nature.com/articles/s41592-021-01100-y>. Number: 4 Publisher: Nature Publishing Group.
- Boumal, N. Optimization and estimation on manifolds. 2014. Publisher: Catholic University of Louvain, Louvain-la-Neuve, Belgium.

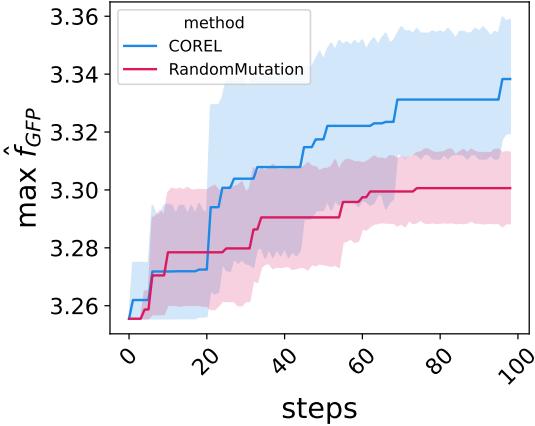


Figure 4: The GFP sequences are optimized by CoRel and random mutations with the CBas oracle (\hat{f}_{GFP}) over 100 steps. We observe best sequences selected at an iteration with mean (line) and 95%CI (shaded) across seven seeds.

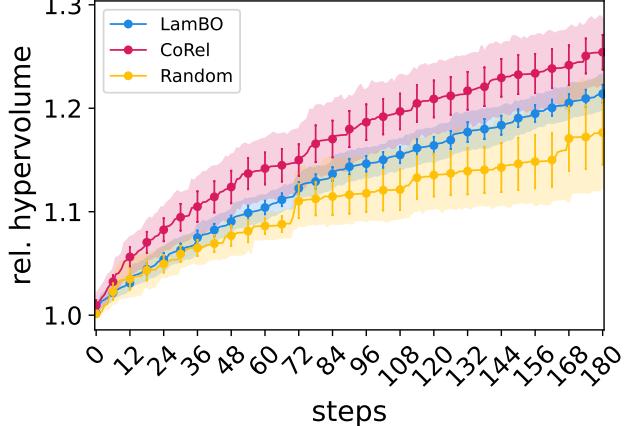


Figure 5: The RFP Pareto front is optimized discretely computing relative hypervolume respective the starting sequences, comparing CoRel with LamBO starting with six observations. Markers (●) indicate batch average and std.err. bars across 21 seeds (random 5).

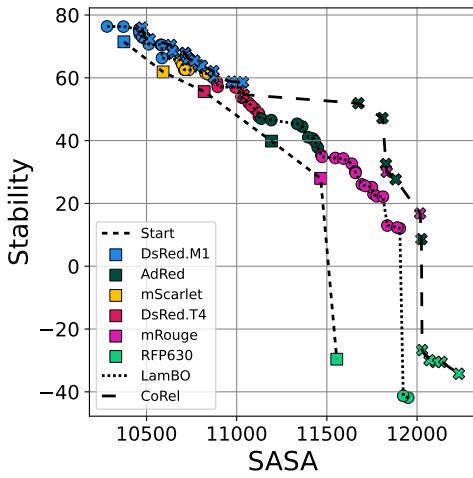


Figure 6: Pareto front of the RFP properties stability and surface accessibility (SASA). The *start* Pareto sequences (■ color-coded), the best variants proposed by LamBO (●) and CoRel (✖) after 32 iterations (21 seeds) from a cold start.

Brookes, D., Park, H., and Listgarten, J. Conditioning by adaptive sampling for robust design. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 773–782. PMLR, May 2019. URL <https://proceedings.mlr.press/v97/brookes19a.html>. ISSN: 2640-3498.

Daulton, S., Balandat, M., and Bakshy, E. Differentiable Expected Hypervolume Improvement for Parallel Multi-

Objective Bayesian Optimization. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9851–9864. Curran Associates, Inc., 2020.

Daulton, S., Wan, X., Eriksson, D., Balandat, M., Osborne, M. A., and Bakshy, E. Bayesian Optimization over Discrete and Mixed Spaces via Probabilistic Reparameterization, October 2022. URL <http://arxiv.org/abs/2210.10199>. arXiv:2210.10199 [cs, math, stat].

Detlefsen, N. S., Hauberg, S., and Boomsma, W. Learning meaningful representations of protein sequences. *Nature Communications*, 13(1):1914, April 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-29443-w. URL <https://www.nature.com/articles/s41467-022-29443-w>. Number: 1 Publisher: Nature Publishing Group.

