VARIATIONAL SEARCH DISTRIBUTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We develop variational search distributions (VSD), a method for finding discrete, combinatorial designs of a rare desired class in a batch sequential manner with a fixed experimental budget. We formalize the requirements and desiderata for active generation and formulate a solution via variational inference. In particular, VSD uses off-the-shelf gradient based optimization routines, can learn powerful generative models for designs, and can take advantage of scalable predictive models. We derive asymptotic convergence rates for learning the true conditional generative distribution of designs with certain configurations of our method. After illustrating the generative model on images, we empirically demonstrate that VSD can outperform existing baseline methods on a set of real sequence-design problems in various biological systems.

1 INTRODUCTION

We consider a variant of the *active search* problem (Garnett et al., 2012; Jiang et al., 2017; Vanchinathan et al., 2015), where we wish to find as many members (designs) of a rare desired class in a batch sequential manner with a fixed experimental budget. We call this problem *active generation*. Examples of this are compounds that could be useful pharmaceutical drugs, or highly active enzymes for catalyzing chemical reactions. We assume the design space is discrete or partially discrete, high-dimensional, and practically *innumerable*. For example, the number possible configurations of a single protein is $20^{\mathcal{O}(100)}$ (see, e.g., Sarkisyan et al., 2016).

We are interested in this objective for a variety of reasons. We may wish to study the properties of the "fitness landscape" (Papkou et al., 2023) to gain a better scientific understanding of a phenomenon such as natural evolution. Or, we may not be able to completely specify the constraints and objectives of a task, but we would like to characterize the space of feasible designs. For example, we want enzymes that can degrade plastics in an industrial setting, but we may not yet know the exact conditions (e.g. temperature, pH), some of which may be anti-correlated with enzyme catalytic activity.

Assuming we can take advantage of a prior distribution over designs, we formulate the search problem as inferring the posterior distribution over rare, desirable designs. Importantly, this posterior can be used for *generating new designs*. Specifically, we use (black-box) variational inference (VI) (Ranganath et al., 2014), and so refer to our method as variational search distributions (VSD). Our major contributions are: (1) we formulate the batch active search objective over a (practically) innumerable discrete design space as an instance of variational inference, (2) we present an algorithm, VSD, which solves this objective, and (3) we show that VSD satisfies well-defined requirements and desiderata specific to our problem. For example, it uses off-the-shelf gradient based optimization routines, is able to learn powerful generative models, and can take advantage of scalable predictive models. In our experiments we show that VSD can outperform existing baseline methods on a set of real applications. Finally, we evaluate our approach on the related sequential black-box optimization (BBO) problem, where we want to find the globally optimal design for a specific objective and show competitive performance when compared with state-of-the-art methods, e.g., based on Bayesian optimization (BO) (Garnett, 2023).

2 METHOD

In this section we formalize our problem and describe its requirements and desiderata. We also develop our proposed solution, based on variational inference, which we will refer to as variational search distributions (VSD).

2.1 THE PROBLEM OF ACTIVE GENERATION

We are given a design space $\mathcal{X}$, which can be discrete or mixed discrete-continuous and high dimensional, and where for each instance that we choose $\mathbf{x} \in \mathcal{X}$, we measure some corresponding property of interest (so-called fitness) $y \in \mathbb{R}$. For example, in our motivating application of DNA/RNA or protein sequences (henceforth referred to as just sequences), $\mathcal{X} = \mathcal{V}^M$ where $\mathcal{V}$ is the sequence vocabulary (e.g., amino acid labels, $|\mathcal{V}| = 20$) and M is the length of the sequence. However, we do not limit the application of our method to sequences. Using this framing, a real world experiment (for example, measuring the activity of an enzyme) can be modeled as an unknown relationship,

$$y = f_\bullet(\mathbf{x}) + \epsilon, \tag{1}$$

for some black-box function (e.g. the experiment), $f_\bullet$, and measurement error $\epsilon \in \mathbb{R}$, distributed according to $p(\epsilon)$ with $\mathbb{E}_{p(\epsilon)}[\epsilon] = 0$. Instead of wanting to model the whole space, we are only interested in a set of events which we choose based on fitness y. In particular, we want to perform active generation, that is to generate (from a probability density) a sample that is of high fitness. For instance (refer to Figure 1), we may be interested in the fittest measurable design; all designs above a minimum level of feasibility (e.g. a wild-type sequence), $\tau \in \mathbb{R}$; the distribution of these feasible designs; or the shape of the black-box function for these feasible designs,

$$\mathbf{x}^* = \operatorname*{argmax}_{\mathbf{x}} f_\bullet(\mathbf{x}), \quad \mathcal{S} := \{\mathbf{x} : y > \tau\}, \quad p(\mathbf{x}|y > \tau), \quad \text{or} \quad \mathcal{F} := \{f_\bullet(\mathbf{x}) : \mathbf{x} \in \mathcal{S}\}. \tag{2}$$

Our primary focus in this work is to estimate the super level-set distribution $p(\mathbf{x}|y > \tau)$ in a sequential manner. We assume that $\mathcal{S}$ are rare events in a high dimensional space, and that we have access to a prior belief, $p(\mathbf{x})$, which helps narrow in on this subset of $\mathcal{X}$. We are given a dataset, $\mathcal{D}_N := \{(y_n, \mathbf{x}_n)\}_{n=1}^N$, which may contain only a few instances of $y_n > \tau$. Given $p(\mathbf{x})$ and $\mathcal{D}_N$ we aim to recommend batches of unique candidates, $\{\mathbf{x}_{bt}\}_{b=1}^B$, for experimental evaluation (Equation 1) in a series of rounds, $t \in \{1, \ldots, T\}$, where $B = \mathcal{O}(1000)$ and we desire $\mathbf{x}_{bt} \in \mathcal{S}$. Each round, $\mathcal{D}_N$ is augmented with the experimental results of the previous batch, so $N \leftarrow N + B$. Estimating this super level-set distribution of $\mathbf{x}$ is computationally and statistically challenging and, therefore, we cast this as a *variational inference* problem. As we shall see later, our solution allows us to satisfy the following requirements and additional desiderata for our problem.

Requirements & Desiderata. *Problem requirements (R) and other desiderata (D).*

(R1) Rare *feasible designs, $y > \tau$, are rare events in $\mathcal{X}$ that need to be identified*

(R2) Sequential *non-myopic candidate generation, $\mathbf{x} \in \mathcal{S} \subset \mathcal{X}$, for sequential experiments*

(R3) Discrete *search over (combinatorially) large design spaces, e.g. $\mathbf{x} \in \mathcal{X} = \mathcal{V}^M$*

(R4) Batch *generation of up to* $\mathcal{O}(1000)$ diverse *candidate designs per round*

(D1) Guaranteed *convergence for certain choices of priors, variational distributions and predictive models*

(D2) Gradient *based optimization strategies for candidate searching*

(D3) Generative *models, $\mathbf{x}^{(s)} \sim q(\mathbf{x})$, that are task-specific for fit designs*

(D4) Scalable *predictive models that enable high-throughput experiments.*

2.2 VARIATIONAL SEARCH DISTRIBUTIONS

We cast the estimation of $p(\mathbf{x}|y > \tau)$ as a sequential optimization problem. A suitable objective for a round, t, is to minimize a divergence,

$$\phi_t^* = \operatorname*{argmin}_{\phi} \mathbb{D}[p(\mathbf{x}|y > \tau) \| q(\mathbf{x}|\phi)] \tag{3}$$

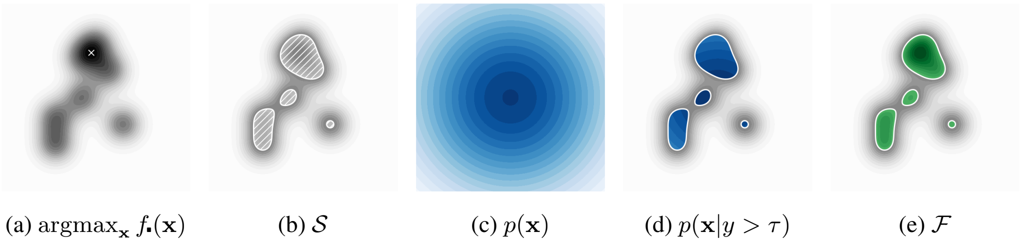

(a) $\operatorname{argmax}_{\mathbf{x}} f_\bullet(\mathbf{x})$ (b) $\mathcal{S}$ (c) $p(\mathbf{x})$ (d) $p(\mathbf{x}|y > \tau)$ (e) $\mathcal{F}$

Figure 1: Fitness landscape tasks. (a) A noise-less fitness landscape, $f_\bullet(\mathbf{x})$ and white '×' – the maximum fitness design, $\mathbf{x}^*$. (b) The super level-set of all fit designs – white hatched area, $\mathcal{S}$. (c) Prior belief $p(\mathbf{x})$. (d) The density/mass function of the super level-set, $p(\mathbf{x}|y > \tau)$ – blue contours. (e) The black box function for the super level-set, $\mathcal{F}$. See Equation 2 for definitions of these these tasks. Our primary goal is to estimate the density or mass function of the super level-set, (d). Since we assume a noisy relationship between $f_\bullet$ and y, the super level-set will not have a hard boundary as depicted, and $p(\mathbf{x}|y > \tau)$ will be non-zero over all $\mathcal{X}$.

where $q(\mathbf{x}|\phi)$ is a parameterized distribution from which we sample experimental candidate designs $\mathbf{x}_{bt}$, (D3), and which we aim to match to $p(\mathbf{x}|y > \tau)$. The difficulty is that we cannot directly evaluate or empirically sample from $p(\mathbf{x}|y > \tau)$. However, if we consider the reverse Kullback-Leibler (KL) divergence,

$$\underset{\phi}{\operatorname{argmin}}\ \mathbb{D}_{\mathrm{KL}}[q(\mathbf{x}|\phi)\|p(\mathbf{x}|y > \tau)] = \underset{\phi}{\operatorname{argmin}}\ \mathbb{E}_{q(\mathbf{x}|\phi)}\left[\log\frac{q(\mathbf{x}|\phi)}{p(\mathbf{x})} - \log p(y > \tau|\mathbf{x})\right], \tag{4}$$

where we have expanded $p(\mathbf{x}|y > \tau)$ using Bayes rule and dropped the constant term $p(y > \tau)$, we note that we no longer require evaluation of $p(\mathbf{x}|y > \tau)$ directly. We recognize the right hand side of Equation 4 as the well known (negative) variational evidence lower bound (ELBO),

$$\mathcal{L}_{\mathrm{ELBO}}(\phi) := \mathbb{E}_{q(\mathbf{x}|\phi)}[\log p(y > \tau|\mathbf{x})] - \mathbb{D}_{\mathrm{KL}}[q(\mathbf{x}|\phi)\|p(\mathbf{x})]\,. \tag{5}$$

For this we assume access to a prior distribution over the space of designs, $p(\mathbf{x})$, that may be informed from the data at hand. Henceforth, as we will develop a sequential algorithm, we will denote this prior with $p(\mathbf{x}|\mathcal{D}_0)$. We note the relationship between $\log p(y > \tau|\mathbf{x})$ and the probability of improvement (PI) acquisition function from BO (Kushner, 1964),

$$\log p(y > \tau|\mathbf{x}) := \log \mathbb{E}_{p(y|\mathbf{x},\mathcal{D}_N)}[\mathbb{1}[y > \tau]] = \log \alpha_{PI}(\mathbf{x}, \mathcal{D}_N, \tau)\,. \tag{6}$$

Here $\mathbb{1} : \{\text{false}, \text{true}\} \to \{0, 1\}$ is the indicator function and $p(y|\mathbf{x}, \mathcal{D}_N)$ is typically estimated using the posterior predictive distribution of a Gaussian process (GP) given data, $\mathcal{D}_N$. So $p(y > \tau|\mathbf{x}, \mathcal{D}_N) = \Psi((\mu_N(\mathbf{x}) - \tau)/\sigma_N(\mathbf{x}))$, where $\Psi(\cdot)$ is a cumulative standard normal distribution function, and $\mu_N(\mathbf{x})$, $\sigma^2_N(\mathbf{x})$ are the posterior predictive mean and variance, respectively, of the GP. We refer to this estimation strategy as GP-PI, and rewrite the ELBO accordingly,

$$\mathcal{L}_{\mathrm{ELBO}}(\phi, \tau, \mathcal{D}_N) = \mathbb{E}_{q(\mathbf{x}|\phi)}[\log \alpha_{PI}(\mathbf{x}, \mathcal{D}_N, \tau)] - \mathbb{D}_{\mathrm{KL}}[q(\mathbf{x}|\phi)\|p(\mathbf{x}|\mathcal{D}_0)]\,. \tag{7}$$

The method that maximizes the objective in Equation 7 we call variational search distributions (VSD), since we are using the variational posterior distribution as a means of searching the space of fit designs, satisfying (R1), (R2) and (R4). It is well known that when the true posterior is a member of the variational family indexed by ϕ, the above variational inference procedure has the potential to recover the exact posterior distribution. To recommend experimental candidates we sample a set of designs from our search distribution each round,

$$\{\mathbf{x}_{bt}\}_{b=1}^{B} \sim \prod_{b=1}^{B} q(\mathbf{x}|\phi^*_t), \quad \text{where} \quad \phi^*_t = \underset{\phi}{\operatorname{argmax}}\, \mathcal{L}_{\mathrm{ELBO}}(\phi, \tau, \mathcal{D}_N)\,. \tag{8}$$

We discuss the relationship between VSD and BO in Appendix F. In general, because of the discrete combinatorial nature of our problem, we cannot use the re-parameterization trick (Kingma & Welling, 2014) to estimate the gradients of the ELBO. Instead, we use the score function gradient estimator (Williams, 1992; Mohamed et al., 2020) with standard gradient descent methods (D2),

$$\nabla_\phi \mathcal{L}_{\mathrm{ELBO}}(\phi, \tau, \mathcal{D}_N) = \mathbb{E}_{q(\mathbf{x}|\phi)}\left[\left(\log \alpha_{PI}(\mathbf{x}, \mathcal{D}_N, \tau) - \log\frac{q(\mathbf{x}|\phi)}{p(\mathbf{x}|\mathcal{D}_0)}\right)\nabla_\phi \log q(\mathbf{x}|\phi)\right], \tag{9}$$

where we use Monte-Carlo sampling to approximate this expectation with a suitable variance reduction scheme, such as using a control variate or baseline (Mohamed et al., 2020). We find that the exponentially smoothed average of the ELBO works well in practice, and is the same strategy employed in Daulton et al. (2022). Effectively, VSD implements black-box variational inference (Ranganath et al., 2014) for parameter estimation, and despite the high-dimensional nature of $\mathcal{X}$, we find we only need $\mathcal{O}(1000)$ samples to estimate the required expectations for ELBO optimization on problems with $M = \mathcal{O}(100)$, satisfying (R3). Note that Equation 7 – 9 do not involve any data ($\mathcal{D}_N$) directly, only indirectly through the acquisition function. Hence the scalability of VSD is dependent on the complexity of training the underlying estimator of $p(y|\mathbf{x}, \mathcal{D}_N)$.

2.3 CLASS PROBABILITY ESTIMATION

So far our method indirectly computes the PI by transforming the predictions of a GP surrogate model, $p(y|\mathbf{x}, \mathcal{D}_N)$, as in Equation 6. Instead we may choose to follow the reasoning used by Bayesian optimization by density-ratio estimation (BORE) in Tiao et al. (2021); Oliveira et al. (2022); Song et al. (2022), and directly estimate the quantity we care about, $p(y > \tau|\mathbf{x}, \mathcal{D}_N)$. We do this with class probability estimation (CPE) on the labels $z := \mathbb{1}[y > \tau] \in \{0, 1\}$ so $p(y > \tau|\mathbf{x}, \mathcal{D}_N) = p(z = 1|\mathbf{x}, \mathcal{D}_N) \approx \pi_\theta(\mathbf{x})$, where $\pi_\theta : \mathcal{X} \to [0, 1]$. We can recover the class probability estimates using a proper scoring rule (Gneiting & Raftery, 2007) such as Brier score or log-loss on training data, $\mathcal{D}^z_N = \{(z_n, \mathbf{x}_n)\}^N_{n=1}$, e.g.,

$$\mathcal{L}_{\text{CPE}}(\theta, \mathcal{D}^z_N) := -\frac{1}{N}\sum_{n=1}^{N} z_n \log \pi_\theta(\mathbf{x}_n) + (1 - z_n) \log(1 - \pi_\theta(\mathbf{x}_n)). \tag{10}$$

The VSD objective using CPE becomes,

$$\mathcal{L}_{\text{ELBO}}(\phi, \theta) = \mathbb{E}_{q(\mathbf{x}|\phi)}[\log \pi_\theta(\mathbf{x})] - \mathbb{D}_{\text{KL}}[q(\mathbf{x}|\phi)\|p(\mathbf{x}|\mathcal{D}_0)]\,, \tag{11}$$

into which we plug $\theta^*_t = \text{argmin}_\theta\, \mathcal{L}_{\text{CPE}}(\theta, \mathcal{D}^z_N)$. Using a CPE also opens up the choice of estimators that are more scalable than GP-PI, satisfying our desiderata (D4). This may be crucial if we choose to run more than a few rounds of experiments with $B = \mathcal{O}(1000)$. Additionally, since VSD is a black-box method, we can choose to use CPEs that are non-differentiable, such as decision tree ensembles. The complete VSD algorithm is given in Algorithm 1, in which we have allowed for a threshold function, $\tau_t = f_\tau(\{y : y \in \mathcal{D}_N\}, \gamma_t)$. This function can be used to modify the threshold each round, e.g. following Tiao et al. (2021), an empirical quantile function $\tau_t = \hat{Q}_y(\gamma_t)$ where $\gamma_t \in (0, 1)$, or a constant τ in the case of estimating the distribution of the super level-set.

Algorithm 1 VSD optimization loop with CPE.

```
Require: Threshold $\gamma_1$ and $f_\tau$, dataset $\mathcal{D}_N$, black-box $f_\bullet$, prior $p(\mathbf{x}|\mathcal{D}_0)$, CPE $\pi_\theta(\mathbf{x})$, variational
    family $q(\mathbf{x}|\phi)$, budget $T$ and $B$.
function FitModels($\mathcal{D}_N, \tau$)
    $\mathcal{D}^z_N \leftarrow \{(z_n, \mathbf{x}_n)\}^N_{n=1}$, where $z_n = \mathbb{1}[y_n > \tau]$
    $\theta^* \leftarrow \text{argmin}_\theta\, \mathcal{L}_{\text{CPE}}(\theta, \mathcal{D}^z_N)$
    $\phi^* \leftarrow \text{argmax}_\phi\, \mathcal{L}_{\text{ELBO}}(\phi, \theta^*)$
    return $\phi^*, \theta^*$
for round $t \in \{1, \ldots, T\}$ do
    $\tau_t \leftarrow f_\tau(\{y : y \in \mathcal{D}_N\}, \gamma_t)$
    $\phi^*_t, \theta^*_t \leftarrow$ FitModels($\mathcal{D}_N, \tau_t$)
    $\{\mathbf{x}_{bt}\}^B_{b=1} \leftarrow q(\mathbf{x}|\phi^*_t)$
    $\{y_{bt}\}^B_{b=1} \leftarrow \{f_\bullet(\mathbf{x}_{bt}) + \epsilon_{bt}\}^B_{b=1}$
    $\mathcal{D}_{N+B} \leftarrow \mathcal{D}_N \cup \{(\mathbf{x}_{bt}, y_{bt})\}^B_{b=1}$
$\tau_* \leftarrow f_\tau(\{y : y \in \mathcal{D}_N\}, \gamma_*)$
$\phi^*, \theta^* \leftarrow$ FitModels($\mathcal{D}_N, \tau_*$)
return $\phi^*, \theta^*$
```

2.4 THEORETICAL ANALYSIS

In this section we summarize the main theoretical results concerning VSD and its estimates. We show that VSD sampling distributions converge to a target distribution that characterizes the level set given by τ, satisfying (D1) in two general settings. We first derive results assuming $f_\bullet$ is drawn from a Gaussian process, i.e., $f_\bullet \sim \mathcal{GP}(0, k)$, with a positive-semidefinite covariance (or kernel) function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (Appendix D) and then using GP-PI as the CPE for VSD. These results are then extended to probabilistic classifiers based on wide neural networks (Appendix E) by means of the neural tangent kernel (NTK) for the given architecture (Jacot et al., 2018). For simplicity, we set $B = 1$ and $N = t$, though sampling with $B > 1$ should improve rates by a multiplicative factor.