Dubach, J. M., Kim, E., Yang, K., Cuccarese, M., Giedt, R. J., Meimetis, L. G., Vinegoni, C., and Weissleder, R. Quantitating drug-target engagement in single cells in vitro and in vivo. *Nature chemical biology*, 13(2):168–173, 2017. Publisher: Nature Publishing Group US New York.

Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1 edition, April 1998. ISBN 978-0-521-62041-3 978-0-521-62971-3 978-0-511-79049-2. doi: 10.1017/CBO9780511790492. URL <https://www.cambridge.org/core/product/identifier/9780511790492/type/book>.

- Eddy, S. R. Accelerated Profile HMM Searches. *PLOS Computational Biology*, 7(10):e1002195, October 2011. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1002195. URL <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002195>. Publisher: Public Library of Science.
- Feragen, A., Lauze, F., and Hauberg, S. Geodesic Exponential Kernels: When Curvature and Linearity Conflict. pp. 3032–3042, 2015. URL https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Feragen_Geodesic_Exponential_Kernels_2015_CVPR_paper.html.
- Frazer, J., Notin, P., Dias, M., Gomez, A., Min, J. K., Brock, K., Gal, Y., and Marks, D. S. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, November 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-04043-8. URL <https://www.nature.com/articles/s41586-021-04043-8>. Number: 7883 Publisher: Nature Publishing Group.
- Gao, W., Fu, T., Sun, J., and Coley, C. W. Sample Efficiency Matters: A Benchmark for Practical Molecular Optimization, October 2022. URL <http://arxiv.org/abs/2206.12411>. arXiv:2206.12411 [cs, q-bio].
- Garnett, R. *Bayesian Optimization*. Cambridge University Press, January 2022.
- Garrido-Merchán, E. C. and Hernández-Lobato, D. Dealing with categorical and integer-valued variables in Bayesian Optimization with Gaussian processes. *Neurocomputing*, 380:20–35, March 2020. ISSN 0925-2312. doi: 10.1016/j.neucom.2019.11.004. URL <https://www.sciencedirect.com/science/article/pii/S0925231219315619>.
- Gruver, N., Stanton, S., Frey, N. C., Rudner, T. G. J., Hotzel, I., Lafrance-Vanassee, J., Rajpal, A., Cho, K., and Wilson, A. G. Protein Design with Guided Discrete Diffusion, May 2023. URL <http://arxiv.org/abs/2305.20009>. arXiv:2305.20009 [cs, q-bio].
- Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, 4(2):268–276, February 2018. ISSN 2374-7943. doi: 10.1021/acscentsci.7b00572. URL <https://doi.org/10.1021/acscentsci.7b00572>. Publisher: American Chemical Society.
- Harandi, M., Salzmann, M., and Baktashmotagh, M. Beyond Gauss: Image-Set Matching on the Riemannian Manifold of PDFs. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- Hellinger, E. Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136):210–271, July 1909. ISSN 1435-5345. doi: 10.1515/crll.1909.136.210. URL <https://www.degruyter.com/document/doi/10.1515/crll.1909.136.210/html>. Publisher: De Gruyter.
- Jones, D. R., Schonlau, M., and Welch, W. J. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492, December 1998. ISSN 1573-2916. doi: 10.1023/A:1008306431147. URL <https://doi.org/10.1023/A:1008306431147>.
- Kingma, D. P. and Welling, M. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000056. URL <http://arxiv.org/abs/1906.02691>. arXiv:1906.02691 [cs, stat].
- Lu, X., Gonzalez, J., Dai, Z., and Lawrence, N. D. Structured Variationally Auto-encoded Optimization. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 3267–3275. PMLR, July 2018. URL <https://proceedings.mlr.press/v80/lu18c.html>. ISSN: 2640-3498.
- Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL <https://www.tensorflow.org/>.
- Matthews, A. G. d. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, April 2017. URL <http://jmlr.org/papers/v18/16-537.html>.

- Maus, N., Wu, K., Eriksson, D., and Gardner, J. Discovering Many Diverse Solutions with Bayesian Optimization, May 2023. URL <http://arxiv.org/abs/2210.10953> [cs].
- Maynard Smith, J. Natural Selection and the Concept of a Protein Space. *Nature*, 225(5232):563–564, February 1970. ISSN 1476-4687. doi: 10.1038/225563a0. URL <https://www.nature.com/articles/225563a0>. Number: 5232 Publisher: Nature Publishing Group.
- Moss, H., Leslie, D., Beck, D., Gonzalez, J., and Rayson, P. Boss: Bayesian optimization over string spaces. *Advances in neural information processing systems*, 33: 15476–15486, 2020.
- Močkus, J. On Bayesian Methods for Seeking the Extremum. In Marchuk, G. I. (ed.), *Optimization Techniques IFIP Technical Conference: Novosibirsk, July 1–7, 1974*, Lecture Notes in Computer Science, pp. 400–404. Springer, Berlin, Heidelberg, 1975. ISBN 978-3-662-38527-2. doi: 10.1007/978-3-662-38527-2_55. URL https://doi.org/10.1007/978-3-662-38527-2_55.
- Notin, P. M., Van Niekerk, L., Kollasch, A. W., Ritter, D., Gal, Y., and Marks, D. TranceptEVE: Combining Family-specific and Family-agnostic Models of Protein Sequences for Improved Fitness Prediction. preprint, Genetics, December 2022. URL <http://biorxiv.org/lookup/doi/10.1101/2022.12.07.519495>.
- Picheny, V., Berkeley, J., Moss, H. B., Stojic, H., Granta, U., Ober, S. W., Artemev, A., Ghani, K., Goodall, A., Paleyes, A., Vakili, S., Pascual-Diaz, S., Markou, S., Qing, J., Loka, N. R. B. S., and Couckuyt, I. Trieste: Efficiently Exploring The Depths of Black-box Functions with TensorFlow, 2023. URL <https://arxiv.org/abs/2302.08436>.
- Rasmussen, C. E. and Williams, C. K. I. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2006. ISBN 978-0-262-18253-9. OCLC: ocm61285753.
- Riesselman, A. J., Ingraham, J. B., and Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10):816–822, October 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0138-4. URL <https://www.nature.com/articles/s41592-018-0138-4>. Number: 10 Publisher: Nature Publishing Group.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175, January 2016. ISSN 0018-9219, 1558-2256. doi: 10.1109/JPROC.2015.2494218. URL <https://ieeexplore.ieee.org/document/7352306/>.
- Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. *IEEE Transactions on Information Theory*, 58(5):3250–3265, May 2012. ISSN 0018-9448, 1557-9654. doi: 10.1109/TIT.2011.2182033. URL <https://arxiv.org/abs/0912.3995> [cs].
- Stanton, S., Maddox, W., Gruver, N., Maffettone, P., Delaney, E., Greenside, P., and Wilson, A. G. Accelerating Bayesian Optimization for Biological Sequence Design with Denoising Autoencoders. Technical Report arXiv:2203.12742, arXiv, July 2022. URL <https://arxiv.org/abs/2203.12742>. arXiv:2203.12742 [cs, q-bio, stat] type: article.
- Swersky, K., Snoek, J., and Adams, R. P. Multi-Task Bayesian Optimization. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL <https://proceedings.neurips.cc/paper/2013/hash/f33ba15effa5c10e873bf3842afb46a6-Abstract.html>.
- Swersky, K., Rubanova, Y., Dohan, D., and Murphy, K. Amortized Bayesian Optimization over Discrete Spaces. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 769–778. PMLR, August 2020. URL <https://proceedings.mlr.press/v124/swersky20a.html>. ISSN: 2640-3498.
- Tian, P. and Best, R. B. How Many Protein Sequences Fold to a Given Structure? A Coevolutionary Analysis. *Biophysical Journal*, 113(8):1719–1730, October 2017. ISSN 00063495. doi: 10.1016/j.bpj.2017.08.039. URL <https://linkinghub.elsevier.com/retrieve/pii/S0006349517309347>.
- Tripp, A., Daxberger, E., and Hernández-Lobato, J. M. Sample-Efficient Optimization in the Latent Space of Deep Generative Models via Weighted Retraining. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11259–11272. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/hash/81e3225c6ad49623167a4309eb4b2e75-Abstract.html>.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M.,

Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, I., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Wan, X., Nguyen, V., Ha, H., Ru, B., Lu, C., and Osborne, M. A. Think Global and Act Local: Bayesian Optimisation over High-Dimensional Categorical and Mixed Search Spaces. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 10663–10674. PMLR, July 2021. URL <https://proceedings.mlr.press/v139/wan21b.html>. ISSN: 2640-3498.

A. Proof that the relaxed objective has the same optima

Proposition 1. For f and \bar{f} : $\arg \min f = \arg \min \bar{f}$.