Theorem 2.1. *Let assumptions D.1 to D.5 hold. Then VSD equipped with GP-PI approaches the level-set distribution at the following rate:*

$$\mathbb{D}[p(\mathbf{x}|y > \tau_t, \mathcal{D}_t) \| p(\mathbf{x}|y > \tau_t, f_\bullet)] \in \mathcal{O}_{\mathbb{P}}(t^{-1/2})\,.$$

This result is based on showing that the GP posterior variance vanishes at an optimal rate of $\mathcal{O}(t^{-1})$ in our setting (Lemma D.5). We also analyze the rate at which VSD finds feasible designs, or "hits", compared to an oracle with full knowledge of $f_\bullet$. After T rounds, the number of hits found by VSD is $H_T = \sum_{t=1}^{T} \mathbb{1}[y_t > \tau_{t-1}]$, where y_t is generated from Equation 1 and $\mathbf{x}_t \sim p(\mathbf{x}|y > \tau_{t-1}, \mathcal{D}_{t-1})$. The number of hits, H_T^*, from an agent that fully knows $f_\bullet$ is the same but for generating conditioned on $f_\bullet$ with $\mathbf{x}_t \sim p(\mathbf{x}|y > \tau_{t-1}, f_\bullet)$. Using this definition and Theorem 2.1 we show the following.

Corollary 2.1. *Under the settings in Theorem 2.1, we also have that:*

$$\mathbb{E}[|H_T - H_T^*|] \in \mathcal{O}(\sqrt{T})\,.$$

$\mathbb{E}[H_T]$ is related to the empirical recall measure in Equation 16 up to the normalization constant, but it does not account for repeated hits, which are treated as false discoveries (false positives) under recall. Lastly, for NN-based CPEs, we obtain convergence rates dependent on the spectrum of the NTK (Proposition E.2), which we instantiate for ReLU networks below. For the full results and proofs, please see Appendix D for the GP-based analysis and Appendix E for the NTK results.

Corollary 2.2. *Let π_θ be modeled via a fully connected ReLU network. Then, under the assumptions in Proposition E.2, VSD achieves:*

$$\mathbb{D}[p(\mathbf{x}|y > \tau_t, \mathcal{D}_t) \| p(\mathbf{x}|y > \tau, f_\bullet)] \in \widetilde{\mathcal{O}}_{\mathbb{P}}\left(t^{-\frac{1}{2(M+1)}}\right), \tag{12}$$

which asymptotically vanishes for all finite sequence lengths M.

3 RELATED WORK

VSD can be viewed as one of many methods that makes use of the bound (Staines & Barber, 2013),

$$\max_{\mathbf{x}} f_\bullet(\mathbf{x}) \geq \max_{\phi} \mathbb{E}_{q(\mathbf{x}|\phi)}[f_\bullet(\mathbf{x})]\,. \tag{13}$$

The maximum is always greater than or equal to the expected value of a random variable. This bound is useful for black-box optimization (BBO) of $f_\bullet$, and becomes tight if $q(\mathbf{x}|\phi) \to \delta(\mathbf{x}^*)$, see Appendix F for more detail and VSD's relation. Other well known methods that make use of this bound are evolution strategies (ES) and natural evolution strategies (NES) (Wierstra et al., 2014), variational optimization (VO) (Staines & Barber, 2013; Bird et al., 2018), estimation of distribution algorithms (EDA) (Larrañaga & Lozano, 2001), and Bayesian optimization with probabilistic reparameterisation (BOPR) (Daulton et al., 2022). For learning the parameters of the variational distribution, ϕ, they variously make use of maximum likelihood estimation or the score function gradient estimator (REINFORCE) (Williams, 1992). Algorithms that modify Equation 13 to stop the collapse of $q(\mathbf{x}|\phi)$ to a point mass for batch design include design by adaptive sampling (DbAS) (Brookes & Listgarten, 2018) and conditioning by adaptive sampling (CbAS) (Brookes et al., 2019). They use fixed samples $\mathbf{x}^{(s)}$ from $q(\mathbf{x}|\phi^*_{t-1})$ for approximating the expectation, and then optimize ϕ using a weighted maximum-likelihood or variational style procedure. We can take a unifying view of many of these algorithms by recognizing the general gradient estimator,

$$\mathbb{E}_{q(\mathbf{x}|\phi')}[w(\mathbf{x}) \nabla_\phi \log q(\mathbf{x}|\phi)]\,, \tag{14}$$

where we give each component in Table 1. BORE* has been adapted to discrete $\mathcal{X}$ by using the score function gradient estimator and CbAS and DbAS have been adapted to use a CPE – their original derivations use the equivalent of a PI acquisition function.

Method	$w(\mathbf{x})$	ϕ'	Fixed $\mathbf{x}^{(s)} \sim q(\mathbf{x}\|\phi')$?
VSD	$\log \pi_{\theta^*}(\mathbf{x}) + \log p(\mathbf{x}\|\mathcal{D}_0) - \log q(\mathbf{x}\|\phi)$	ϕ	No
CbAS	$\pi_{\theta^*}(\mathbf{x}) p(\mathbf{x}\|\mathcal{D}_0) / q(\mathbf{x}\|\phi^*_{t-1})$	ϕ^*_{t-1}	Yes
DbAS	$\pi_{\theta^*}(\mathbf{x})$	ϕ^*_{t-1}	Yes
BORE*	$\pi_{\theta^*}(\mathbf{x})$	ϕ	No
BOPR	$\alpha(\mathbf{x}, \mathcal{D}_N)$	ϕ	No

Table 1: How related methods can be adapted from Equation 14. VSD, CbAS and DbAS may also use a cumulative distribution representation of $\alpha_{\text{PI}}(\mathbf{x}, \mathcal{D}_N, \tau)$ in place of $\pi_{\theta^*}(\mathbf{x})$.

Method	Rare $\mathbf{x} \in \mathcal{S}$ (R1)	Sequential (R2)	Discrete $\mathcal{X}$ (R3)	Batch $\{\mathbf{x}_{bt}\}_{b=1}^{B}$ (R4)	Guaranteed (D1)	Gradient descent (D2)	Generative $q(\mathbf{x}\|\phi)$ (D3)	Scalable (D4)	General acq./reward fn.	Amortization
BOPR (Daulton et al., 2022)	✗	✓	✓	✗	✓	✓	–	✗	✓	–
BORE (Tiao et al., 2021)	✗	✓	–	✗	✓	✓	–	✓	✗	–
Batch BORE (Oliveira et al., 2022)	✓	✓	✗	✓	✓	✓	✓	✓	✗	✓
DbAS (Brookes & Listgarten, 2018)	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓
CbAS (Brookes et al., 2019)	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓
Amortized BO (Swersky et al., 2020)	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓
GFlowNets (Jain et al., 2022)	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓
DynaPPO (Angermueller et al., 2019)	✗	✓	✓	✓	✗	✓	✓	–	✓	✓
AdaLead (Sinai et al., 2020)	✗	✓	✓	✓	✗	✗	✗	–	✗	✗
PEX (Ren et al., 2022)	✗	✓	✓	✓	✗	✗	✗	–	✗	✗
GGS (Kirjner et al., 2024)	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗
LSO e.g. (Tripp et al., 2020)	✗	✓	✓	✗	✗	✓	✓	–	✓	–
LaMBO (Stanton et al., 2022)	✗	✓	✓	✓	✗	✓	✓	–	✓	✓
LaMBO-2 (Gruver et al., 2024)	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓
VSD (ours)	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓

Table 2: Feature table of competing methods: ✓ has feature, ✗ does not have feature, – partially has feature, or requires simple modification. We follow Swersky et al. (2020) in their definition of amortization referring to the ability to use $q(\mathbf{x}|\phi^*_{t-1})$ for warm-starting the optimization of ϕ_t.

A number of finite horizon methods have been applied to biological sequence BBO tasks, such as Amortized BO (Swersky et al., 2020), GFlowNets (Jain et al., 2022), and the reinforcement learning based DynaPPO (Angermueller et al., 2019). Latent space optimization (LSO) (Tripp et al., 2020; Stanton et al., 2022; Gruver et al., 2024) tackles optimization of sequences by encoding them into a continuous latent space within which BO methods are used. Selected candidates are decoded back into sequences before black box evaluation; see González-Duque et al. (2024) for a comprehensive survey. VSD does not require a latent space nor an encoder, and as such is more closely related to probabilistic reparameterisation methods (Daulton et al., 2022) or continuous relaxations (Michael et al., 2024). Heuristic stochastic search methods such as AdaLead (Sinai et al., 2020) and proximal exploration (PEX) (Ren et al., 2022) have also demonstrated strong empirical performance on these tasks. We compare the properties of the most relevant methods to our problem in Table 2.

Other methods that explicitly attempt to estimate the super level-set distribution, $p(\mathbf{x}|y > \tau)$, include CbAS, which optimizes the forward KL divergence, $\mathbb{D}_{\text{KL}}[p(\mathbf{x}|y > \tau) \| q(\mathbf{x}|\phi)]$, in a sequential fashion using importance weighted cross entropy estimation (Rubinstein, 1999). Batch-BORE (Oliveira et al., 2022) also optimizes the reverse KL divergence and uses CPE, but with Stein variational inference (Liu & Wang, 2016) for continuous and diverse batch candidates $\{\mathbf{x}_{tb} \in \mathbb{R}^M\}_{b=1}^{B}$ (the authors optimize on the simplex, and then choose the maximum coordinate for their discrete experiments). There is a rich literature on the related task of active learning and BO for level set estimation (LSE) (Bryan et al., 2005; Gotovos et al., 2013; Bogunovic et al., 2016; Zhang et al., 2023). However we focus on the task of learning a generative model over a practically innumerable discrete space.

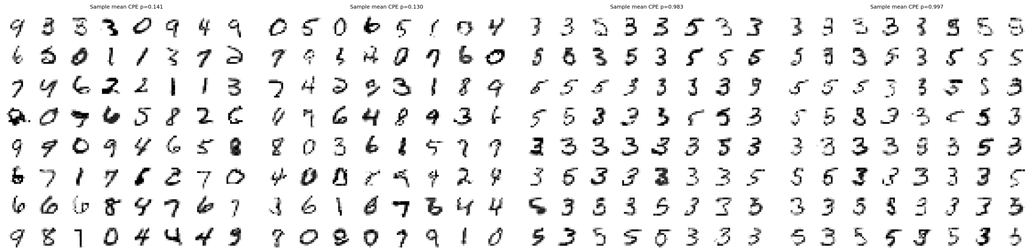
4 EXPERIMENTS

Firstly we test our method, VSD, on its ability to generate complex, structured candidates, $\mathbf{x}$, in a single round by training it to generate a subset of handwritten digits from flattened MNIST images (LeCun et al., 1998) in Sec. 4.1. We then compare VSD on two sequence design tasks against existing baseline methods. The first of these tasks (Sec. 4.2) is to discover as many unique, fit sequences as possible using the datasets DHFR (Papkou et al., 2023), TrpB (Johnston et al., 2024) and TFBIND8 (Barrera et al., 2016). These datasets contain near complete evaluations of $\mathcal{X}$, and to our knowledge DHFR and TrpB are novel in the machine learning literature. The second (Sec. 4.3) is a more traditional black-box optimization task of finding the maximum of an unknown function; using datasets AAV (Bryant et al., 2021) and GFP (Sarkisyan et al., 2016). A complete combinatorial assessment is infeasible for these latter datasets, and so we use the convolution neural network oracle presented in Kirjner et al. (2024) as *in-silico* ground truth. The corresponding datasets involve $|\mathcal{V}| \in \{4, 20\}$, $4 \leq M \leq 237$ and $65,000 < |\mathcal{X}| < 20^{237}$. We discuss the settings and properties of these datasets in greater detail in Appendix B.

For the biological sequence experiments we run a predetermined number of experimental rounds, $T = 10$, we set the batch size to $B = 128$, and use five different seeds for random initialization. For the first set of real-sequence experiments in Sec. 4.2, we use a fixed threshold, τ, with the aim of finding $\mathbf{x} \in \mathcal{S}$ using both probabilistic and non probabilistic models. For the next set of experiments in Sec. 4.3, we set τ_t adaptively each round for testing VSD's ability to find the maximizer, $\mathbf{x}^*$, of the black-box function, $f_\bullet$. We compare against DbAS (Brookes & Listgarten, 2018), CbAS (Brookes et al., 2019), AdaLead (Sinai et al., 2020), PEX (Ren et al., 2022), BORE (Tiao et al., 2021) adapted to use the score function gradient estimator, and a naïve baseline that uses random samples from the prior, $p(\mathbf{x}|\mathcal{D}_0)$. To reduce confounding, all methods share the same surrogate model and other components where possible (acquisition functions, priors and variational distributions). We optimize VSD, CbAS, DbAS and BORE for a minimum of 5000 iterations each round. When we use a CPE, AdaLead's κ parameter is set to 0.5 since the CPE already incorporates the appropriate threshold.

4.1 CONDITIONAL GENERATION OF HANDWRITTEN DIGITS

Our motivating application for VSD is to model the space of fit DNA and protein sequences, which are string-representations of complex 3-dimensional structures. In this experiment we aim to demonstrate, by analogy, that VSD can operate with the types of generative models required for representing these complex structures. For this task, we have chosen to 'unroll' (reverse the order of every odd row, and flatten) down-scaled (14×14 pixel, 8-bit) MNIST (LeCun et al., 1998) images into sequences, $\mathbf{x}$, where M = 196 and $|\mathcal{V}| = 8$. We then train long short-term memory (LSTM) recurrent neural network (RNN) and decoder-only causal transformer generative models on the entire MNIST training set by maximum likelihood (ML). These generative distributions are used as our prior models, $p(\mathbf{x}|\mathcal{D}_0)$, for VSD and we detail their form in Sec. B.1. The task is then to use VSD in one round to estimate the posterior $p(\mathbf{x}|y \in \{3, 5\})$ using a CPE trained on labels $z_n = \mathbb{1}[y_n \in \{3, 5\}]$. We use a convolutional architecture for the CPE given in Sec. B.2, and it achieves a test balanced accuracy score of $\sim 99\%$. We parameterize the variational distributions, $q(\mathbf{x}|\phi)$, in the same way as the priors, and initialize these distribution parameters from the prior distribution parameters. During training with ELBO the prior distribution parameters are locked, and we run training for 5000 iterations. This is exactly lines 8 and 9 in Algorithm 1. Samples are visualized from the resulting variational distributions with the corresponding priors in Figure 2. We can see that the prior LSTM and transformer are able to generate convincing digits once the sampled sequences are 're-rolled', and that VSD is able to effectively refine these distributions, even though it does not have access to



(a) LSTM Prior (b) Transformer Prior (c) LSTM Posterior (d) Transformer Posterior

Figure 2: (a) and (b) are samples from the LSTM and transformer priors, respectively. (c) and (d) show samples from the LSTM and transformer VSD variational distributions respectively. We also report the samples mean scores according to the CPE probabilities.

any data directly – only scores from the CPE. Both the LSTM and transformer yield qualitatively similar results, and have similar mean scores from the CPE.

4.2 FITNESS LANDSCAPES

In this setting we wish to find $\mathbf{x} \in \mathcal{S}$, so we fix τ over all rounds for all competing methods. We only consider the combinatorially (near) complete datasets to avoid any pathological behavior from relying on machine learning oracles (Surana et al., 2024). Results are presented in Figure 3. The primary measures by which we compare methods are precision, recall and performance (the last adapted from Jain et al. (2022)),

$$\text{Precision}_t = \frac{1}{\min\{tB, |\mathcal{S}|\}} \sum_{r=1}^{t} \sum_{b=1}^{B} \mathbb{1}[y_{br} > \tau] \cdot \mathbb{1}[\mathbf{x}_{br} \notin \mathcal{X}^q_{b-1,r}], \tag{15}$$

$$\text{Recall}_t = \frac{1}{\min\{TB, |\mathcal{S}|\}} \sum_{r=1}^{t} \sum_{b=1}^{B} \mathbb{1}[y_{br} > \tau] \cdot \mathbb{1}[\mathbf{x}_{br} \notin \mathcal{X}^q_{b-1,r}], \tag{16}$$

$$\text{Performance}_t = \sum_{r=1}^{t} \sum_{b=1}^{B} y_{br} \cdot \mathbb{1}[\mathbf{x}_{br} \notin \mathcal{X}^q_{b-1,r}]. \tag{17}$$

Here $\mathcal{X}^q_{br} \subset \mathcal{X}$ is the set of experimentally queried sequences by the bth batch member of the rth round, including the initial training set. These measures are comparable among probabilistic and non probabilistic methods. Precision and recall measure the ability of a method to efficiently explore $\mathcal{S}$, where $\min\{tB, |\mathcal{S}|\}$ is the size of the selected set at round t (bounded by the number of good solutions), and $\min\{TB, |\mathcal{S}|\}$ is the number of positive elements possible in the experimental budget. Strictly, recall should be normalized by $|\mathcal{S}|$, but we use TB here since it may not be realistic to have the experimental budget to fully explore $\mathcal{S}$. Performance measures the cumulative fitness of the unique batch members, and unlike Jain et al. (2022) we do not normalize this measure.