Proof.

$$\begin{aligned}\bar{f}(\mathbf{p}) &= \sum_{\mathbf{x}} f(\mathbf{x}) p(\mathbf{x}) \\ &\leq \sum_{\mathbf{x}} \max_{\mathbf{x}'} f(\mathbf{x}') p(\mathbf{x}) \\ &= \max_{\mathbf{x}'} f(\mathbf{x}') \sum_{\mathbf{x}} p(\mathbf{x}) \\ &= \max_{\mathbf{x}'} f(\mathbf{x}')\end{aligned}$$

On the other hand: for an optimal \mathbf{x}_* , i.e. $f(\mathbf{x}_*) = \max_{\mathbf{x}'} f(\mathbf{x}')$ and choose $\mathbf{p}_*(\mathbf{x}) := \mathbf{1}_{\mathbf{x}=\mathbf{x}_*}(\mathbf{x})$, then $\bar{f}(\mathbf{p}_*) = \sum_{\mathbf{x}} f(\mathbf{x}) p_*(\mathbf{x}) f(\mathbf{x}) = \max_{\mathbf{x}'} f(\mathbf{x}')$. \square

B. The weighted Hellinger distance

Proposition 2. The squared weighted Hellinger distance

$$HD_{\mathbf{p}}(\mathbf{q}, \mathbf{r})^2 := \frac{1}{2} \sum_{\mathbf{x}} \mathbf{p}(\mathbf{x}) \left(\sqrt{\mathbf{q}(\mathbf{x})} - \sqrt{\mathbf{r}(\mathbf{x})} \right)^2$$

is negative definite.

Proof. The proof follows Harandi et al. (2015). Let $N \in \mathbb{N}$ and $c_1, \dots, c_N \in \mathbb{R}$ s.t. $\sum_{n=1}^N c_n = 0$.

$$\sum_{n,m=1}^N c_n c_m HD_{\mathbf{p}}(\mathbf{q}_n, \mathbf{q}_m)^2 \tag{5}$$

$$= \frac{1}{2} \sum_{n,m=1}^N c_n c_m \sum_{\mathbf{x} \in \mathbb{X}} \mathbf{p}(\mathbf{x}) \left(\sqrt{\mathbf{q}_n(\mathbf{x})} - \sqrt{\mathbf{q}_m(\mathbf{x})} \right)^2 \tag{6}$$

// by definition

$$= \frac{1}{2} \sum_{n,m=1}^N c_n c_m \sum_{\mathbf{x} \in \mathbb{X}} \mathbf{p}(\mathbf{x}) \left(\mathbf{q}_n(\mathbf{x}) + \mathbf{q}_m(\mathbf{x}) - 2\sqrt{\mathbf{q}_n(\mathbf{x})\mathbf{q}_m(\mathbf{x})} \right) \tag{7}$$

// expanding the square

$$= \sum_{\mathbf{x} \in \mathbb{X}} \mathbf{p}(\mathbf{x}) \left(\sum_{n=1}^N c_n \mathbf{q}_n(\mathbf{x}) \sum_{m=1}^N c_m + \sum_{m=1}^N c_m \mathbf{q}_m(\mathbf{x}) \sum_{n=1}^N c_n - \sum_{n=1}^N c_n \sqrt{\mathbf{q}_n(\mathbf{x})} \sum_{m=1}^N c_m \sqrt{\mathbf{q}_m(\mathbf{x})} \right) \tag{8}$$

// changing order of summation

$$= - \sum_{\mathbf{x} \in \mathbb{X}} \mathbf{p}(\mathbf{x}) \sum_{n=1}^N c_n \sqrt{\mathbf{q}_n(\mathbf{x})} \sum_{m=1}^N c_m \sqrt{\mathbf{q}_m(\mathbf{x})} \tag{9}$$

// since $\sum_{n=1}^N c_n = 0$

$$= - \sum_{\mathbf{x} \in \mathbb{X}} \mathbf{p}(\mathbf{x}) \left(\sum_{n=1}^N c_n \sqrt{\mathbf{q}_n(\mathbf{x})} \right)^2 \tag{10}$$

// writing the identical sums over n and m as a square

$$\leq 0 \tag{11}$$

□

To show that the square-root of the tilted Hellinger distance is a kernel, we follow the same reasoning as in Harandi et al. (2015).

C. Efficient evaluation of the Hellinger distance

Proposition 3. *For product measures $\mathbf{p}, \mathbf{q} \in \mathbb{P}_f$, the Hellinger distance can be written as*

$$HD(\mathbf{p}, \mathbf{q})^2 = 1 - \prod_{l=1}^L \sum_{a_l=1}^A \sqrt{\mathbf{p}[a_l, l] \mathbf{q}[a_l, l]}$$

Proof.

$$HD(\mathbf{p}, \mathbf{q})^2 = \frac{1}{2} \sum_{\mathbf{x} \in \mathbb{X}} \left(\sqrt{\mathbf{p}(\mathbf{x})} - \sqrt{\mathbf{q}(\mathbf{x})} \right)^2 \quad (12)$$

// expanding the square and using that \mathbf{p} and \mathbf{q} sum to 1.

$$= 1 - \sum_{\mathbf{x} \in \mathbb{X}} \sqrt{\mathbf{p}(\mathbf{x}) \mathbf{q}(\mathbf{x})} \quad (13)$$

// property of the Hellinger distance

$$= 1 - \underbrace{\sum_{a_1=1}^A \cdots \sum_{a_L=1}^A}_{L \text{ times}} \sqrt{\mathbf{p}(a_1, \dots, a_L) \mathbf{q}(a_1, \dots, a_L)} \quad (14)$$

// rewriting the sum

$$= 1 - \underbrace{\sum_{a_1=1}^A \cdots \sum_{a_L=1}^A}_{L \text{ times}} \sqrt{\prod_{l=1}^L \mathbf{p}[a_l, l] \mathbf{q}[a_l, l]} \quad (15)$$

// using $\mathbf{p}, \mathbf{q} \in \mathbb{P}_f$

$$= 1 - \underbrace{\sum_{a_1=1}^A \cdots \sum_{a_L=1}^A}_{L \text{ times}} \prod_{l=1}^L \sqrt{\mathbf{p}[a_l, l] \mathbf{q}[a_l, l]} \quad (16)$$

// moving the square-root

$$= 1 - \sum_{a_1=1}^A \sqrt{\mathbf{p}[a_1, 1] \mathbf{q}[a_1, 1]} \cdot \dots \cdot \sum_{a_L=1}^A \sqrt{\mathbf{p}[a_L, L] \mathbf{q}[a_L, L]} \quad (17)$$

// moving unaffected parts of the product out of the sum

$$= 1 - \prod_{l=1}^L \sum_{a_l=1}^A \sqrt{\mathbf{p}[a_l, l] \mathbf{q}[a_l, l]} \quad (18)$$

// rearranging

□

Proposition 4. For product measures $\mathbf{p}, \mathbf{q}, \mathbf{r} \in \mathbb{P}_f$, the weighted Hellinger distance can be written as

$$\begin{aligned} HD(\mathbf{p}, \mathbf{q}, \mathbf{r})^2 &= \prod_{l=1}^L \sum_{a_l=1}^A \left[\frac{1}{2} \mathbf{r}[a_l, l] \mathbf{p}[a_l, l] + \frac{1}{2} \mathbf{r}[a_l, l] \mathbf{q}[a_l, l] \right] - \prod_{l=1}^L \sum_{a_l=1}^A \mathbf{r}[a_l, l] \sqrt{\mathbf{p}[a_l, l] \mathbf{q}[a_l, l]} \\ &= \prod_{l=1}^L \sum_{a_l=1}^A \frac{1}{2} \mathbb{E}_{\mathbf{p}}[\mathbf{x}] \mathbf{r}[a_l, l] + \frac{1}{2} \mathbb{E}_{\mathbf{q}}[\mathbf{x}] \mathbf{r}[a_l, l] - \prod_{l=1}^L \sum_{a_l=1}^A \mathbf{r}[a_l, l] \sqrt{\mathbf{p}[a_l, l] \mathbf{q}[a_l, l]}. \end{aligned}$$

Proof.

$$HD_{\mathbf{r}}(\mathbf{p}, \mathbf{q})^2 = \frac{1}{2} \sum_{\mathbf{x} \in \mathbb{X}} r(\mathbf{x}) \left(\sqrt{\mathbf{p}(\mathbf{x})} - \sqrt{\mathbf{q}(\mathbf{x})} \right)^2 \quad (19)$$

$$= \frac{1}{2} \sum_{\mathbf{x} \in \mathbb{X}} r(\mathbf{x}) \left(\mathbf{p}(\mathbf{x}) - 2\sqrt{\mathbf{p}(\mathbf{x})\mathbf{q}(\mathbf{x})} + \mathbf{q}(\mathbf{x}) \right) \quad (20)$$

// expanding the square

$$= \frac{1}{2} \sum_{\mathbf{x} \in \mathbb{X}} \left[\mathbf{r}(\mathbf{x}) \mathbf{p}(\mathbf{x}) - 2\mathbf{r}(\mathbf{x}) \sqrt{\mathbf{p}(\mathbf{x})\mathbf{q}(\mathbf{x})} + \mathbf{r}(\mathbf{x}) \mathbf{q}(\mathbf{x}) \right] \quad (21)$$