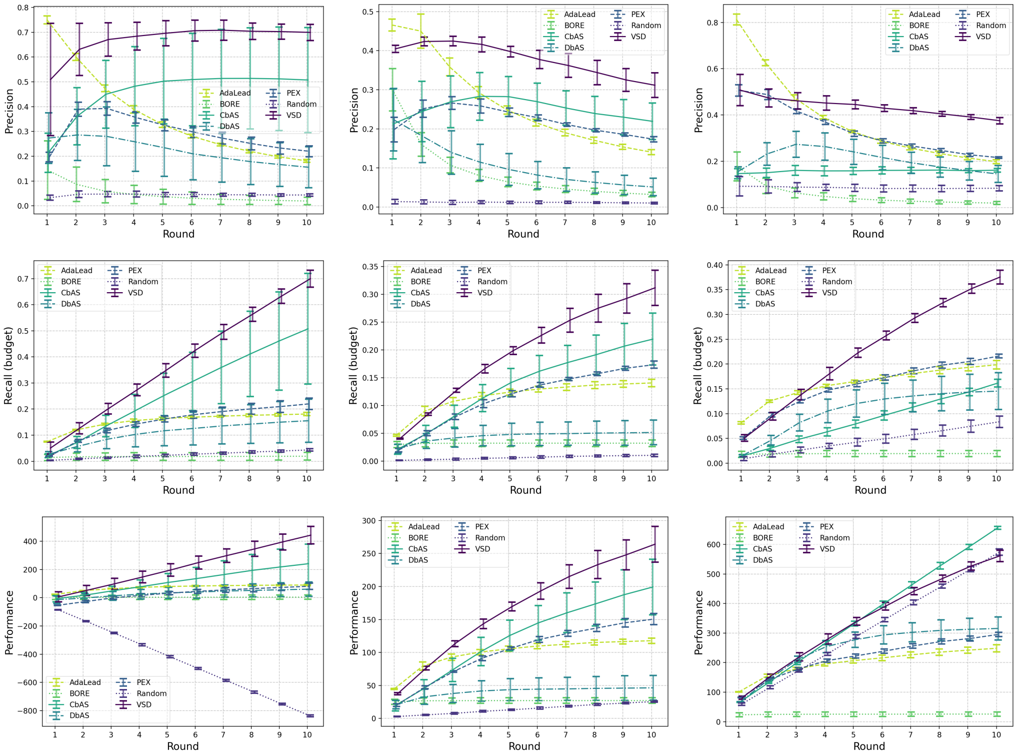
For the DHFR and TrpB experiments we set maximum fitness in the training dataset to be that of the wild type, and τ to be slightly below the wild type fitness value (so we have ~ 10 positive examples to train the CPE with). We use a randomly selected $N_{\text{train}} = 2000$ below the wild-type fitness to initially train the CPE, we also explicitly include the wild-type. The thresholds and wild-type fitness values are; DHRF: $\tau = -0.1$, $y_{\text{wt}} = 0$, TrpB: $\tau = 0.35$, $y_{\text{wt}} = 0.409$. We follow the same procedure for the TFBIND8 experiment, however, there is no notion of a wild-type sequence in this data, and so we set $\tau = 0.75$, and $y_{\text{train max}} = 0.85$. We use a uniform prior over sequences, $p(\mathbf{x}) = \prod_{m=1}^{M} \text{Categ}(x_m|\mathbf{1} \cdot |\mathcal{V}|^{-1})$, since these are relatively small search spaces, and the sub-sequences of nucleic/amino acids have been specifically selected for their task. Similarly, we find that relatively simple independent (mean-field) variational distributions of the form in Equation 20 and MLP based CPEs work best for these experiments (details in Sec. B.2).

Results are presented in Figure 3. VSD is the best performing method by most of the measures, with the related method CbAS also often performing well. We have found the AdaLead and PEX evolutionary-search based methods to be effective on lower-dimensional problems (TFBIND8 being the lowest here), however we consistently observe their performance degrading as the dimension of the problem increases. We suspect this is a direct consequence of their random mutation strategies being suited to exploration in low dimensions, but less efficient in higher dimensions compared to the learned generative models employed by VSD, CbAS, and DbAS. Our modified version of BORE (which is just the expected log-likelihood component of Equation 11) performs badly in all cases, and this is a direct consequence of its variational distribution collapsing to a point mass. In a non-batch setting this behavior is not problematic, but shows the importance of the KL divergence of VSD in this batch setting. We replicate these experiments in Sec. C.1 using GP-PI, also backed by our guarantees. In all cases VSD's results remain similar or improve slightly, whereas the other methods results remain similar or degrade. We also report on batch diversity scores in Sec. C.2.

4.3 BLACK-BOX OPTIMIZATION

In this experiment we aim to find the global maximizers, $(y^*, \mathbf{x}^*)$, of the black-box function, $f_\bullet$. For this, we set τ_t adaptively by specifying it as an empirical quantile of the observed target values,

$$\tau_t = \tilde{Q}^t_y(\gamma_t = p^\eta_{t-1}) \tag{18}$$



(a) DHFR (b) TrpB (c) TFBIND8

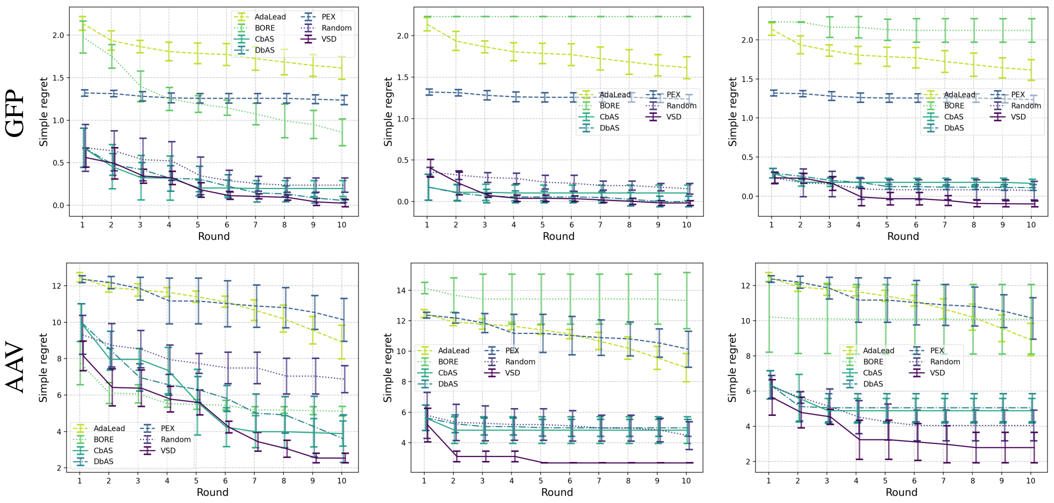
Figure 3: Fitness landscape results. Precision (Equation 15), recall (Equation 16) and performance (Equation 17) – higher is better – for the combinatorially (near) complete datasets, DHFR and TrpB and TFBIND8. The random method is implemented by drawing B samples uniformly.

where $\tilde{Q}_y^t$ is the empirical quantile function of targets at round t, p_{t-1} is a percentile from the previous round, and $\eta \in [0, 1]$ is a parameter that controls an annealing-like schedule for τ_t that prioritizes exploration of the fitness landscape in earlier rounds and exploitation of known fit regions in later rounds. This is a strategy loosely-similar to Srinivas et al. (2010). The measure of interest for these experiments is simple/instantaneous regret r_t which quantifies how close the methods get to finding the globally fittest sequence,

$$\mathrm{r}_t = y^* - \max_y \{y_{bi}\}_{b=1,i=1}^{B,t}, \tag{19}$$

where y^* is the fitness value of the fittest sequence $\mathbf{x}^*$. We use the higher dimensional AAV (y^*=19.54) and GFP (y^*=4.12) datasets to show that VSD can scale to higher dimensional problems. However, the $\mathcal{X}$ of these experiments is completely intractable to fully explore experimentally, and so we use a predictive oracle trained on all of the original experimental data as the ground-truth black-box function. This is the same strategy used in Brookes et al. (2019); Jain et al. (2022); Trabucco et al. (2021); Kirjner et al. (2024) among others, and we use the exact CNN-based oracles from Kirjner et al. (2024) for these experiments. However, we note here that some of the oracles used in these experiments do not predict well out-of-distribution (Surana et al., 2024), which limits their real-world applicability.

We follow Kirjner et al. (2024) in the experimental settings for the AAV and GFP datasets, but we modify the maximum fitness training point and training dataset sizes to make them more amenable to a sequential optimization setting. The initial percentiles, schedule, and max training fitness values are; AAV: $p_0 = 0.8$, $\eta = 0.7$, $y_{\max} = 5$, GFP: $p_0 = 0.8$, $\eta = 0.7$ $y_{\max} = 1.9$. The edit distance between $\mathbf{x}^*$ and the fittest sequence in the CPE training data is 8 for GFP, and 13 for AAV. We again use a random $N_{\text{train}} = 2000$ for training the CPEs, which in this case are CNNs – architecture specifics are in Sec. B.2.



(a) Independent (b) LSTM (c) Transformer

Figure 4: BBO results. Simple regret (Equation 19) – lower is better – on GFP and AAV with independent and auto-regressive variational distributions. The PEX and AdaLead results are replicated between the plots, since they are unaffected by choice of variational distribution. The random method is implemented by drawing B samples from the prior models.

In these higher dimensional settings, we find that performance of the methods heavily relies on using an informed prior (in the case of VSD and CbAS), or initial variational distribution (in the case of DbAS and BORE). To this end, we follow Brookes et al. (2019) and fit the initial variational distribution to the CPE training sequences (regardless of fitness), but we use maximum likelihood. Then for VSD and CbAS we copy this distribution and fix its parameters for the remainder of the experiment for use as a prior. We also use this prior for the Random method, and AdaLead and PEX use alternative generative heuristics. For these experiments we use the simple independent variational distribution from before, and also the same LSTM and causal decoder-only transformer models from Sec. 4.1. We present additional results in Sec. C.3 with uninformed priors and transition variational distributions of the form $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \phi)$.

The results are summarized in Figure 4. VSD is among the leading methods for all experiments. VSD appears better take advantage of the more complex variational distributions than the competing methods CbAS and DbAS since it can sample from the adapted variational distribution while learning it. We can see that AdaLead, PEX and often BORE all perform worse than random for reasons previously mentioned. Simple regret can drop below zero for these experiments since an oracle is used as the black box function, but the global maximizer is taken from the experimental data. This potentially highlights some of the overconfidence issues when using oracles as outlined in Surana et al. (2024). Batch diversity scores for these experiments are presented in Sec. C.2.

5 CONCLUSION

We have presented the problem of active generation (finding designs of a rare class sequentially under some experimental constraints), and a method for efficiently generating samples which we call variational search distributions (VSD). Underpinned by variational inference, VSD satisfies critical requirements and important desiderata, including learning generative models for fit sequences and batch candidate generation. We show that VSD converges asymptotically to the true level-set distribution at the same rate as a Monte-Carlo estimator with full knowledge of the true distribution. We showcased the benefits of our method empirically on a set of combinatorially complete and high dimensional sequential-design biological problems and show that it can effectively learn powerful generative models of fit designs. Finally, our framework can be generalized to more complex application scenarios, potentially involving other challenging combinatorial optimization problems (Bengio et al., 2021), such as graph structures (Annadani et al., 2023), and mixed discrete-continuous variables, which are worth investigating as future work directions.

REFERENCES

Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *International conference on learning representations*, 2019.

Yashas Annadani, Nick Pawlowski, Joel Jennings, Stefan Bauer, Cheng Zhang, and Wenbo Gong. BayesDAG: Gradient-based posterior inference for causal discovery. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020.

Luis A Barrera, Anastasia Vedenko, Jesse V Kurland, Julia M Rogers, Stephen S Gisselbrecht, Elizabeth J Rossin, Jaie Woodard, Luca Mariani, Kian Hong Kock, Sachi Inukai, et al. Survey of variation in human transcription factors reveals prevalent dna binding changes. *Science*, 351 (6280):1450–1454, 2016.

David G. T. Barrett and Benoit Dherin. Implicit gradient regularization. In *International Conference on Learning Representations (ICLR)*. OpenReview, 2021.

Heinz Bauer. *Probability theory and elements of measure theory*. Academic Press, 2nd edition, 1981.

Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2): 405–421, 2021. ISSN 0377-2217.

Thomas Bird, Julius Kunze, and David Barber. Stochastic variational optimization. *arXiv preprint arXiv:1809.04855*, 2018.

Ilija Bogunovic, Jonathan Scarlett, Andreas Krause, and Volkan Cevher. Truncated variance reduction: A unified approach to bayesian optimization and level-set estimation. *Advances in neural information processing systems*, 29, 2016.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

David Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In *International conference on machine learning*, pp. 773–782. PMLR, 2019.

David H Brookes and Jennifer Listgarten. Design by adaptive sampling. *arXiv preprint arXiv:1810.03714*, 2018.

Brent Bryan, Robert C Nichol, Christopher R Genovese, Jeff Schneider, Christopher J Miller, and Larry Wasserman. Active learning for identifying function threshold boundaries. *Advances in neural information processing systems*, 18, 2005.

Drew H Bryant, Ali Bashir, Sam Sinai, Nina K Jain, Pierce J Ogden, Patrick F Riley, George M Church, Lucy J Colwell, and Eric D Kelsic. Deep diversification of an aav capsid protein by machine learning. *Nature Biotechnology*, 39(6):691–696, 2021.

Lin Chen and Sheng Xu. Deep neural tangent kernel and laplace kernel have the same {rkhs}. In *International Conference on Learning Representations*, 2021.

Sayak Ray Chowdhury and Aditya Gopalan. On Kernelized Multi-armed Bandits. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Sydney, Australia, 2017.

Samuel Daulton, Xingchen Wan, David Eriksson, Maximilian Balandat, Michael A Osborne, and Eytan Bakshy. Bayesian optimization over discrete and mixed spaces via probabilistic reparameterization. *Advances in Neural Information Processing Systems*, 35:12760–12774, 2022.

Lester E. Dubins and David A. Freedman. A sharper form of the Borel-Cantelli lemma and the strong law. *The Annals of Mathematical Statistics*, 36(3):800–807, 1965.

Audrey Durand, Odalric-Ambrym Maillard, and Joelle Pineau. Streaming kernel regression with provably adaptive mean, variance, and regularization. *Journal of Machine Learning Research*, 19, 2018.

Rick Durrett. *Probability: theory and examples*. Cambridge University Press, New York, NY, 5th edition, 2019.

Ronen Eldan, Dan Mikulincer, and Tselil Schramm. Non-asymptotic approximations of neural networks by Gaussian processes. In *34th Annual Conference on Learning Theory (COLT)*, volume 134 of *Proceedings of Machine Learning Research*, 2021.

Henry E. Fleming. Equivalence of regularization and truncated iteration in the solution of ill-posed image reconstruction problems. *Linear Algebra and Its Applications*, 130(C):133–150, 1990.

E. García-Portugués. *Notes for Nonparametric Statistics*. Bookdown, 2024. URL `https://bookdown.org/egarpor/NP-UC3M/`. Version 6.9.1. ISBN 978-84-09-29537-1.

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

Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Jeff G. Schneider, and Richard P. Mann. Bayesian optimal active search and surveying. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Miguel González-Duque, Richard Michael, Simon Bartels, Yevgen Zainchkovskyy, Søren Hauberg, and Wouter Boomsma. A survey and benchmark of high-dimensional bayesian optimization of discrete sequences. *arXiv preprint arXiv:2406.04739*, 2024.

Miguel González-Duque, Simon Bartels, and Richard Michael. poli: a libary of discrete sequence objectives, January 2024. URL `https://github.com/MachineLearningLifeScience/poli`.

Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pp. 1344–1350, 2013.

Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36, 2024.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. Infinite attention: NNGP and NTK for deep attention networks. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*. PMLR 108, 2020.

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *International Conference on Learning Representations (ICLR)*. OpenReview, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, Montreal, Canada, 2018.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure FP Dossou, Chanakya Ajit Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, et al. Biological sequence design with gflownets. In *International Conference on Machine Learning*, pp. 9786–9801. PMLR, 2022.

Shali Jiang, Gustavo Malkomes, Geoff Converse, Alyssa Shofner, Benjamin Moseley, and Roman Garnett. Efficient nonmyopic active search. In *International Conference on Machine Learning*, pp. 1714–1723. PMLR, 2017.

Kadina E. Johnston, Patrick J. Almhjell, Ella J. Watkins-Dulaney, Grace Liu, Nicholas J. Porter, Jason Yang, and Frances H. Arnold. A combinatorially complete epistatic fitness landscape in an enzyme active site. *Proceedings of the National Academy of Sciences*, 121(32):e2400439121, 2024. doi: 10.1073/pnas.2400439121.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Andrew Kirjner, Jason Yim, Raman Samusevich, Shahar Bracha, Tommi S Jaakkola, Regina Barzilay, and Ila R Fiete. Improving protein optimization with smoothed fitness landscapes. In *The Twelfth International Conference on Learning Representations*, 2024.

Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 1964.

Pedro Larrañaga and Jose A Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*, volume 2. Springer Science & Business Media, 2001.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. On the linearity of large non-linear models: When and why the tangent kernel is constant. In *Advances in Neural Information Processing Systems*, 2020.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

Odalric-Ambrym Maillard. Self-normalization techniques for streaming confident regression. *HAL*, <hal-01349, 2016.

Richard Michael, Simon Bartels, Miguel González-Duque, Yevgen Zainchkovskyy, Jes Frellsen, Søren Hauberg, and Wouter Boomsma. A continuous relaxation for discrete bayesian optimization. *arXiv preprint arXiv:2404.17452*, 2024.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020.

Michael Murray, Hui Jin, Benjamin Bowman, and Guido Montufar. Characterizing the spectrum of the NTK via a power series expansion. In *International Conference on Learning Representations (ICLR)*, Kigali, Rwanda, 2023.

Mojmír Mutný and Andreas Krause. Efficient High Dimensional Bayesian Optimization with Additivity and Quadrature Fourier Features. In *Conference on Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2018.

Rafael Oliveira, Louis Tiao, and Fabio T Ramos. Batch bayesian optimisation via density-ratio estimation with guarantees. *Advances in Neural Information Processing Systems*, 35:29816–29829, 2022.

Andrei Papkou, Lucia Garcia-Pastor, José Antonio Escudero, and Andreas Wagner. A rugged yet easily navigable fitness landscape. *Science*, 382(6673):eadh3860, 2023.

Mary Phuong and Marcus Hutter. Formal algorithms for transformers. *arXiv preprint arXiv:2207.09238*, 2022.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pp. 814–822. PMLR, 2014.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA, 2006.

Zhizhou Ren, Jiahan Li, Fan Ding, Yuan Zhou, Jianzhu Ma, and Jian Peng. Proximal exploration for model-guided protein sequence design. In *International Conference on Machine Learning*, pp. 18520–18536. PMLR, 2022.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1:127–190, 1999.

Karen S Sarkisyan, Dmitry A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitry N Ivankov, Nina G Bozhanova, Mikhail S Baranov, Onuralp Soylemez, et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, 2016.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2001.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

Sam Sinai, Richard Wang, Alexander Whatley, Stewart Slocum, Elina Locane, and Eric D Kelsic. Adalead: A simple and robust adaptive greedy search algorithm for sequence design. *arXiv preprint arXiv:2010.02141*, 2020.

Samuel L. Smith, Benoit Dherin, David G. T. Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *Proceedings of the 10th International Conference on Learning Representations (ICLR 2021)*. OpenReview, 2021.

Jiaming Song, Lantao Yu, Willie Neiswanger, and Stefano Ermon. A general recipe for likelihood-free bayesian optimization. In *International Conference on Machine Learning*, pp. 20384–20404. PMLR, 2022.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suria Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research*, 19, 2018.

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

Joe Staines and David Barber. Optimization by variational bounding. In *ESANN*, 2013.

Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating bayesian optimization for biological sequence design with denoising autoencoders. In *International Conference on Machine Learning*, pp. 20459–20478. PMLR, 2022.

Samuel Stanton, Robert Alberstein, Nathan Frey, Andrew Watkins, and Kyunghyun Cho. Closed-form test functions for biophysical sequence optimization algorithms. *arXiv preprint arXiv:2407.00236*, 2024.