// note that $\sum_{\mathbf{x} \in \mathbb{X}} \mathbf{r}(\mathbf{x}) \mathbf{p}(\mathbf{x}) \neq 1$

$$= \frac{1}{2} \sum_{\mathbf{x} \in \mathbb{X}} \mathbf{r}(\mathbf{x}) \mathbf{p}(\mathbf{x}) + \frac{1}{2} \sum_{\mathbf{x} \in \mathbb{X}} \mathbf{r}(\mathbf{x}) \mathbf{q}(\mathbf{x}) - \sum_{\mathbf{x} \in \mathbb{X}} \mathbf{r}(\mathbf{x}) \sqrt{\mathbf{p}(\mathbf{x})\mathbf{q}(\mathbf{x})} \quad (22)$$

// property of the Hellinger distance

$$= \frac{1}{2} \underbrace{\sum_{a_1=1}^A \cdots \sum_{a_L=1}^A}_{L \text{ times}} \mathbf{r}(x_1, \dots, x_L) \mathbf{p}(x_1, \dots, x_L) + \frac{1}{2} \underbrace{\sum_{a_1=1}^A \cdots \sum_{a_L=1}^A}_{L \text{ times}} \mathbf{r}(x_1, \dots, x_L) \mathbf{q}(x_1, \dots, x_L) \quad (23)$$

$$- \underbrace{\sum_{a_1=1}^A \cdots \sum_{a_L=1}^A}_{L \text{ times}} \mathbf{r}(x_1, \dots, x_L) \sqrt{\mathbf{p}(x_1, \dots, x_L)\mathbf{q}(x_1, \dots, x_L)} \quad (24)$$

// factorize

$$= \frac{1}{2} \underbrace{\sum_{a_1=1}^A \cdots \sum_{a_L=1}^A}_{L \text{ times}} \prod_{l=1}^L \mathbf{r}[a_l, l] \mathbf{p}[a_l, l] + \frac{1}{2} \underbrace{\sum_{a_1=1}^A \cdots \sum_{a_L=1}^A}_{L \text{ times}} \prod_{l=1}^L \mathbf{r}[a_l, l] \mathbf{q}[a_l, l] \quad (25)$$

$$- \underbrace{\sum_{a_1=1}^A \cdots \sum_{a_L=1}^A}_{L \text{ times}} \prod_{l=1}^L \mathbf{r}[a_l, l] \sqrt{\mathbf{p}[a_l, l]\mathbf{q}[a_l, l]} \quad (26)$$

// rearrange, and sums of products as products of sums - see C

$$= \prod_{l=1}^L \sum_{a_l=1}^A \frac{1}{2} \mathbf{r}[a_l, l] \mathbf{p}[a_l, l] + \frac{1}{2} \mathbf{r}[a_l, l] \mathbf{q}[a_l, l] - \prod_{l=1}^L \sum_{a_l=1}^A \mathbf{r}[a_l, l] \sqrt{\mathbf{p}[a_l, l]\mathbf{q}[a_l, l]} \quad (27)$$

□

D. CoRel algorithm specifications

D.1. A continuous optimization algorithm

Algorithm 3 CoRel using continuous optimization

Input: acquisition $a : \mathbb{P}_f \rightarrow \mathbb{R}$, black-box $f : \mathbb{X} \rightarrow \mathbb{R}$, dataset $\mathcal{D}_1 = \{X, y\}$, pretrained LVM $\phi : \mathbb{R}^D \rightarrow \mathbb{P}_f$

```

for  $t \in 1, \dots, t_{\max}$  do
     $m \leftarrow \text{trainModel}((\mathbb{1}_{x_i}, y_i)_{i=1}^t)$ 
     $\mathbf{p}_* \leftarrow \arg \max_{\mathbf{p}} a(\mathbf{p}, m)$ 
     $\mathbf{x} \leftarrow \text{getSequenceFromDistribution}(\mathbf{p}_*)$ 
     $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{\mathbf{x}, f(\mathbf{x})\}$ 
end for

```

D.2. A discrete optimization algorithm

Algorithm 4 CoRel using discrete optimization

Input: acquisition $a : \mathbb{P}_f \rightarrow \mathbb{R}$, black-box $f : \mathbb{X} \rightarrow \mathbb{R}$, dataset $\mathcal{D}_1 = \{X, y\}$, pretrained LVM $\phi : \mathbb{R}^D \rightarrow \mathbb{P}_f$

```

for  $t \in 1, \dots, t_{\max}$  do
     $m \leftarrow \text{trainModel}((\mathbb{1}_{x_i}, y_i)_{i=1}^t)$ 
     $\mathbf{x}' \leftarrow \arg \max_{\mathbf{x}'} a(\mathbb{1}_{x'}, m)$ 
     $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{\mathbf{x}', f(\mathbf{x}')\}$ 
end for

```

We require a function for finding pareto optimal points, given a set of all points S with $x \subset S$:

$$p_{\text{opt}}(x) := \{x \in S \mid \nexists x' \in S \text{ s.t. } x' \preceq x \wedge x' \neq x\}. \quad (28)$$

. In our experiments pareto optimal points are determined by the y vector.

E. Baseline implementations and hyperparameters

Implementation and optimization Models and experiments are implemented with Tensorflow (Martín Abadi et al., 2015), Tensorflow-probability, GPflow (Matthews et al., 2017), and Trieste (Picheny et al., 2023). Model hyperparameters are optimized using the scipy LBFGS optimizer (Virtanen et al., 2020) on the model likelihood as previously described.

Discrete biological sequence optimization library This library contains both the RFP and GFP problem as well as the stable *LamBO* implementation for experimental queries. We define the RFP problem in generally with FoldX and SASA computations, respective the LamBO defined pareto front. We additionally include a reference objective that is equivalent to the LamBO setup and includes significantly more sequences.

CBas VAE model The *GFP* problem is presented in Brookes et al. (2019) with a VAE as a latent variable model and a custom predictive GP model. The full training corpus are 54025 unique sequences for which observations are available. For the VAE: the encoder is a simple neural network (size=50 units) with ELU activation mapping to 20 latent dimensions. The decoder is a deep neural network with 3 layers of dimensions [50, 20*len(sequence), 20] with ELU and softmax activation respectively, such that we first obtain a mapping from latent space to hidden space of size number of amino acids times (aligned) sequence length and ultimately the label-encoded protein sequence. Model input are aligned GFP sequences.

Training data are 5000 sequences of 20 amino-acid tokens. The training data are the upper-quantile (by fluorescence measurement), which is approximately 9% of all available sequences. Training has been done for 100 epochs in batches of 10 using a (tensorflow2) Adam optimizer with default configuration.

The model latent space is used in combination with a predictive GP oracle model as a surrogate for *GFP*. For a specification of the predictive GP model we refer to (Brookes et al., 2019).

The optimization task for the GFP problem is the minimization of the negative oracle values, to find the global minimum - see Fig. 4. For later analysis the sign is inverted, as we ultimately intend to maximize the oracle.

We adapt this architecture to two latent dimensions for visualization purposes only. All empirical results queried against the GP utilize the initial model of full dimensionality. The remaining model specifications remain the same.

LamBO latent optimizer We use the LamBO implementation directly from the stable tagged submission branch of Stanton et al. (2022) (commit 22afec26da0b9ea1810e65f8a60ea7988c021cef Oct-2022) and refer to their exact model specifications. The only addition we make is to define a dedicated task protocol to work with the discrete sequence optimization library we provide, and associated configurations.

We highlight that the original model specification and experiments account for a large range of starting sequences including seed-sequences. Additionally, when analyzing the model and observations we find that the relative hypervolume is computed respective a hard-coded set of reference values. For our benchmarking purposes we record this value also (see Supplementary Fig. 10). In our case we obtain the start values from the black-box function evaluations and compute the relative HV with respect to that. We provide the algorithm only with an exact set of 6 RFP PDB files to use, and do not access any provided seed-sequences.

The LamBO RFP data-set contains 50 PDB files (RFP structure files), 754 related sequences, and 1923 generated proxy seed sequences.

Random Mutations As a naïve baseline, we consider mutating positions at random in the Pareto front. At each iteration, we maintain a population of 16 elements (padding with random mutations if the Pareto front is not large enough). Each of these elements is then mutated in two random positions (taking into account that sequences may have varying lengths, and never performing any inserting/deleting operations). The results provided were averaged over 5 iterations with different random seeds. The performance of this baseline is noteworthy, and the difference between it and e.g. (Stanton et al., 2022)'s NSGA-2 may be attributed to the fact that we mutate *twice* instead of *once* per iteration.