Shikha Surana, Nathan Grinsztajn, Timothy Atkinson, Paul Duckworth, and Thomas D Barrett. Overconfident oracles: Limitations of in silico sequence design benchmarking. In *ICML 2024 AI for Science Workshop*, 2024.

Kevin Swersky, Yulia Rubanova, David Dohan, and Kevin Murphy. Amortized bayesian optimization over discrete spaces. In *Conference on Uncertainty in Artificial Intelligence*, pp. 769–778. PMLR, 2020.

Shion Takeno, Yu Inatsu, Masayuki Karasuyama, and Ichiro Takeuchi. Posterior Sampling-Based Bayesian Optimization with Tighter Bayesian Regret Bounds. In *Proceedings of the 41 st International Conference on Machine Learning (ICML 2024)*, volume 235, Vienna, Austria, 2024. PMLR.

Louis C Tiao, Aaron Klein, Matthias W Seeger, Edwin V Bonilla, Cedric Archambeau, and Fabio Ramos. Bore: Bayesian optimization by density-ratio estimation. In *International Conference on Machine Learning*, pp. 10289–10300. PMLR, 2021.

Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *International Conference on Machine Learning*, pp. 10358–10368. PMLR, 2021.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-bench: Benchmarks for data-driven offline model-based optimization. *CoRR*, abs/2202.08450, 2022.

Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33:11259–11272, 2020.

Sattar Vakili and Julia Olkhovskaya. Kernelized reinforcement learning with order optimal regret bounds. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Sattar Vakili, Kia Khezeli, and Victor Picheny. On information gain and regret bounds in gaussian process bandits. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 82–90. PMLR, 2021.

Hastagiri P Vanchinathan, Andreas Marfurt, Charles-Antoine Robelin, Donald Kossmann, and Andreas Krause. Discovering valuable items from massive data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1195–1204, 2015.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Fengxue Zhang, Jialin Song, James C Bowden, Alexander Ladd, Yisong Yue, Thomas Desautels, and Yuxin Chen. Learning regions of interest for Bayesian optimization with adaptive level-set estimation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 41579–41595. PMLR, 23–29 Jul 2023.

A ACRONYMS

ACRONYMS

BBO black-box optimization. 1, 5, 6, 10, 20–22

BO Bayesian optimization. 1, 3, 6, 35

BOPR Bayesian optimization with probabilistic reparameterisation. 5, 6, 19

BORE Bayesian optimization by density-ratio estimation. 4–8, 10, 17, 19–21

CbAS conditioning by adaptive sampling. 5–8, 10, 17, 19, 21

CPE class probability estimation. 4–10, 17–21

DbAS design by adaptive sampling. 5–8, 10, 17, 19, 21

EDA estimation of distribution algorithms. 5

ELBO evidence lower bound. 3, 4, 7

ES evolution strategies. 5

GP Gaussian process. 3–5, 8, 19, 28, 31

KL Kullback-Leibler. 3, 6, 8

LSE level set estimation. 6

LSO latent space optimization. 6

LSTM long short-term memory. 7, 8, 10, 17, 18, 21, 22

ML maximum likelihood. 7, 20, 21

NES natural evolution strategies. 5

PEX proximal exploration. 6–8, 10, 19, 20

PI probability of improvement. 3–5, 8, 19, 28

RNN recurrent neural network. 7, 17

VI variational inference. 1

VO variational optimization. 5

VSD variational search distributions. 1–10, 17, 19–22, 28, 31, 35

B EXPERIMENTAL DETAILS

We use three well established datasets; a green fluorescent protein (GFP) from Aequorea Victoria (Sarkisyan et al., 2016), an adeno-associated virus (AAV) Bryant et al. (2021); and DNA binding activity to a human transcription factor (TFBIND8) (Trabucco et al., 2022; Barrera et al., 2016). These datasets have been used variously by Brookes & Listgarten (2018); Brookes et al. (2019); Angermueller et al. (2019); Kirjner et al. (2024); Jain et al. (2022) among others. The GFP task is to maximize fluorescence, this protein consists of 238 amino acids, of which 237 can mutate. The AAV task us to maximize the genetic payload that can be delivered, and the associated protein has 28 amino acids, all of which can mutate. A complete combinatorial assessment is infeasible for these tasks, and so we use the convolution neural network oracle presented in Kirjner et al. (2024) as *in-silico* ground truth. TFBIND8 contains a complete combinatorial assessment of the effect of changing 8 nucleotides on binding to human transcription factor SIX6 REF R1 (Barrera et al., 2016). The dataset we use contains all 65536 sequences prepared by Trabucco et al. (2022). We also use two novel datasets from recent works that experimentally assess the (near) complete combinatorial space of short sequences. The first dataset measures the antibiotic resistance of Escherichia coli metabolic gene folA, which encodes dihydrofolate reductase (DHFR) (Papkou et al., 2023). Only a sub-sequence of this gene is varied (9 nucleic acids which encode 3 amino acids), and so a near-complete (99.7%) combinatorial scan is available. For variants that have no fitness (resistance) data available, we give a score of -1. The next dataset is near-complete combinatorial scan of four interacting amino acid residues near the active site of the enzyme tryptophan synthase (TrpB) (Johnston et al., 2024), with 159,129 unique sequences and fitness values, we use -0.2 for the missing fitness values (we do not use the authors' imputed values). These residues are explicitly shown to exhibit epistasis – or non-additive effects on catalytic function – which makes navigating this landscape a more interesting challenge from an optimization perspective. The properties of these datasets are presented in Table 3.

Dataset	$\lvert\mathcal{V}\rvert$	M	$\lvert\mathcal{X}_{\text{available}}\rvert$	$\lvert\mathcal{X}\rvert$
TFBIND8	4	8	65,536	65,536
TrpB	20	4	159,129	160,000
DHFR	4	9	261,333	262,144
AAV	20	28	42,340	20^{28}
GFP	20	237	51,715	20^{237}

Table 3: Alphabet size, sequence length, and number of availbale sequences for each of the datasets we use in this work.

B.1 VARIATIONAL DISTRIBUTIONS

In this section we summarize the main variational distribution architectures considered for VSD, BORE, CbAS and DbAS, and the sampling distributions for the Random baseline method. Somewhat surprisingly, we find that we obtain consistently good results for the biological sequence experiments using a simple independent (or mean-field) variational distribution,

$$q(\mathbf{x}|\phi) = \prod_{m=1}^{M} \text{Categ}(x_m|\text{softmax}(\phi_m)), \tag{20}$$

where $x_m \in \mathcal{V}$ and $\phi_m \in \mathbb{R}^{|\mathcal{V}|}$. However, this simple mean-field distribution was not capable of generating convincing handwritten digits. We have also tested a variety of transition variational distributions,

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}, \phi) = \prod_{m=1}^{M} \text{Categ}(x_{tm}|\text{softmax}(\text{NN}_m(\mathbf{x}_{t-1}, \phi))), \tag{21}$$

where $\text{NN}_m(\mathbf{x}_{t-1}, \phi)$ is the mth vector output of a neural network that takes a sequence from the previous round, $\mathbf{x}_{t-1}$, as input. We have implemented multiple neural net encoder/decoder architectures for $\text{NN}_m(\mathbf{x}_{t-1}, \phi)$, but we did not consider architectures of the form $\text{NN}_m(\phi)$ since the variational distribution in Equation 20 can always learn a $\phi_m = \text{NN}_m(\phi')$. We found that none of these transition architectures significantly outperformed the mean-field distribution (Equation 20) when it was initialized well (e.g. fit to the CPE training sequences), see Sec. C.3 for results. We also implemented auto-regressive variational distributions of the form,

$$q(\mathbf{x}|\phi) = \text{Categ}(x_1|\text{softmax}(\phi_1)) \prod_{m=2}^{M} q(x_m|x_{1:m-1}, \phi_{1:m}) \quad \text{where,} \tag{22}$$

$$q(x_m|x_{1:m-1}, \phi_{1:m}) = \begin{cases} \text{Categ}(x_m|\text{softmax}(\text{LSTM}(x_{m-1}, \phi_{m-1:m}))), \\ \text{Categ}(x_m|\text{softmax}(\text{DTransformer}(x_{1:m-1}, \phi_{1:m}))). \end{cases}$$

For a LSTM RNN and a decoder-only transformer with a causal mask, for the latter see Phuong & Hutter (2022, Algorithm 10 & Algorithm 14) for maximum likelihood training and sampling implementation details respectively. For the digits experiment, the LSTM uses 5 layers of dimension 128, and the transformer uses 8-attention heads, with 4 layers and a feed-forward network size of 256. For the AAV and GFP experiments, the LSTM uses 4 layers of dimension 32, and the transformer uses 4 attention heads for GFP, 2 attention heads for AAV, 1 layer, and a feed-forward network size of 64. Any larger than this, and we found these models would over-fit. We use additive positional encoding for both of these models.

B.2 CLASS PROBABILITY ESTIMATOR ARCHITECTURES

For the fitness landscape experiments on the smaller combinatorially complete datasets we use a two-hidden layer MLP, with an input embedding layer. The architecture is given in Figure 5 (a). For the larger dimensional AAV and GFP datasets, we use the convolutional architecture given in Figure 5 (b). Five fold cross validation is used to select the hyper parameters before the CPEs are trained on the whole training set for use in the subsequent experimental rounds. Model updates are performed by retraining on the whole query set.

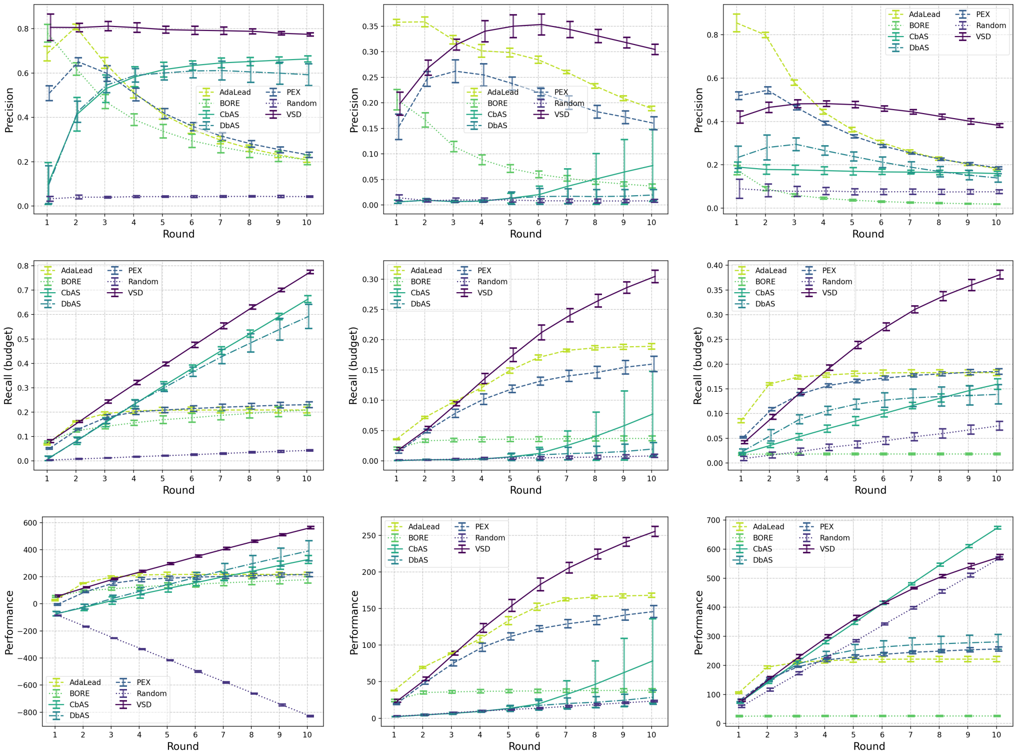
```
Sequential(
    Embedding(
        num_embeddings=A,
        embedding_dim=8
    ),
    Dropout(p=0.2),
    Flatten(),
    LeakyReLU(),
    Linear(
        in_features=8 * M,
        out_features=32
    ),
    LeakyReLU(),
    Linear(
        in_features=32,
        out_features=1
    ),
)
```

(a) MLP architecture

```
Sequential(
    Embedding(
        num_embeddings=A,
        embedding_dim=10
    ),
    Dropout(p=0.2),
    Conv1d(
        in_channels=10,
        out_channels=16,
        kernel_size=7,
    ),
    LeakyReLU(),
    MaxPool1d(
        kernel_size=2 or 4,
        stride=2 or 4,
    ),
    Conv1d(
        in_channels=16,
        out_channels=16,
        kernel_size=7,
    ),
    LeakyReLU(),
    MaxPool1d(
        kernel_size=2 or 4,
        stride=2 or 4,
    ),
    Flatten(),
    LazyLinear(
        out_features=128
    ),
    LeakyReLU(),
    Linear(
        in_features=128,
        out_features=1
    ),
)
```

(b) CNN architecture

Figure 5: CPE architectures used for the experiments in PyTorch syntax. `A` $= |\mathcal{V}|$, `M` $= M$, GFP uses a max pooling kernel size and stride of 4, all other datasets use 2.



(a) DHFR (b) TrpB (c) TFBIND8

Figure 6: Fitness landscape results using GP-PI. Precision (Equation 15), recall (Equation 16) and performance (Equation 17) – higher is better – for the combinatorially (near) complete datasets, DHFR and TrpB and TFBIND8. The random method is implemented by drawing B samples uniformly.

C ADDITIONAL EXPERIMENTAL RESULTS

C.1 FITNESS LANDSCAPES – GAUSSIAN PROCESS PROBABILITY OF IMPROVEMENT

Here we present additional fitness landscape experimental results, where we have used a GP as a surrogate model for $p(y|\mathbf{x}, \mathcal{D}_N)$ in conjunction with a complementary Normal CDF as the PI acquisition function. This is the one of the main frameworks supported by our theoretical analysis. VSD, DbAS, CbAS and BORE make use of the GP-PI acquisition function, and so BORE is BOPR in this instance since we are not using a CPE. PEX and AdaLead only use the GP surrogate, as per their original formulation. The GP uses a simple categorical kernel with automatic relevance determination from Balandat et al. (2020),

$$k(\mathbf{x}, \mathbf{x}') = \sigma \exp\left(-\frac{1}{M}\sum_{m=1}^{M} \frac{\mathbb{1}[x_m = x'_m]}{l_m}\right), \tag{23}$$

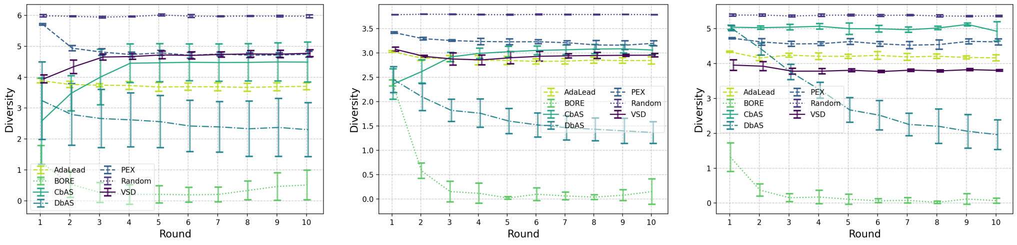
where σ and l_m are hyper-parameters controlling scale and length-scale respectively. See Figure 6 for the results.

C.2 DIVERSITY SCORES

The diversity of batches of candidates is a common thing to report in the literature, and to that end we present the diversity of our results here. We have taken the definition of diversity from (Jain et al., 2022) as,

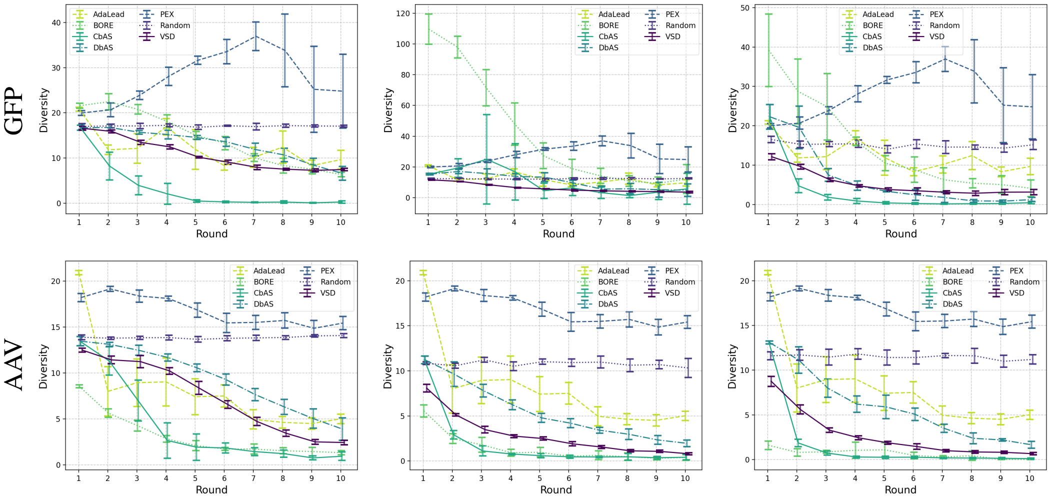
$$\text{Diversity}_t = \frac{1}{B(B-1)} \sum_{\mathbf{x}_i \in \mathcal{D}_{Bt}} \sum_{\mathbf{x}_j \in \mathcal{D}_{Bt} \setminus \{\mathbf{x}_i\}} \text{Lev}(\mathbf{x}_i, \mathbf{x}_j), \tag{24}$$

where Lev : $\mathcal{X} \times \mathcal{X} \to \mathbb{N}_0$ is the Levenshtein distance. We caution the reader as to the interpretation of these results however, as more diverse batches often do not lead to better performance, precision, recall or simple regret (as can be seen from the Random method results). Though insufficient diversity can also explain poor performance, as in the case of BORE. Results for the fitness landscape experiment are presented in Figure 7, and black-box optimization in Figure 8.



(a) DHFR (b) TrpB (c) TFBIND8

Figure 7: Fitness landscape diversity results. Higher is more diverse, as defined by Equation 24.



(a) Independent (b) LSTM (c) Transformer

Figure 8: Black-box optimization results for diversity on GFP and AAV with independent and auto-regressive variational distributions. Higher is more diverse, as defined by Equation 24. The PEX and AdaLead results are replicated between the plots, since they are unaffected by choice of variational distribution.

C.3 ABLATIONS – VARIATIONAL AND PRIOR DISTRIBUTIONS

In Figure 9 we present ablation results for VSD using different priors and variational distributions. We use the BBO experimental datasets for this task as they are higher-dimensional and so more sensitive to these design choices. We test the following prior and variational posterior distributions:

IU Independent categorical variational posterior distribution of the form in Equation 20, and a uniform prior distribution, $p(\mathbf{x}) = \prod_{m=1}^{M} \text{Categ}(x_m | \mathbf{1} \cdot |\mathcal{V}|^{-1})$.

I Independent categorical prior and variational posterior of the form in Equation 20. The prior is fit using ML on the initial CPE training data.

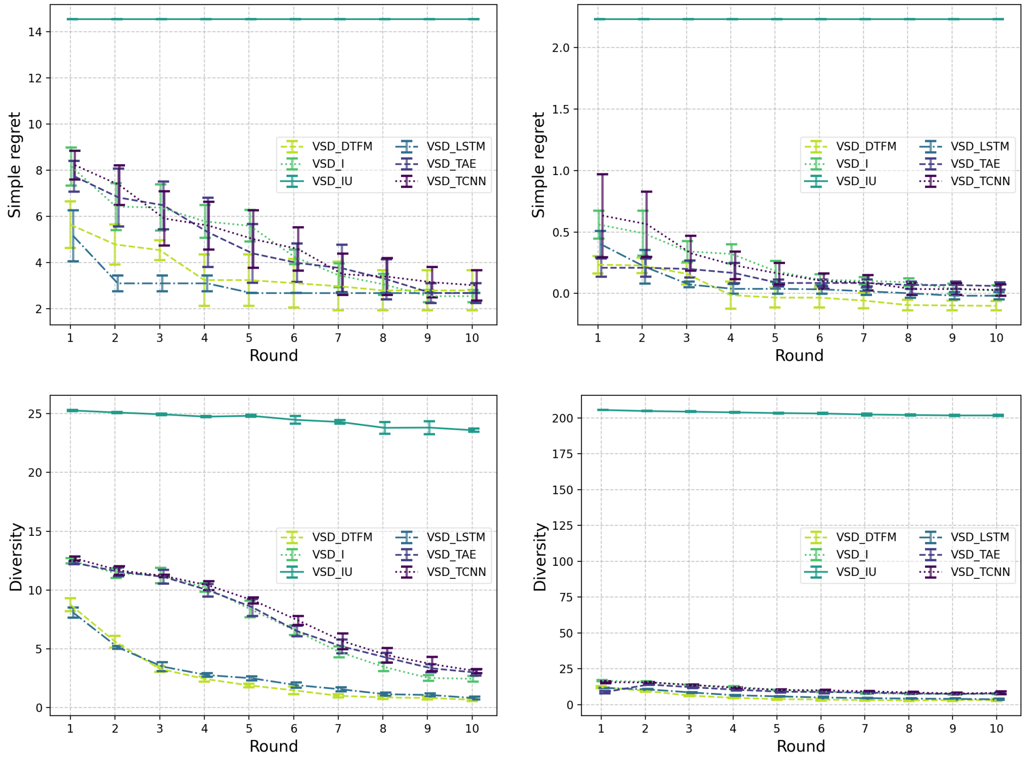
LSTM LSTM prior and variational posterior of the form Equation 22. The prior is fit using ML on the initial CPE training data.

DTFM Decoder-only causal transformer prior and variational posterior of the form Equation 22. The prior is fit using ML on the initial CPE training data.

TAE Independent categorical prior and a transition-style auto-encoder variational posterior of the form Equation 21, where we use two-hidden layer MLPs for the encoder and decoder. The prior is fit using ML on the initial CPE training data.

TCNN Independent categorical prior and a transition-style convolutional auto-encoder variational posterior of the form Equation 21, where we use a convolutional encoder, and transpose convolutional decoder. The prior is fit using ML on the initial CPE training data.

We use the informed-independent priors with the transition variational distributions since they are somewhat counter-intuitive to use as priors themselves.



(a) AAV (b) GFP

Figure 9: Ablation results for the AAV and GFP BBO experiments. VSD is trialed with different prior and variational posterior combinations, "I" indicates a simple independent informed prior and posterior, "IU" is the same but with a uniform prior, "LSTM" and "DTFM" are the LSTM and decoder only transformer prior and posteriors, "TCNN" and "TAE" are transition convolutional encoder-decoder and auto-encoder posteriors, with informed independent priors. See text for details.

From Figure 9 we can see that while using an uninformative prior works in the lower-dimensional fitness landscape experiments, using an informative prior is crucial for these higher dimensional problems. We found a similar result when using this uninformative prior with CbAS, or using a uniform initialization with DbAS and BORE. The methods are not able to make any significant progress within the experimental budget given. The independent and transition variational distributions achieve similar performance, whereas the auto-regressive models generally outperform all others. This is because of the LSTM and transformer's superior generalization performance when generating sequences – measured both when training the priors (on held-out sequences) and during VSD adaptation.

C.4 EHRLICH FUNCTION BBO RESULTS

We now examine VSD's performance for BBO on the the recently proposed Ehrlich test functions (Stanton et al., 2024). These functions are challenging closed form biological analogues, specifically designed to test BBO methods on high dimensional sequence design tasks without having to resort to physical experimentation or machine learning oracles. We use the POLI and POLI-BASELINES software package for the benchmark and baselines (González-Duque et al., 2024). This experiment uses an Ehrlich function with a sequence length of 32, 4 motifs, a motif length of 4 and a quantization of 4. $B = 128$, $T = 32$ and 128 random samples of the function are used for $\mathcal{D}_N$. As before, 5 different random seeds are used for these trials, and for VSD we use an the same scheduling function for τ_t as in Equation 18, with $p_0 = 0.75$ and $\eta = 0.9$ (so $p_T = 0.99$).

The performance measure of interest is simple regret, Equation 19, and we benchmark VSD with independent, LSTM and transformer variational distributions against a genetic algorithm baseline (González-Duque et al., 2024) and the recently developed BBO solver LaMBO-2 (Gruver et al., 2024). Results are presented in Figure 10 for all individual black box evaluations. VSD uses the CNN, LSTM and transformer architecture as in the GFP Sec. 4.3 experiment, but with 3 hidden layers for the LSTM. We see that VSD with LSTM and transformer variational distributions find the fittest sequences, the genetic algorithm and VSD with an independent variational distribution make no progress, and LaMBO-2 also finds fit sequences, but is not as successful asVSD with LSTM or transformer variational distributions.

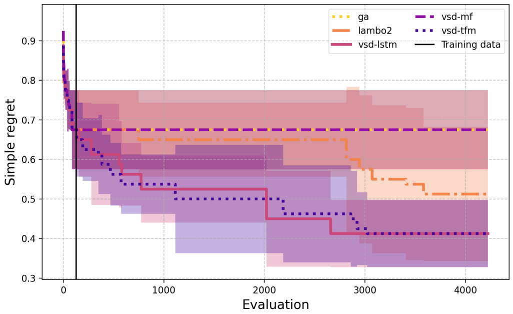


Figure 10: Ehrlich function $M = 32$ BBO results. VSD with different variational distributions compared against 'ga' – genetic algorithm and 'lambo2' – LaMBO-2 baselines. The vertical black line denotes the randomly sampled training data, and the x-axis indexes individual black-box evaluations. Mean and 1-standard deviation results are plotted for 5 random seeds.

D THEORETICAL ANALYSIS FOR GP-BASED CPES

In this section, we present theoretical results concerning VSD and its estimates when equipped with Gaussian process regression models (Rasmussen & Williams, 2006). We show that VSD sampling distributions converge to a target distribution that characterizes the level set given by τ. The approximation error mainly depends on the predictive uncertainty of the probabilistic model with respect to the true underlying function $f_\bullet$. For the analysis, we will assume that $f_\bullet$ is drawn from a Gaussian process, i.e., $f_\bullet \sim \mathcal{GP}(0, k)$, with a positive-semidefinite covariance (or kernel) function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. In this case, we can show that the predictive uncertainty of the model converges (in probability) to zero as the number of observations grows. From this result, we prove asymptotic convergence guarantees for VSD equipped with GP-PI-based CPEs. These results form the basis for our analysis of CPEs based on neural networks (Appendix E).

D.1 GAUSSIAN PROCESS POSTERIOR

Let $f_\bullet \sim \mathcal{GP}(0, k)$ be a zero-mean Gaussian process with a positive-semidefinite covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Assume that we are given a set $\mathcal{D}_N := \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ of $N \geq 1$ observations $y_i = f_\bullet(\mathbf{x}_i) + \epsilon_i$, where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ and $\mathbf{x}_i \in \mathcal{X}$. The GP posterior predictive distribution at any

$\mathbf{x} \in \mathcal{X}$ is then given by (Rasmussen & Williams, 2006):

$$f_{\bullet}(\mathbf{x})|\mathcal{D}_N \sim \mathcal{N}\big(\mu_N(\mathbf{x}), \sigma_N^2(\mathbf{x})\big) \tag{25}$$

$$\mu_N(\mathbf{x}) = \mathbf{k}_N(\mathbf{x})^\top (\mathbf{K}_N + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{y}_N \tag{26}$$

$$k_N(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_N(\mathbf{x})^\top (\mathbf{K}_N + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{k}_N(\mathbf{x}') \tag{27}$$

$$\sigma_N^2(\mathbf{x}) = k_N(\mathbf{x}, \mathbf{x}), \tag{28}$$

where $\mathbf{k}_N(\mathbf{x}) := [k(\mathbf{x}, \mathbf{x}_i)]_{i=1}^N \in \mathbb{R}^N$, $\mathbf{K}_N := [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^{N,N} \in \mathbb{R}^{N \times N}$, and $\mathbf{y}_N := [y_i]_{i=1}^N \in \mathbb{R}^N$.

Batch size. In the following, we will assume a batch of size $B = 1$ to keep the proofs simple. With this assumption, at every iteration $t \geq 1$, we have $N = t$ observations available in the dataset. We would, however, like to emphasize that sampling a batch of multiple observations, instead of a single observation, per iteration should only improve the convergence rates by a constant (batch-size-dependent) multiplicative factor. Therefore, our results remain valid as an upper bound for the convergence rates of VSD in the batch setting.

D.2 BACKGROUND

We will consider an underlying probability space $(\Omega, \mathfrak{A}, \mathbb{P})$, where Ω is the sample space, $\mathfrak{A}$ denotes the σ-algebra of events, and $\mathbb{P}$ is a probability measure. For any event $\mathcal{A} \in \mathfrak{A}$, we have that $\mathbb{P}[\mathcal{A}] \in [0, 1]$ quantifies the probability of that event. For events involving a random variable, e.g., $\chi : (\Omega, \mathfrak{A}) \to (\mathbb{R}, \mathfrak{B}_\mathbb{R})$, where $\mathfrak{B}_\mathbb{R}$ denotes the Borel σ-algebra of the real line with its usual topology, we will let:

$$\mathbb{P}[\chi > 0] = \mathbb{P}[\{\omega \in \Omega : \chi(\omega) > 0\}] \,. \tag{29}$$

We will also use conditional expectations, i.e., given a σ-sub-algebra $\mathfrak{S}$ of $\mathfrak{A}$, the conditional expectation $\mathbb{E}[\chi|\mathfrak{S}]$ is a $\mathfrak{S}$-measurable random variable such that:

$$\forall \mathcal{A} \in \mathfrak{S}\,, \quad \int_\mathcal{A} \mathbb{E}[\chi|\mathfrak{S}] \,\mathrm{d}\mathbb{P} = \int_\mathcal{A} \chi \,\mathrm{d}\mathbb{P} = \mathbb{E}[\chi|\mathcal{A}]\,. \tag{30}$$

We will denote by $\{\mathfrak{F}_t\}_{t=0}^\infty$ an increasing filtration on $\mathfrak{A}$. For instance, we could set $\mathfrak{F}_t$ as the σ-algebra generated by the random variables in the algorithm (i.e., the candidates, target observations, etc.) at time t. For more details on the measure-theoretic definition of probability, we refer the reader to classic textbooks in the area (e.g. Bauer, 1981; Durrett, 2019)

We will use the following well known notation for asymptotic convergence results. For a given strictly positive function $g : \mathbb{N} \to \mathbb{R}$, we define $\mathcal{O}(g(t))$ as the set of functions asymptotically bounded by g (up to a constant factor) as:

$$\mathcal{O}(g(t)) := \left\{ h : \mathbb{N} \to \mathbb{R} \;\middle|\; \limsup_{t \to \infty} \frac{|h(t)|}{g(t)} < \infty \right\}, \tag{31}$$

and for convergence in probability we use its stochastic counterpart:

$$\mathcal{O}_\mathbb{P}(g(t)) := \left\{ \rho : \mathbb{N} \times (\Omega, \mathfrak{A}) \to (\mathbb{R}, \mathfrak{B}_\mathbb{R}) \;\middle|\; \lim_{C \to \infty} \limsup_{t \to \infty} \mathbb{P}\left[\frac{|\rho(t)|}{g(t)} > C \right] = 0 \right\}, \tag{32}$$

which is equivalent to:

$$\forall \varepsilon > 0, \quad \exists C_\varepsilon \in (0, \infty) : \quad \mathbb{P}[|\rho_t| > C_\varepsilon] \leq \varepsilon, \quad \forall t \geq T_\varepsilon\,, \tag{33}$$

for some $T_\varepsilon \in \mathbb{N}$. For almost sure convergence, we may also say that a sequence of random variables ρ_t, $t \in \mathbb{N}$, is almost surely $\mathcal{O}(g(t))$ if $\mathbb{P}[\rho_t \in \mathcal{O}(g(t))] = 1$. A deeper overview on these notations and their properties can be found in García-Portugués (2024).

D.3 AUXILIARY RESULTS

We start with a few technical results which will form the basis for our derivations. The following recursive relations allow us to derive convergence rates for the variance of a GP posterior by analyzing how much it reduces per iteration.

Lemma D.1 (Chowdhury & Gopalan (2017, Appendix F)). *The posterior mean and covariance functions of a Gaussian process given $t \geq 1$ observations obey the following recursive identities:*

$$\mu_t(\mathbf{x}) = \mu_{t-1}(\mathbf{x}) + \frac{k(\mathbf{x}, \mathbf{x}_t)}{\sigma_\epsilon^2 + \sigma_{t-1}^2(\mathbf{x}_t)}(y_t - \mu_{t-1}(\mathbf{x})) \tag{34}$$

$$k_t(\mathbf{x}, \mathbf{x}') = k_{t-1}(\mathbf{x}, \mathbf{x}') - \frac{k_{t-1}(\mathbf{x}, \mathbf{x}_t)k_{t-1}(\mathbf{x}_t, \mathbf{x}')}{\sigma_\epsilon^2 + \sigma_{t-1}^2(\mathbf{x}_t)} \tag{35}$$

$$\sigma_t^2(\mathbf{x}) = \sigma_{t-1}^2(\mathbf{x}) - \frac{k_{t-1}^2(\mathbf{x}, \mathbf{x}_t)}{\sigma_\epsilon^2 + \sigma_{t-1}^2(\mathbf{x}_t)}, \tag{36}$$

for $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$.

We will also make use of the following version of the second Borel-Cantelli lemma adapted from Durrett (2019, Thr. 4.5.5) and its original statement in Dubins & Freedman (1965).

Lemma D.2 (Second Borel-Cantelli lemma). *Let $\{\mathcal{A}_t\}_{t=1}^\infty$ be a sequence of events where $\mathcal{A}_t \in \mathfrak{F}_t$, for all $t \in \mathbb{N}$, and let $\chi_t : \omega \mapsto \mathbb{1}[\omega \in \mathcal{A}_t]$, for $\omega \in \Omega$. Then the following holds with probability 1:*

$$\lim_{T\to\infty} \frac{\sum_{t=1}^T \chi_t}{\sum_{t=1}^T \mathbb{P}[\mathcal{A}_t|\mathfrak{F}_{t-1}]} = L < \infty, \tag{37}$$

assuming $\mathbb{P}[\mathcal{A}_1|\mathfrak{F}_0] > 0$. In addition, if $\lim_{T\to\infty} \sum_{t=1}^T \mathbb{P}[\mathcal{A}_t|\mathfrak{F}_{t-1}] = \infty$, then $L = 1$.

The next result provides us with an upper bound on the posterior variance of a Gaussian process which is valid for any covariance function.

Lemma D.3. *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be any positive-semidefinite kernel on $\mathcal{X}$, and let $\tilde{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel defined as:*

$$\tilde{k}(\mathbf{x}, \mathbf{x}') = \begin{cases} k(\mathbf{x}, \mathbf{x}), & \mathbf{x} = \mathbf{x}' \\ 0, & \mathbf{x} \neq \mathbf{x}', \end{cases} \tag{38}$$

for $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. Given any set of observations $\{\mathbf{x}_i, y_i\}_{i=1}^t$, for $t \geq 1$, denote by σ_t^2 the predictive variance of a GP model with prior covariance given by k, and let $\tilde{\sigma}_t^2$ denote the predictive variance of a GP model configured with $\tilde{k}$ as prior covariance function, where both models are given the same set of observations. Then the following holds for all $t \geq 0$:

$$\sigma_t^2(\mathbf{x}) \leq \tilde{\sigma}_t^2(\mathbf{x}) = \frac{\sigma_\epsilon^2 \tilde{\sigma}_0^2(\mathbf{x})}{\sigma_\epsilon^2 + N_t(\mathbf{x})\tilde{\sigma}_0^2(\mathbf{x})}, \quad \forall \mathbf{x} \in \mathcal{X}, \tag{39}$$

where $N_t(\mathbf{x})$ denotes the number of observations at $\mathbf{x}$, and $\tilde{\sigma}_0^2(\mathbf{x}) = \sigma_0^2(\mathbf{x}) := k(\mathbf{x}, \mathbf{x})$, for $\mathbf{x} \in \mathcal{X}$.