F. Additional Results

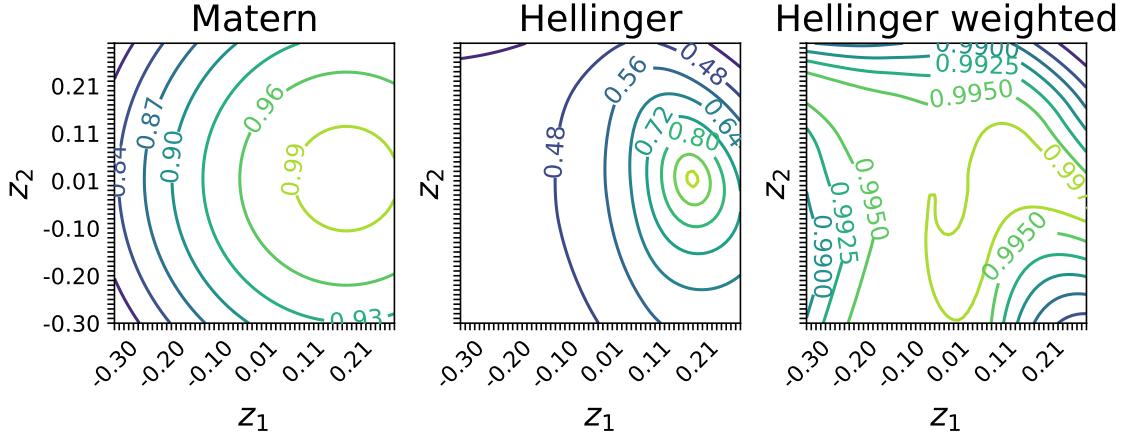


Figure 7: Covariance function values for the GFP (2D) latent space where the reference point is the GFP wild-type sequence. Comparing Matérn 5/2 with the (weighted) Hellinger kernel. The reference points corresponds to the *start* point in Fig. 3.

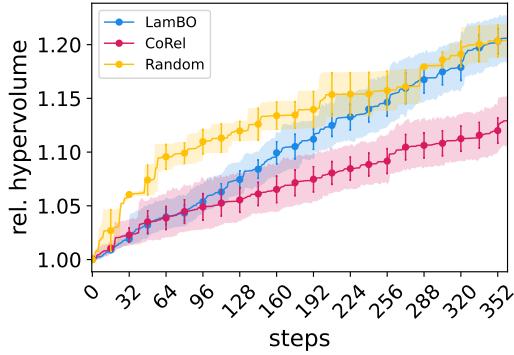


Figure 8: Discretely optimizing RFP. We compare against LamBO in the warm-start setting. Starting N=50, batch-size=16 across seven seeds (random two seeds). Markers indicate batch averages with std.err. bars. Shaded region is 95% CI.

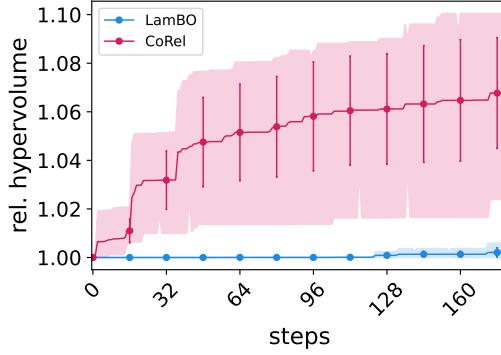


Figure 9: Optimizing RFP discretely in the reference case with 512 starting sequences. We compare CoRel against LamBO in the reference setup, batch-size=16 across three seeds. Given that we start with a relatively large starting hypervolume only marginal improvements can be achieved. Markers indicate batch averages with std.err. bars. Shaded region is 95% CI.

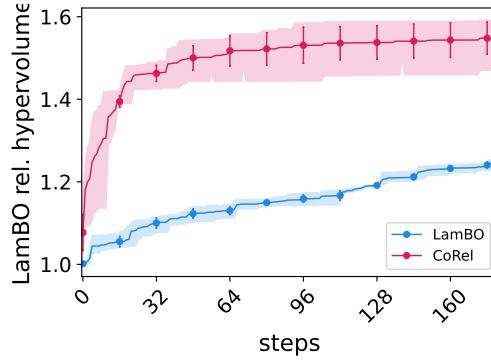


Figure 10: Optimizing RFP discretely in the reference case with 512 starting sequences using the LamBO specific relative hypervolume improvement. This metric is computed with internal reference Pareto front values, which remain fixed across all experiments and display a larger relative improvement over time. Batch-size is 16 across three seeds. Markers indicate batch averages with std.err. bars. Shaded region is 95% CI.

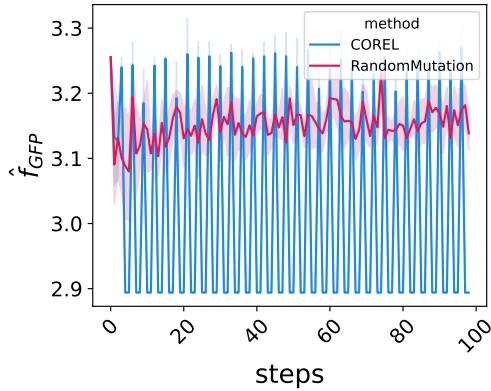


Figure 11: Oracle observations during the course of 100 GFP optimization steps. CoRel jumps between extreme values in contrast to the random exploration.