Proof. It is not hard to show that $\tilde{k}$ defines a valid positive-semidefinite covariance function whenever k is positive semidefinite. We will then focus on proving the main statement by an induction argument. The proof that the statement holds for the base case at $t = 0$ is trivial given the definition:

$$\sigma_0^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) = \tilde{k}(\mathbf{x}, \mathbf{x}) = \tilde{\sigma}_0^2(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{X}. \tag{40}$$

Now assume that, for a given $t > 0$, it holds that $\sigma_t^2(\mathbf{x}) \leq \tilde{\sigma}_t^2(\mathbf{x})$, for all $\mathbf{x} \in \mathcal{X}$. We will then check if the inequality remains valid at $t + 1$. By Lemma D.1, we have that:

$$\sigma_{t+1}^2(\mathbf{x}) = \sigma_t^2(\mathbf{x}) - \frac{k_t^2(\mathbf{x}, \mathbf{x}_{t+1})}{\sigma_t^2(\mathbf{x}_{t+1}) + \sigma_\epsilon^2} \tag{41}$$

For any $\mathbf{x} \in \mathcal{X}$ such that $\mathbf{x} \neq \mathbf{x}_{t+1}$, we know that $\tilde{k}_t(\mathbf{x}, \mathbf{x}_{t+1}) \geq 0$, so that (again by Lemma D.1):

$$\tilde{k}_t^2(\mathbf{x}, \mathbf{x}_{t+1}) \leq \tilde{k}^2(\mathbf{x}, \mathbf{x}_{t+1}) = 0, \tag{42}$$

which shows that:

$$\forall \mathbf{x} \neq \mathbf{x}_{t+1}, \quad \sigma_{t+1}^2(\mathbf{x}) \leq \sigma_t^2(\mathbf{x}) \leq \tilde{\sigma}_t^2(\mathbf{x}) = \tilde{\sigma}_{t+1}^2(\mathbf{x}). \tag{43}$$

At $\mathbf{x} = \mathbf{x}_{t+1}$, we can rewrite $\sigma^2_{t+1}(\mathbf{x}) = \sigma^2_{t+1}(\mathbf{x}_{t+1})$ as:

$$\sigma^2_{t+1}(\mathbf{x}_{t+1}) = \frac{\sigma^2_\epsilon \sigma^2_t(\mathbf{x}_{t+1})}{\sigma^2_t(\mathbf{x}_{t+1}) + \sigma^2_\epsilon}\,. \tag{44}$$

We then check the difference:

$$\begin{aligned}
\sigma^2_{t+1}(\mathbf{x}_{t+1}) - \tilde{\sigma}^2_{t+1}(\mathbf{x}_{t+1}) &= \frac{\sigma^2_\epsilon \sigma^2_t(\mathbf{x}_{t+1})}{\sigma^2_t(\mathbf{x}_{t+1}) + \sigma^2_\epsilon} - \frac{\sigma^2_\epsilon \tilde{\sigma}^2_t(\mathbf{x}_{t+1})}{\tilde{\sigma}^2_t(\mathbf{x}_{t+1}) + \sigma^2_\epsilon} \\
&= \frac{\sigma^2_\epsilon \sigma^2_t(\mathbf{x}_{t+1})(\tilde{\sigma}^2_t(\mathbf{x}_{t+1}) + \sigma^2_\epsilon) - \sigma^2_\epsilon \tilde{\sigma}^2_t(\mathbf{x}_{t+1})(\sigma^2_t(\mathbf{x}_{t+1}) + \sigma^2_\epsilon)}{(\sigma^2_t(\mathbf{x}_{t+1}) + \sigma^2_\epsilon)(\tilde{\sigma}^2_t(\mathbf{x}_{t+1}) + \sigma^2_\epsilon)} \\
&= \frac{\sigma^4_\epsilon(\sigma^2_t(\mathbf{x}_{t+1}) - \tilde{\sigma}^2_t(\mathbf{x}_{t+1}))}{(\sigma^2_t(\mathbf{x}_{t+1}) + \sigma^2_\epsilon)(\tilde{\sigma}^2_t(\mathbf{x}_{t+1}) + \sigma^2_\epsilon)} \\
&\leq 0\,,
\end{aligned} \tag{45}$$

since $\sigma^2_t(\mathbf{x}_{t+1}) \leq \tilde{\sigma}^2_t(\mathbf{x}_{t+1})$ by our assumption for time t. Therefore, we have shown that:

$$\sigma^2_t(\mathbf{x}) \leq \tilde{\sigma}^2_t(\mathbf{x}) \implies \sigma^2_{t+1}(\mathbf{x}) \leq \tilde{\sigma}^2_{t+1}(\mathbf{x})\,, \quad \forall \mathbf{x} \in \mathcal{X}\,. \tag{46}$$

From the conclusion above and the base case, the inequality in the main result follows by induction.

Now we derive an explicit form for $\tilde{\sigma}^2_t$. Note that this case corresponds to an independent Gaussian model, i.e., $f_\bullet(\mathbf{x}) \perp\!\!\!\perp f_\bullet(\mathbf{x}')$ whenever $\mathbf{x} \neq \mathbf{x}'$, for $f_\bullet \sim \mathcal{GP}(0, \tilde{k})$. For any $t \geq 1$, this model's predictive variance at any $\mathbf{x} \in \mathcal{X}$ is given by:

$$\tilde{\sigma}^2_t(\mathbf{x}) = \begin{cases} \tilde{\sigma}^2_{t-1}(\mathbf{x}), & \mathbf{x} \neq \mathbf{x}_t \\ \dfrac{\sigma^2_\epsilon \tilde{\sigma}^2_{t-1}(\mathbf{x}_t)}{\sigma^2_\epsilon + \tilde{\sigma}^2_{t-1}(\mathbf{x}_t)} = \left(\dfrac{1}{\tilde{\sigma}^2_{t-1}(\mathbf{x}_t)} + \dfrac{1}{\sigma^2_\epsilon}\right)^{-1}, & \mathbf{x} = \mathbf{x}_t \end{cases} \tag{47}$$

Looking at the reciprocal, we have that:

$$\forall t \geq 1, \quad \frac{1}{\tilde{\sigma}^2_t(\mathbf{x})} = \frac{1}{\tilde{\sigma}^2_{t-1}(\mathbf{x}_t)} + \frac{\mathbb{1}[\mathbf{x}_t = \mathbf{x}]}{\sigma^2_\epsilon}, \quad \forall \mathbf{x} \in \mathcal{X}. \tag{48}$$

Therefore, every observation at $\mathbf{x}$ is simply adding a factor of σ_ϵ^{-2} to $\tilde{\sigma}_t^{-2}(\mathbf{x})$. Unwrapping this recursion leads us to:

$$\forall t \geq 1, \quad \frac{1}{\tilde{\sigma}^2_t(\mathbf{x})} = \frac{1}{\tilde{\sigma}^2_0(\mathbf{x})} + \frac{1}{\sigma^2_\epsilon}\sum_{i=1}^{t} \mathbb{1}[\mathbf{x}_i = \mathbf{x}]\,, \quad \forall \mathbf{x} \in \mathcal{X}\,. \tag{49}$$

The result in Lemma D.3 then follows as the reciprocal of the above, which concludes the proof. ☐

Lemma D.4. *Let $f_\bullet \sim \mathcal{GP}(0, k)$ for a given $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where $\sigma^2_{\mathcal{X}} := \sup_{\mathbf{x}\in\mathcal{X}} k(\mathbf{x}, \mathbf{x}) < \infty$, and $|\mathcal{X}| < \infty$. Then $f_\bullet$ is almost surely bounded, and:*

$$\mathbb{E}\left[\sup_{\mathbf{x}\in\mathcal{X}} |f_\bullet(\mathbf{x})|\right] \leq \sigma_{\mathcal{X}}\sqrt{2\log|\mathcal{X}|}\,. \tag{50}$$

Proof. The result follows by an application of a concentration inequality for the maximum of a finite collection of sub-Gaussian random variables (Boucheron et al., 2013, Sec. 2.5). Note that $\{f_\bullet(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$ is a collection of $|\mathcal{X}|$ Gaussian, and therefore sub-Gaussian, random variables with sub-Gaussian parameter given by $\sigma^2_{\mathcal{X}} \geq \sigma^2_t(\mathbf{x})$, for all $\mathcal{X}$. Applying the maximal inequality for a finite collection sub-Gaussian random variables (Boucheron et al., 2013, Thr. 2.5), we have that:

$$\mathbb{E}\left[\max_{\mathbf{x}\in\mathcal{X}} f_\bullet(\mathbf{x})\right] \leq \sigma_{\mathcal{X}}\sqrt{2\log|\mathcal{X}|} < \infty\,. \tag{51}$$

By symmetry, we know that $-f_\bullet(\mathbf{x})$ is also sub-Gaussian with the same parameter, so that the bound remains valid for $\max_{\mathbf{x}\in\mathcal{X}} -f_\bullet(\mathbf{x})$. As a consequence, the expected value of the maximum of $|f_\bullet(\mathbf{x})|$ is upper bounded by the same constant. On a finite set, the maximum and the supremum coincide. As the expected value of the supremum is finite, the supremum must be almost surely finite by Markov's inequality, and therefore $f_\bullet$ is almost surely bounded. ☐

D.4 ASYMPTOTIC CONVERGENCE

The main assumption we will be working with in this section is the following.

Assumption D.1. *The objective function is a sample from a Gaussian process* $f_{\blacksquare} \sim \mathcal{GP}(0, k)$, *where* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ *is a bounded positive-semidefinite kernel on* $\mathcal{X}$.

The next result allows us to derive a convergence rate for the posterior variance of a GP as a function of the sampling probabilities. This result might also be useful by itself for other sampling problems involving GP-based approximations.

Lemma D.5. *Let* $\{\mathbf{x}_t\}_{t \geq 1}$ *be a sequence of* $\mathcal{X}$*-valued random variables adapted to the filtration* $\{\mathfrak{F}_t\}_{t \geq 1}$. *For a given* $\mathbf{x} \in \mathcal{X}$, *assume that the following holds:*

$$\exists T_* \in \mathbb{N} : \quad \forall T \geq T_*, \quad \sum_{t=1}^{T} \mathbb{P}[\mathbf{x}_t = \mathbf{x} \mid \mathfrak{F}_{t-1}] \geq B_T > 0 \,, \tag{52}$$

for a some sequence of lower bounds $\{B_t\}_{t \in \mathbb{N}}$. *Then, under Assumption D.1, given observations at* $\{\mathbf{x}_i\}_{i=1}^{t}$, *the following holds with probability 1:*

$$\sigma_t^2(\mathbf{x}) \in \mathcal{O}(B_t^{-1}). \tag{53}$$

In addition, if $B_t \to \infty$, *then* $\lim_{t \to \infty} B_t \sigma_t^2(\mathbf{x}) \leq \sigma_\epsilon^2$.

Proof. At any iteration t, the posterior variance σ_t^2 of a GP model is upper bounded by a worst case assumption of no correlation between observations (see Lemma D.3). In this case, we have that:

$$\sigma_t^2(\mathbf{x}) \leq \tilde{\sigma}_t^2(\mathbf{x}) = \frac{\sigma_\epsilon^2 \tilde{\sigma}_0^2(\mathbf{x})}{\sigma_\epsilon^2 + N_t \tilde{\sigma}_0^2(\mathbf{x})} \,, \tag{54}$$

where $\tilde{\sigma}_0^2(\mathbf{x}) := \tilde{k}(\mathbf{x}, \mathbf{x}) = k(\mathbf{x}, \mathbf{x})$, and $N_t := N_t(\mathbf{x}) \leq t$ denotes the total number of observations taken at $\mathbf{x}$ as of iteration t. Without loss of generality, assume that $\tilde{\sigma}_0^2(\mathbf{x}) = 1$.

The only random variable to be bounded in Equation 54 is N_t. Let $\chi_t := \mathbb{1}[\mathbf{x}_t = \mathbf{x}]$, so that:

$$N_t = \sum_{i=1}^{t} \chi_i = \sum_{i=1}^{t} \mathbb{1}[\mathbf{x}_t = \mathbf{x}] \,, \quad t \geq 1. \tag{55}$$

We now apply the second Borel-Cantelli lemma (Lemma D.2) to N_t. Namely, let $\widehat{N}_t$ denote the sum of conditional expectations of $\{\chi_i\}_{i=1}^{t}$ given available data, i.e.:

$$\widehat{N}_t := \sum_{i=1}^{t} \mathbb{E}[\chi_i \mid \mathfrak{F}_{i-1}] = \sum_{i=1}^{t} \mathbb{E}[\mathbb{1}[\mathbf{x}_t = \mathbf{x}] \mid \mathfrak{F}_{i-1}] = \sum_{i=1}^{t} \mathbb{P}[\mathbf{x}_i = \mathbf{x} \mid \mathfrak{F}_{i-1}] \,. \tag{56}$$

By Lemma D.2, we know that the following holds for some $L \in \mathbb{R}$:

$$\lim_{t \to \infty} \frac{N_t}{\widehat{N}_t} = L < \infty \,. \tag{57}$$

Hence, N_t is asymptotically equivalent to $\widehat{N}_t$. Applying this fact to $\tilde{\sigma}_t^2$, we have that:

$$\begin{aligned}
\lim_{t \to \infty} B_t \tilde{\sigma}_t^2(\mathbf{x}) &= \lim_{t \to \infty} \frac{B_t \sigma_\epsilon^2}{\sigma_\epsilon^2 + N_t} \\
&= \lim_{t \to \infty} \frac{B_t \sigma_\epsilon^2}{\sigma_\epsilon^2 + L\widehat{N}_t} \\
&\leq \lim_{t \to \infty} \frac{B_t \sigma_\epsilon^2}{\sigma_\epsilon^2 + LB_t} \\
&\leq \frac{1}{L} \lim_{t \to \infty} \min\{LB_t, \sigma_\epsilon^2\} \\
&< \infty \,,
\end{aligned} \tag{58}$$

which holds with probability 1. Lastly, note that, if $B_t \to \infty$, then $L = 1$ by Lemma D.2, and the last limit above becomes σ_ϵ^2. The main result then follows by an application of Lemma D.3 and the definition of the big-$\mathcal{O}$ notation (see Equation 31).[1] $\square$

We assume a finite search space, which is the case for spaces of discrete sequences of bounded length. However, we conjecture that our results can be extended to continuous or mixed discrete-continuous search spaces via a discretization argument under further assumptions on the kernel k (e.g., ensuring that $f_\bullet$ is Lipschitz continuous, as in Srinivas et al. (2010)).

Assumption D.2. *The search space $\mathcal{X}$ is finite, $|\mathcal{X}| < \infty$.*

We assume that our family of variational distributions is rich enough to be able to represent the PI-based distribution $p(\mathbf{x}|y > \tau_t, \mathcal{D}_t)$, which is the optimum of our variational objective when the optimal classifier is given by GP-PI. Although this assumption could be seen as strong, note that, due to Gaussian noise, the classification probability $p(y > \tau_t|\mathbf{x}, \mathcal{D}_t)$ should be a reasonably smooth function of $\mathbf{x}$, which facilitates the approximation of the resulting posterior by a generative model.

Assumption D.3. *For every $t \geq 0$, $p(\mathbf{x}|y > \tau_t, \mathcal{D}_t)$ is a member of the variational family, i.e.:*

$$\exists \phi_t^* : \quad \mathbb{D}[q(\mathbf{x}|\phi_t^*) \| p(\mathbf{x}|y > \tau_t, \mathcal{D}_t)] = 0. \tag{59}$$

The next assumption is a technical one to ensure that the thresholds will not diverge to infinity.

Assumption D.4. *The sequence of thresholds is almost surely bounded:*[2]

$$\sup_{t\in\mathbb{N}} |\tau_t| \leq \tau_* < \infty\,. \tag{60}$$

We can now state our main result regarding the GP-based approximations learned by VSD.

Theorem D.1. *Let assumptions D.1 to D.4 hold. Then the following holds with probability 1 for VSD equipped with GP-PI:*

$$\sigma_t^2(\mathbf{x}) \in \mathcal{O}(t^{-1})\,, \tag{61}$$

at every $\mathbf{x} \in \mathcal{X}$ such that $p(\mathbf{x}) > 0$.

Proof. Let $\ell_t(\mathbf{x}) := p(y > \tau_t|\mathbf{x}, \mathcal{D}_t)$. For any given $\mathbf{x} \in \mathcal{X}$ where $p(\mathbf{x}) > 0$, by Assumption D.2, we have that the next candidate will be sampled according to:

$$\begin{aligned}\forall t \geq 0, \quad \mathbb{P}[\mathbf{x}_{t+1} = \mathbf{x} \mid \mathfrak{F}_t] &= p(\mathbf{x}|y > \tau_t, \mathcal{D}_t)\\ &= \frac{\ell_t(\mathbf{x})p(\mathbf{x})}{\mathbb{E}_{p(\mathbf{x})}[\ell_t(\mathbf{x})]}\\ &\geq \ell_t(\mathbf{x})p(\mathbf{x}),\end{aligned} \tag{62}$$

where we used the fact that $\mathbb{E}_{p(\mathbf{x})}[\ell_t(\mathbf{x})] \leq 1$, since $\ell_t(\mathbf{x}) \leq 1$, for all $\mathbf{x} \in \mathcal{X}$. As $p(\mathbf{x}) > 0$, we only have to derive a lower bound on $\ell_t(\mathbf{x})$ to apply Lemma D.5 and derive a convergence rate.

A lower bound on $\ell_t(\mathbf{x})$ is given by:

$$\forall t \geq 0, \quad \ell_t(\mathbf{x}) = \Psi\left(\frac{\mu_t(\mathbf{x}) - \tau_t}{\sqrt{\sigma_t^2(\mathbf{x}) + \sigma_\epsilon^2}}\right) \geq \Psi\left(-\frac{\|\mu_t\|_\infty + \tau_*}{\sigma_\epsilon}\right), \tag{63}$$

where $\Psi(\cdot)$ denotes the cumulative distribution function of a standard normal random variable, and $\|\cdot\|_\infty$ denotes the essential supremum of a function under $\mathbb{P}$ (the probability measure of the underlying abstract probability space). Therefore, if $\lim_{t\to\infty}\|\mu_t\|_\infty < \infty$, we will have that $\lim_{t\to\infty} \ell_t(\mathbf{x}) > 0$, and the sum in Lemma D.5 will diverge.

By Jensen's inequality for conditional expectations, we have that:

$$\forall t \geq 0, \quad \|\mu_t\|_\infty = \|\mathbb{E}[f_\bullet \mid \mathfrak{F}_t]\|_\infty \leq \mathbb{E}[\|f_\bullet\|_\infty \mid \mathfrak{F}_t]. \tag{64}$$

---

[1] Recall that for convergent sequences $\lim$ and $\limsup$ coincide.

[2] We do not require τ_* to be known, only finite.

As $\mathbb{E}[\mathbb{E}[\|f_\bullet\|_\infty \mid \mathfrak{F}_t]] = \mathbb{E}[\|f_\bullet\|_\infty] < \infty$ (cf. Lemma D.4), an application of Markov's inequality implies that:

$$\lim_{a\to\infty} \mathbb{P}[\mathbb{E}[\|f_\bullet\|_\infty \mid \mathfrak{F}_t] \geq a] \leq \lim_{a\to\infty} \frac{1}{a}\mathbb{E}[\|f_\bullet\|_\infty] = 0. \tag{65}$$

Furthermore, $m_t := \mathbb{E}[\|f_\bullet\|_\infty \mid \mathfrak{F}_t]$ also defines a non-negative martingale, and by the martingale convergence theorem (Durrett, 2019, Thr. 4.2.11), $\lim_{t\to\infty} m_t = m_\infty := \mathbb{E}[\|f_\bullet\|_\infty \mid \mathfrak{F}_\infty]$ is well defined and $\mathbb{E}[\mathbb{E}[\|f_\bullet\|_\infty \mid \mathfrak{F}_\infty]] = \mathbb{E}[\|f_\bullet\|_\infty] < \infty$. Again, by Markov's inequality, for any $a > 0$, we have that:

$$\mathbb{P}\Big[\lim_{t\to\infty} \|\mu_t\|_\infty \geq a\mathbb{E}[\|f_\bullet\|_\infty]\Big] \leq \frac{\mathbb{E}\left[\lim_{t\to\infty} \|\mu_t\|_\infty\right]}{a\mathbb{E}[\|f_\bullet\|_\infty]} \leq \frac{\mathbb{E}\left[\lim_{t\to\infty} \mathbb{E}[\|f_\bullet\|_\infty \mid \mathfrak{F}_t]\right]}{a\mathbb{E}[\|f_\bullet\|_\infty]} = \frac{1}{a}\,. \tag{66}$$

Therefore, for any $a > 0$ and any given $\mathbf{x} \in \mathcal{X}$, with probability at least $1 - \frac{1}{a}$, the following holds:

$$\begin{aligned}
\lim_{t\to\infty} \mathbb{P}[\mathbf{x}_t = \mathbf{x} \mid \mathfrak{F}_{t-1}] &\geq p(\mathbf{x}) \lim_{t\to\infty} \ell_{t-1}(\mathbf{x}) \\
&\geq p(\mathbf{x}) \lim_{t\to\infty} \Psi\left(-\frac{\|\mu_{t-1}\|_\infty + \tau_*}{\sigma_\epsilon}\right) \\
&\geq p(\mathbf{x})\Psi\left(-\frac{a\mathbb{E}[\|f_\bullet\|_\infty] + \tau_*}{\sigma_\epsilon}\right) \\
&=: b_\infty(a) > 0\,.
\end{aligned} \tag{67}$$

Hence, for any $\varepsilon_a \in (0, b_\infty(a))$, there is $N_a \in \mathbb{N}$, such that $\mathbb{P}[\mathbf{x}_t = \mathbf{x} \mid \mathfrak{F}_{t-1}] \geq b_\infty(a) - \varepsilon_a > 0$, for all $t \geq N_a$. As a result, $\sum_{t'=1}^{t} \mathbb{P}[\mathbf{x}_{t'} = \mathbf{x} \mid \mathfrak{F}_{t'-1}] \geq (b_\infty(a) - \varepsilon_a)(t - N_a)$, for all $t \geq N_a$, which asymptotically diverges at a rate proportional to t. By Lemma D.5 and the definition of the big-$\mathcal{O}$ notation, for any $\mathbf{x} \in \mathcal{X}$, we then have that:

$$\forall a > 0, \quad \mathbb{P}\left[\limsup_{t\to\infty} \left|t\sigma_t^2(\mathbf{x})\right| \leq \sigma_\epsilon^2 < \infty\right] \geq 1 - \frac{1}{a}\,. \tag{68}$$

Taking the limit as $a \to \infty$, we can finally conclude that:

$$\mathbb{P}\left[\limsup_{t\to\infty} \left|t\sigma_t^2(\mathbf{x})\right| < \infty\right] = 1, \tag{69}$$

i.e., σ_t^2 is almost surely $\mathcal{O}(t^{-1})$, which concludes the proof. □

Remark D.1. *The convergence rate in Theorem D.1 is optimal and cannot be further improved. As shown by previous works in the online learning literature (Mutný & Krause, 2018; Takeno et al., 2024), a lower bound on the GP variance at each iteration* $t \geq 1$ *is given by* $\sigma_t^2(\mathbf{x}) \geq \sigma_\epsilon^2(\sigma_\epsilon^2 + t)^{-1}$ *(assuming* $k(\mathbf{x}, \mathbf{x}) = 1$*), which is the case when every observation in the dataset was collected at the same point* $\mathbf{x} \in \mathcal{X}$ *(see Takeno et al., 2024, Lem. 4.2). Therefore, the lower and upper bounds on the asymptotic convergence rates for the GP variance differ by only up to a multiplicative constant.*

The result in Theorem D.1 now allows us to derive a convergence rate for VSD's approximations to the level-set distributions. To do so, however, we will require the following mild assumption, which is satisfied by any prior distribution which has support on the entire domain $\mathcal{X}$.

Assumption D.5. *The prior distribution is such that* $p(\mathbf{x}) > 0$*, for all* $\mathbf{x} \in \mathcal{X}$.

Theorem 2.1. *Let assumptions D.1 to D.5 hold. Then VSD equipped with GP-PI approaches the level-set distribution at the following rate:*

$$\mathbb{D}[p(\mathbf{x}|y > \tau_t, \mathcal{D}_t) \| p(\mathbf{x}|y > \tau_t, f_\bullet)] \in \mathcal{O}_\mathbb{P}(t^{-1/2})\,.$$

Proof. We first prove an upper bound for the KL divergence in terms of the PI approximation error. We then derive a bound for this term and apply Theorem D.1 to obtain a convergence rate.

KL bound formulation. Let $\ell_t(\mathbf{x}) := p(y > \tau_t|\mathbf{x}, \mathcal{D}_t)$ and $\ell^*_t(\mathbf{x}) := p(y > \tau_t|\mathbf{x}, f_\bullet)$, for $\mathbf{x} \in \mathcal{X}$. From the definition of the KL divergence, we have that:

$$\begin{aligned}\mathbb{D}[p(\mathbf{x}|y > \tau_t, \mathcal{D}_t)\|p(\mathbf{x}|y > \tau_t, f_\bullet)] &= \mathbb{E}_{p(\mathbf{x}|y>\tau_t,\mathcal{D}_t)}[\log p(\mathbf{x}|y > \tau_t, \mathcal{D}_t) - \log p(\mathbf{x}|y > \tau_t, f_\bullet)]\\ &= \mathbb{E}_{p(\mathbf{x}|y>\tau_t,\mathcal{D}_t)}[\log \ell_t(\mathbf{x}) - \log \ell^*_t(\mathbf{x})]\\ &\quad + \log \mathbb{E}_{p(\mathbf{x})}[\ell^*_t(\mathbf{x})] - \log \mathbb{E}_{p(\mathbf{x})}[\ell_t(\mathbf{x})]\\ &= \mathbb{E}_{p(\mathbf{x}|y>\tau_t,\mathcal{D}_t)}\left[\log\left(\frac{\ell_t(\mathbf{x})}{\ell^*_t(\mathbf{x})}\right)\right] + \log\left(\frac{\mathbb{E}_{p(\mathbf{x})}[\ell^*_t(\mathbf{x})]}{\mathbb{E}_{p(\mathbf{x})}[\ell_t(\mathbf{x})]}\right).\end{aligned} \tag{70}$$

For logarithms, we know that $\log(1 + a) \leq a$, for all $a > -1$, which shows that:

$$\log\left(\frac{\ell_t(\mathbf{x})}{\ell^*_t(\mathbf{x})}\right) = \log\left(1 + \frac{\ell_t(\mathbf{x}) - \ell^*_t(\mathbf{x})}{\ell^*_t(\mathbf{x})}\right) \leq \frac{\ell_t(\mathbf{x}) - \ell^*_t(\mathbf{x})}{\ell^*_t(\mathbf{x})} \tag{71}$$

$$\log\left(\frac{\mathbb{E}_{p(\mathbf{x})}[\ell^*_t(\mathbf{x})]}{\mathbb{E}_{p(\mathbf{x})}[\ell_t(\mathbf{x})]}\right) = \log\left(1 + \frac{\mathbb{E}_{p(\mathbf{x})}[\ell^*_t(\mathbf{x}) - \ell_t(\mathbf{x})]}{\mathbb{E}_{p(\mathbf{x})}[\ell_t(\mathbf{x})]}\right) \leq \frac{\mathbb{E}_{p(\mathbf{x})}[\ell^*_t(\mathbf{x}) - \ell_t(\mathbf{x})]}{\mathbb{E}_{p(\mathbf{x})}[\ell_t(\mathbf{x})]}. \tag{72}$$

Combining the above into Equation 70 yields:

$$\mathbb{D}[p(\mathbf{x}|y > \tau_t, \mathcal{D}_t)\|p(\mathbf{x}|y > \tau_t, f_\bullet)] \leq \mathbb{E}_{p(\mathbf{x}|y>\tau_t,\mathcal{D}_t)}\left[\frac{\ell_t(\mathbf{x}) - \ell^*_t(\mathbf{x})}{\ell^*_t(\mathbf{x})}\right] + \frac{\mathbb{E}_{p(\mathbf{x})}[\ell^*_t(\mathbf{x}) - \ell_t(\mathbf{x})]}{\mathbb{E}_{p(\mathbf{x})}[\ell_t(\mathbf{x})]}. \tag{73}$$

The denominator in the expression above is such that:

$$\forall t \geq 0, \quad \ell^*_t(\mathbf{x}) = p(y > \tau_t|\mathbf{x}, f_\bullet) = \Psi\left(\frac{f_\bullet(\mathbf{x}) - \tau_t}{\sigma_\epsilon}\right) \geq \Psi\left(-\frac{\|f_\bullet\|_\infty + \tau_*}{\sigma_\epsilon}\right), \quad \forall \mathbf{x} \in \mathcal{X}. \tag{74}$$

By Lemma D.4, we know that $\mathbb{E}[\|f_\bullet\|_\infty] < \infty$, which implies that $\mathbb{P}[\|f_\bullet\|_\infty < \infty] = 1$ by Markov's inequality. Next, we derive a bound for the approximation error term.

Error bound. We now derive an upper bound for the difference $\Delta\ell_t(\mathbf{x}) := \ell_t(\mathbf{x}) - \ell^*_t(\mathbf{x})$ and then show that it asymptotically vanishes. Applying Taylor's theorem to Ψ, we can bound $\Delta\ell_t$ as a function of the approximation error between the mean μ_t and the true function $f_\bullet$ as:

$$\begin{aligned}\forall t \geq 0, \quad |\Delta\ell_t(\mathbf{x})| &= \left|\Psi\left(\frac{\mu_t(\mathbf{x}) - \tau_t}{\sqrt{\sigma_t^2(\mathbf{x}) + \sigma_\epsilon^2}}\right) - \Psi\left(\frac{f_\bullet(\mathbf{x}) - \tau_t}{\sigma_\epsilon}\right)\right|\\ &\leq \frac{1}{\sqrt{2\pi}}\left|\frac{\mu_t(\mathbf{x}) - \tau_t}{\sqrt{\sigma_t^2(\mathbf{x}) + \sigma_\epsilon^2}} - \frac{f_\bullet(\mathbf{x}) - \tau_t}{\sigma_\epsilon}\right|\\ &= \frac{1}{\sqrt{2\pi}}\left|\frac{\sigma_\epsilon\mu_t(\mathbf{x}) - f_\bullet(\mathbf{x})\sqrt{\sigma_t^2(\mathbf{x}) + \sigma_\epsilon^2} + \tau_t(\sqrt{\sigma_t^2(\mathbf{x}) + \sigma_\epsilon^2} - \sigma_\epsilon)}{\sigma_\epsilon\sqrt{\sigma_t^2(\mathbf{x}) + \sigma_\epsilon^2}}\right|\\ &\leq \frac{|\sigma_\epsilon\mu_t(\mathbf{x}) - f_\bullet(\mathbf{x})\sqrt{\sigma_t^2(\mathbf{x}) + \sigma_\epsilon^2}| + |\tau_t|\sigma_t(\mathbf{x})}{\sigma_\epsilon^2\sqrt{2\pi}}\\ &\leq \frac{\sigma_\epsilon|\mu_t(\mathbf{x}) - f_\bullet(\mathbf{x})| + \sigma_t(\mathbf{x})(|f_\bullet(\mathbf{x})| + |\tau_t|)}{\sigma_\epsilon^2\sqrt{2\pi}}, \quad \forall \mathbf{x} \in \mathcal{X},\end{aligned} \tag{75}$$

since $\sup_{\epsilon\in\mathbb{R}}\left|\frac{\mathrm{d}\Psi(\epsilon)}{\mathrm{d}\epsilon}\right| = \frac{1}{\sqrt{2\pi}} < 1$, and we used the fact that $\sigma_\epsilon \leq \sqrt{\sigma_t^2(\mathbf{x}) + \sigma_\epsilon^2} \leq \sigma_t(\mathbf{x}) + \sigma_\epsilon$ to obtain the last two inequalities.

Convergence rate. To derive a convergence rate, given any $\mathbf{x} \in \mathcal{X}$ and $t \geq 0$, we have that:

$$\mathbb{E}[|\Delta\ell_t(\mathbf{x})| \mid \mathfrak{F}_t] \leq \frac{\sigma_\epsilon\mathbb{E}[|\mu_t(\mathbf{x}) - f_\bullet(\mathbf{x})| \mid \mathfrak{F}_t] + \sigma_t(\mathbf{x})(\mathbb{E}[|f_\bullet(\mathbf{x})| \mid \mathfrak{F}_t] + |\tau_t|)}{\sigma_\epsilon^2\sqrt{2\pi}}. \tag{76}$$

We know that $\mathbb{E}[|f_\bullet(\mathbf{x})| \mid \mathfrak{F}_t]$ is almost surely bounded, and by Jensen's inequality, it also holds that:

$$\mathbb{E}[|\mu_t(\mathbf{x}) - f_\bullet(\mathbf{x})| \mid \mathfrak{F}_t] \leq \sigma_t(\mathbf{x}). \tag{77}$$

Applying Theorem D.1, we then have that:

$$|\Delta \ell_t(\mathbf{x})| \in \mathcal{O}_{\mathbb{P}}(t^{-1/2}). \tag{78}$$

Since $\|\mu_t\|_\infty \leq \mathbb{E}[\|f_\bullet\|_\infty | \mathfrak{F}_t] \in \mathcal{O}_{\mathbb{P}}(1)$, we also have that:

$$\frac{1}{\mathbb{E}_{p(\mathbf{x})}[\ell_t(\mathbf{x})]} \in \mathcal{O}_{\mathbb{P}}(1)\,. \tag{79}$$

Lastly, we know that $\frac{1}{\ell_t^*(\mathbf{x})} \in \mathcal{O}_{\mathbb{P}}(1)$ by Equation 74 and the observation that $\|f_\bullet\|_\infty \in \mathcal{O}_{\mathbb{P}}(1)$. The main result then follows by combining the rates above into Equation 73. □

D.5 PERFORMANCE ANALYSIS

At every iteration $t \geq 1$, VSD samples $\mathbf{x}_t$ from (an approximation to) the target $p(\mathbf{x}|y > \tau_{t-1}, \mathcal{D}_{t-1})$ and obtains an observation $y_t \sim p(y|\mathbf{x}_t)$. A positive hit consists of an event $y_t > \tau_{t-1}$, where τ_{t-1} is computed based on the data available in $\mathcal{D}_{t-1}$ or a constant. Therefore, we can compute the probability of a positive hit for a given realization of $f_\bullet$ as:

$$\mathbb{P}[y_t > \tau_{t-1} \mid \mathcal{D}_{t-1}, f_\bullet] = \mathbb{E}_{p(\mathbf{x}|y>\tau_{t-1},\mathcal{D}_{t-1})}[p(y > \tau_{t-1}|\mathbf{x}, f_\bullet)]\,. \tag{80}$$

Then the expected number of hits H_T after $T \geq 1$ iterations is given by:

$$\mathbb{E}[H_T \mid f_\bullet] = \sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{x}|y>\tau_{t-1},\mathcal{D}_{t-1})}[p(y > \tau_{t-1}|\mathbf{x}, f_\bullet)]\,. \tag{81}$$

We will compare this quantity with the expected number of hits H_T^* obtained by a sampling distribution with full knowledge of the objective function $f_\bullet$:

$$\mathbb{E}[H_T^* \mid f_\bullet] = \sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{x}|y>\tau_{t-1},f_\bullet)}[p(y_t > \tau_{t-1}|\mathbf{x}, f_\bullet)]\,. \tag{82}$$

The next result allows us to bound the difference between these two quantities.

Corollary 2.1. *Under the settings in Theorem 2.1, we also have that:*

$$\mathbb{E}[|H_T - H_T^*|] \in \mathcal{O}(\sqrt{T})\,.$$

Proof. For all $T \geq 1$, we have that:

$$\begin{aligned}
\mathbb{E}[H_T - H_T^*] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}_{p(\mathbf{x}|y>\tau_{t-1},\mathcal{D}_{t-1})}[p(y > \tau_{t-1}|\mathbf{x}, f_\bullet)] - \mathbb{E}_{p(\mathbf{x}|y>\tau_{t-1},f_\bullet)}[p(y_t > \tau_{t-1}|\mathbf{x}, f_\bullet)]\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \sum_{\mathbf{x}\in\mathcal{X}} p(y > \tau_{t-1}|\mathbf{x}, f_\bullet)\left(p(\mathbf{x}|y > \tau_{t-1}, \mathcal{D}_{t-1}) - p(\mathbf{x}|y > \tau_{t-1}, f_\bullet)\right)\right] \\
&= \mathbb{E}\left[\sum_{t=0}^{T-1} \sum_{\mathbf{x}\in\mathcal{X}} p(y > \tau_t|\mathbf{x}, f_\bullet)p(\mathbf{x})\left(\frac{\ell_t(\mathbf{x})}{\mathbb{E}_{p(\mathbf{x}')}[\ell_t(\mathbf{x}')]} - \frac{\ell_t^*(\mathbf{x})}{\mathbb{E}_{p(\mathbf{x}')}[\ell_t^*(\mathbf{x}')]}\right)\right] \\
&\leq \mathbb{E}\left[\sum_{t=0}^{T-1} \sum_{\mathbf{x}\in\mathcal{X}} p(\mathbf{x})\left(\frac{|\Delta\ell_t(\mathbf{x})|}{\min\{\mathbb{E}_{p(\mathbf{x}')}[\ell_t(\mathbf{x}')]\,, \mathbb{E}_{p(\mathbf{x}')}[\ell_t^*(\mathbf{x}')]\}}\right)\right],
\end{aligned} \tag{83}$$

since $p(y > \tau_{t-1}|\mathbf{x}, f_\bullet) \leq 1$, for all $t \geq 1$. As both $\|\mu_t\|_\infty$ and $\|f_\bullet\|_\infty$ are in $\mathcal{O}_{\mathbb{P}}(1)$, $\min\{\mathbb{E}_{p(\mathbf{x}')}[\ell_t(\mathbf{x}')]\,, \mathbb{E}_{p(\mathbf{x}')}[\ell_t^*(\mathbf{x}')]\}$ is lower bounded by some constant. As $\Delta\ell_t(\mathbf{x}) \in \mathcal{O}_{\mathbb{P}}(t^{-1/2})$, for T large enough and some $C > 0$, we then have that:

$$\mathbb{E}[|H_T - H_T^*|] \leq C \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 2C\sqrt{T} \in \mathcal{O}(\sqrt{T}), \tag{84}$$

which follows by an application of the Euler-Maclaurin formula, since $\int_1^T \frac{1}{\sqrt{t}}\,\mathrm{d}t = 2\sqrt{T} - 2$ and the remainder term asymptotically vanishes. □

Remark D.2. *If the oracle achieves $\mathbb{E}[H_T^*] = T$, the error bound in Corollary 2.1 suggests an increasing rate of positive hits by VSD as $\frac{1}{T}\mathbb{E}[H_T] \geq 1 - CT^{-1/2}$, for some constant $C > 0$ and large enough T. Therefore, VSD should asymptotically achieve a full rate of 1 positive hit per iteration in the single-point batch setting we consider. Note, however, that the results above do not discount for repeated samples, though should still indicate that VSD achieves a high discovery rate over the course of its execution.*

E VSD WITH NEURAL NETWORK CPEs

In this section, we consider VSD with class probability estimators that are not based on GP regression, which was the case for the previous section, while specifically focusing on neural network models. We will, however, show that with a kernel-based formulation we are able to capture the classification models based on neural networks which we use. This is possible by analyzing the behavior of infinite-width neural networks (Jacot et al., 2018; Lee et al., 2019), whose approximation error with respect to the finite-width model can be bounded (Liu et al., 2020; Eldan et al., 2021).

Although our classifiers are learned by minimizing the cross-entropy (CE) loss, we can connect their approximations with theoretical results from the infinite-width neural network (NN) literature, which are mostly based on the mean squared error (MSE) loss. Recall that, given a dataset $\mathcal{D}_N^z := \{(\mathbf{x}_n, z_n)\}_{n=1}^N$ with binary labels $z_n \in \{0, 1\}$, the cross-entropy loss for a probabilistic classifier $\pi_\theta : \mathcal{X} \to [0, 1]$ parameterized by θ is given by[3]:

$$\mathcal{L}_{\text{CPE}}(\theta, \mathcal{D}_N^z) := -\frac{1}{N}\sum_{n=1}^N z_n \log \pi_\theta(\mathbf{x}_n) + (1 - z_n)\log(1 - \pi_\theta(\mathbf{x}_n)) \,. \tag{85}$$

The MSE loss for the same model corresponds to:

$$\mathcal{L}_{\text{MSE}}(\theta, \mathcal{D}_N^z) := \frac{1}{N}\sum_{n=1}^N (z_n - \pi_\theta(\mathbf{x}_n))^2 \,. \tag{86}$$

The following result establishes a connection between the two loss functions.

Proposition E.1. *Given a binary classification dataset $\mathcal{D}_N^z$ of size $N \geq 1$, the following holds for the cross-entropy and the mean-square error losses:*

$$\mathcal{L}_{CPE}(\theta, \mathcal{D}_N^z) \geq \mathcal{L}_{\text{MSE}}(\theta, \mathcal{D}_N^z), \quad \forall N \in \mathbb{N} \,. \tag{87}$$

Proof. Applying the basic logarithmic inequality $\log(1 + a) \leq a$, for all $a > -1$, to the cross-entropy loss definition yields:

$$\begin{aligned}
\mathcal{L}_{\text{CPE}}(\theta, \mathcal{D}_N^z) &:= -\frac{1}{N}\sum_{n=1}^N z_n \log \pi_\theta(\mathbf{x}_n) + (1 - z_n)\log(1 - \pi_\theta(\mathbf{x}_n)) \\
&\geq -\frac{1}{N}\sum_{n=1}^N z_n(\pi_\theta(\mathbf{x}_n) - 1) - (1 - z_n)\pi_\theta(\mathbf{x}_n) \\
&= -\frac{1}{N}\sum_{n=1}^N 2z_n\pi_\theta(\mathbf{x}_n) - z_n - \pi_\theta(\mathbf{x}_n) \\
&= \frac{1}{N}\sum_{n=1}^N z_n - 2z_n\pi_\theta(\mathbf{x}_n) + \pi_\theta(\mathbf{x}_n) \,.
\end{aligned} \tag{88}$$

Now note that $z_n = z_n^2$, for $z_n \in \{0, 1\}$, and $\pi_\theta(\mathbf{x}_n) \geq \pi_\theta(\mathbf{x}_n)^2$, as $\pi_\theta(\mathbf{x}_n) \in [0, 1]$, for all $n \in \{1, \ldots, N\}$. Making these substitutions in Equation 88, we obtain:

$$\mathcal{L}_{\text{CPE}}(\theta, \mathcal{D}_N^z) \geq \frac{1}{N}\sum_{n=1}^N z_n^2 - 2z_n\pi_\theta(\mathbf{x}_n) + \pi_\theta(\mathbf{x}_n)^2 = \mathcal{L}_{\text{MSE}}(\theta, \mathcal{D}_N^z) \,, \tag{89}$$

which concludes the proof. □

---

[3] We implicitly assume that $0 < \pi_\theta(\mathbf{x}_n) < 1$, for $n \in \{1, \ldots, N\}$, so that the CE loss is well defined. This assumption can, however, be relaxed when dealing with the MSE loss, which remains well defined otherwise.

The result in Proposition E.1 suggests that minimizing the cross-entropy loss will lead us to minimize the MSE loss as well, since the latter is upper bounded by the former. This result provides us with theoretical justification to derive convergence results based on the MSE loss, which has been better analyzed in the NN literature (Jacot et al., 2018; Lee et al., 2019), as a proxy to establish convergence guarantees for the CE-based VSD setting.

E.1 LINEAR APPROXIMATIONS VIA THE NEURAL TANGENT KERNEL

Let π^* denote the unknown true classifier, i.e., $\pi(\mathbf{x}) := p(y > \tau | \mathbf{x}, f_\bullet)$, for $\mathbf{x} \in \mathcal{X}$. In the following, we will assume that π^* is an unknown, fixed element of a reproducing kernel Hilbert space (RKHS) associated with a given kernel (Schölkopf & Smola, 2001). In the case of infinite-width neural networks, we know that under certain assumptions the NN trained via gradient descent under the MSE loss will asymptotically converge to a kernel ridge regression solution whose kernel is given by the neural tangent kernel (NTK, Jacot et al. (2018)). This asymptotic solution is equivalent to the posterior mean of a Gaussian process that assumes no observation noise. For a finite amount of training steps with a non-infinitesimal learning rate, however, the literature has shown that gradient-based training provides a form of implicit regularization. In that case, we recover a regularized kernel ridge regression solution which can be robust to label noise (Hu et al., 2020) for us to use. Lastly, although our analysis will be based on NTK results, the approximation error between the infinite-width and the finite-width NN vanishes with the square root of the network width for most popular NN architectures (Liu et al., 2020). Therefore, we can assume that these approximation guarantees will remain useful for wide-enough, finite-width NN models.

Implicit regularization. Several results in the literature have shown that training overparameterized neural networks via gradient descent provides a form of implicit regularization on the learned model (Fleming, 1990; Yao et al., 2007; Soudry et al., 2018; Barrett & Dherin, 2021), with some of the same behavior extending to the stochastic gradient setting (Smith et al., 2021). In particular, Fleming (1990) showed a direct equivalence between an early stopped gradient-descent linear model and the solution of a regularized least-squares problem with a penalty on the parameters vector Euclidean norm. Therefore, as wide deep neural networks behave as linear models in the infinite-width limit when trained via gradient descent (Jacot et al., 2018; Lee et al., 2019), it is reasonable to model our least-squares problem with the MSE loss via its regularized version:

$$\hat{\pi}_N \in \operatorname*{argmin}_{\pi \in \mathcal{F}_k} \sum_{i=1}^{N} (\pi(\mathbf{x}_i) - z_i)^2 + \rho \|\pi\|_k^2, \tag{90}$$

where $\rho > 0$ is a possibly unknown regularization factor, $\mathcal{F}_k$ denotes the RKHS associated with the neural tangent kernel k for the given CPE NN architecture, and $\|\cdot\|_k$ denotes the RKHS norm. The problem above is equivalent to regularized kernel ridge regression (Shawe-Taylor & Cristianini, 2004), whose solution is given by:

$$\hat{\pi}_N(\mathbf{x}) = \pi_0(\mathbf{x}) + \mathbf{k}_N(\mathbf{x})^\top (\mathbf{K}_N + \rho \mathbf{I})^{-1} (\mathbf{z}_N - \pi_0(\mathcal{X}_N)), \tag{91}$$

where $\mathbf{k}_N$ and $\mathbf{K}_N$ are defined in the same way as for the GP case, π_0 denotes the untrained NN classifier at initialization, and $\pi_0(\mathcal{X}_N) := [\pi_0(\mathbf{x}_i)]_{i=1}^N$. For our analysis, we may assume that $\pi_0 = 0$ at times, noting that the least-squares problem can always be solved for the residuals $z - \pi_0(\mathbf{x})$ and then have π_0 added back to the solution. We refer the reader to Jacot et al. (2018) for further discussion on the effect of the network initialization.

Approximation for finite-width networks. For fully connected, convolutional or residual networks equipped with smooth activation functions (e.g., sigmoid or $\tanh$), Liu et al. (2020) showed that the approximation error between the linear model and the finite-width NN is $\widetilde{\mathcal{O}}(m^{-1/2})$, where m denotes the minimum layer width, and the $\widetilde{\mathcal{O}}$ notation corresponds to the $\mathcal{O}$-notation with logarithmic factors suppressed. NTK results for other activation functions, e.g., ReLU (Chen & Xu, 2021), and different neural network architectures, such as multi-head attention (Hron et al., 2020), are also available in the literature.

E.2 ASSUMPTIONS

In the following, we present a series of mild technical assumptions needed for our theoretical analysis of NN-based CPEs. Firstly, we assume a bounded NTK.

Assumption E.1. *The NTK k corresponding to the network architecture in π_θ is bounded:*

$$\sup_{\mathbf{x}\in\mathcal{X}} k(\mathbf{x},\mathbf{x}) < \infty\,, \tag{92}$$

for some constant $b_k > 0$.

We will also assume that the threshold is fixed to simplify the analysis. However, our results should still be applicable to the time-varying threshold setting after minor adjustments.

Assumption E.2. *The threshold is fixed, i.e., $\tau_t = \tau \in \mathbb{R}$, for all $t \geq 1$.*

The following assumption on label noise should always hold for Bernoulli random variables (Boucheron et al., 2013). Any upper bound on the sub-Gaussian parameter should suffice for the analysis (e.g., $\sigma_z \leq 1$ for Bernoulli variables).

Assumption E.3. *For all $t \in \mathbb{N}$ and all $\mathbf{x} \in \mathcal{X}$, label noise $\zeta = \mathbb{1}[y > \tau] - \pi^*(\mathbf{x})$, with $y \sim p(y|\mathbf{x}, f_\bullet)$, is σ_z-sub-Gaussian:*

$$\forall a \in \mathbb{R}, \quad \mathbb{E}\left[\exp\left(a\zeta\right)\right] \leq \exp\left(\frac{a^2\sigma_z^2}{2}\right), \tag{93}$$

for some $\sigma_z \geq 0$.

For this analysis, we mainly assume that the true classifier $\pi^*(\mathbf{x}) = p(y > \tau|\mathbf{x}, f_\bullet)$ is a fixed, though unknown, element of the RKHS given by the NTK, which is formalized by the following assumption.

Assumption E.4. *There is $\pi^* \in \mathcal{F}_k$ such that:*

$$\pi^*(\mathbf{x}) = p(y > \tau|\mathbf{x}, f_\bullet), \quad \forall \mathbf{x} \in \mathcal{X}. \tag{94}$$

For a rich enough RKHS, such assumption is mild, especially given that most popular NN architectures possess universal approximation guarantees (Hornik et al., 1989). Finally, the next assumption ensures enough sampling asymptotically over the domain $\mathcal{X}$, which we still assume is finite.

Assumption E.5. *For any $t \geq 1$, the variational family is such that sampling probabilities are bounded away from 0, i.e.:*

$$\exists b > 0: \quad \forall t \in \mathbb{N}, \quad q(\mathbf{x}|\phi_t) \geq b\,. \tag{95}$$

The assumption above only imposes mild constraints on the generative models $q(\mathbf{x}|\phi)$, so that probabilities for all candidates $\mathbf{x} \in \mathcal{X}$ are never exactly 0, though still allowed to be arbitrarily small.

E.3 APPROXIMATION ERROR FOR NN-BASED CPES

Similar to the GP-PI setting, we will assume a batch size of 1, so that we can simply use the iteration index $t \geq 0$ for our estimators. We recall that convergence rates for the batch setting should only be affected by a batch-size-dependent multiplicative factor, preserving big-$\mathcal{O}$ convergence rates. We start by defining the following *proxy* variance:

$$t \geq 1, \quad \hat{\sigma}_t^2(\mathbf{x}) = k(\mathbf{x},\mathbf{x}) - \mathbf{k}_t(\mathbf{x})^\top(\mathbf{K}_t + \rho\mathbf{I})^{-1}\mathbf{k}_t(\mathbf{x})\,, \quad \mathbf{x} \in \mathcal{X}, \tag{96}$$

which is equivalent to a GP posterior variance when observation noise is assumed to be zero-mean Gaussian with variance given by ρ. Given its similarities, we have that if enough sampling is asymptotically guaranteed, we can apply the same convergence results available for the GP-PI-based CPE, i.e., $\hat{\sigma}_t^2 \in \mathcal{O}(t^{-1/2})$ almost surely. We then invoke Assumption E.5 to derive the following result.

Lemma E.1. *Let Assumption E.1 and Assumption E.5 hold. Then the following almost surely holds for the proxy variance:*

$$\hat{\sigma}_t^2 \in \mathcal{O}(t^{-1})\,. \tag{97}$$

Proof. The proof follows by verifying that the sum of sampling probabilities at any point $\mathbf{x} \in \mathcal{X}$ diverges as $t \to \infty$ by Assumption E.5, and then by Lemma D.5 the result follows. $\square$

Lemma E.2. *Let assumptions E.1 to E.4 hold. Then, given any $\delta \in (0, 1]$, the following holds with probability at least $1 - \delta$ for the approximation error between $\hat{\pi}_t$ and π^*:*

$$\forall t \geq 1, \quad |\hat{\pi}_t(\mathbf{x}) - \pi^*(\mathbf{x})| \leq \beta_t(\delta)\hat{\sigma}_t(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X}, \tag{98}$$

where $\beta_t(\delta) := \|\pi^*\|_k + \sigma_\zeta\sqrt{2\rho^{-1}\log(\det(\mathbf{I} + \rho^{-1}\mathbf{K}_t)^{1/2}/\delta)}$.

Proof. The result above is a direct application of Theorem 3.5 in Maillard (2016) which provides an upper confidence bound on the kernelized least-squares regressor approximation error (another version of the same result is also available in Durand et al. (2018, Thr. 1)). $\square$

For the next result, we need to define the following quantity:

$$\xi_T := \max_{\mathcal{X}_T \subset \mathcal{X} : |\mathcal{X}_T| \leq T} \frac{1}{2} \log\det(\mathbf{I} + \rho^{-1}\mathbf{K}(\mathcal{X}_T))\,, \tag{99}$$

where $\mathbf{K}(\mathcal{X}_T) := [k(\mathbf{x}, \mathbf{x}')]_{\mathbf{x}, \mathbf{x}' \in \mathcal{X}_T} \in \mathbb{R}^{|\mathcal{X}_T| \times |\mathcal{X}_T|}$. Note that ξ_T corresponds to the maximum information gain of a GP model (Srinivas et al., 2010) with covariance function given by the NTK, assuming Gaussian observation noise with variance given by ρ. Then ξ_T is mainly dependent on the eigenvalue decay of the kernel under its spectral decomposition (Vakili et al., 2021). For the spectrum of the NTK, a few results are available in the literature (Murray et al., 2023).

Proposition E.2. *Let assumptions E.1 to E.5 hold. Then, given $\delta \in (0, 1]$, the following holds with probability at least $1 - \delta$ for VSD equipped with a wide enough NN-based CPE model $\hat{\pi}_t$:*

$$\mathbb{D}[p(\mathbf{x}|y > \tau_t, \mathcal{D}_t) \| p(\mathbf{x}|y > \tau_t, f_\bullet)] \in \mathcal{O}\left(\sqrt{\frac{\xi_t}{t}}\right)\,. \tag{100}$$

Proof. The result follows by applying the same steps as in the proof of Theorem 2.1. We note that $\ell_t^*(\mathbf{x}) = \pi^*(\mathbf{x}) > 0$, due to observation noise, so that $\ell_t^*(\mathbf{x})^{-1} \in \mathcal{O}_\mathbb{P}(1)$. Similarly, Lemma E.2 implies that $|\hat{\pi}_t(\mathbf{x}) - \pi^*(\mathbf{x})| \leq \beta_t(\delta)\sigma_t(\mathbf{x})$ with probability at least $1 - \delta$ simultaneously over all $\mathbf{x} \in \mathcal{X}$, so that ratio-dependent terms in Theorem 2.1 should remain bounded in probability. The upper bound in the result then follows by noticing that in our case $|\Delta\ell_t(\mathbf{x})| \leq \beta_t(\delta)\hat{\sigma}_t(\mathbf{x})$ with high probability, where $\hat{\sigma}_t \in \mathcal{O}(t^{-1/2})$ by Lemma E.1, and $\beta_t(\delta) \in \mathcal{O}(\sqrt{\xi_t})$ by Lemma E.2 and the definition of ξ_t in Equation 99. $\square$

The result above tells us that VSD is able to recover a similar asymptotic convergence guarantee to the one we derived for the GP-PI case, depending on the choice of NN architecture and more specifically on the spectrum of its associated NTK. In the case of a fully connected multi-layer ReLU network, for example, Chen & Xu (2021) showed an equivalence between the RKHS of the ReLU NTK and that of the Laplace kernel $k(\mathbf{x}, \mathbf{x}') = \exp(-C\|\mathbf{x} - \mathbf{x}'\|)$. As the latter is equivalent to a Matérn kernel with smoothness parameter set to 0.5 (Rasmussen & Williams, 2006), the corresponding information gain bound is $\xi_t \in \widetilde{\mathcal{O}}(t^{\frac{d}{1+d}})$, where d here denotes the dimensionality of the domain $\mathcal{X}$ (Vakili & Olkhovskaya, 2023). In the case of discrete sequences of length M, the dimensionality of $\mathcal{X}$ is determined by M. Therefore, in this case, we have proven Corollary 2.2.[4]

Corollary 2.2. *Let π_θ be modeled via a fully connected ReLU network. Then, under the assumptions in Proposition E.2, VSD achieves:*

$$\mathbb{D}[p(\mathbf{x}|y > \tau_t, \mathcal{D}_t) \| p(\mathbf{x}|y > \tau, f_\bullet)] \in \widetilde{\mathcal{O}}_\mathbb{P}\left(t^{-\frac{1}{2(M+1)}}\right)\,, \tag{12}$$

which asymptotically vanishes for all finite sequence lengths M.

Similar steps can be applied to derive convergence guarantees for VSD with other neural network architectures based on the eigenspectrum of their NTK (Murray et al., 2023) and following the recipe in, e.g., Vakili et al. (2021) or Srinivas et al. (2010).

---

[4] Here $\widetilde{\mathcal{O}}_\mathbb{P}$ suppresses logarithmic factors, as in $\widetilde{\mathcal{O}}$.

F VSD AS A BLACK-BOX OPTIMIZATION LOWER BOUND

A natural question to ask is how VSD relates to the BO objective for probability of improvement (Garnett, 2023, Ch.7),

$$\mathbf{x}_t^* = \underset{\mathbf{x}}{\operatorname{argmax}} \log \alpha_{PI}(\mathbf{x}, \mathcal{D}_N, \tau) \,. \tag{101}$$

Firstly, we can see that the expected log-likelihood of term of Equation 7 lower-bounds this quantity.

Proposition F.1. *For a parametric model, $q(\mathbf{x}|\phi)$, given $\phi \in \Phi \subseteq \mathbb{R}^m$ and $q \in \mathcal{P} : \mathcal{X} \times \Phi \to [0, 1]$,*

$$\max_{\mathbf{x}} \log \alpha_{PI}(\mathbf{x}, \mathcal{D}_N, \tau) \geq \max_{\phi} \mathbb{E}_{q(\mathbf{x}|\phi)} [\log \alpha_{PI}(\mathbf{x}, \mathcal{D}_N, \tau)] \,, \tag{102}$$

and the bound becomes tight as $q(\mathbf{x}|\phi_t^*) \to \delta(\mathbf{x}_t^*)$, a Dirac delta function at the maximizer $\mathbf{x}_t^*$.

Taking the argmax of the RHS will result in the variational distribution collapsing to a delta distribution at $\mathbf{x}_t^*$ for an appropriate choice of $q(\mathbf{x}|\phi)$. The intuition for Equation 102 is that the expected value of a random variable is always less than or equal to its maximum. The proof of this is in Daulton et al. (2022); Staines & Barber (2013). Extending this lower bound, we can show the following.

Proposition F.2. *For a divergence $\mathbb{D} : \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \to [0, \infty)$, and a prior $p_0 \in \mathcal{P}(\mathcal{X})$,*

$$\max_{\mathbf{x}} \log \alpha_{PI}(\mathbf{x}, \mathcal{D}_N, \tau) \geq \max_{\phi} \mathbb{E}_{q(\mathbf{x}|\phi)} [\log \alpha_{PI}(\mathbf{x}, \mathcal{D}_N, \tau)] - \mathbb{D}[q(\mathbf{x}|\phi) \| p_0(\mathbf{x})] \,. \tag{103}$$

We can see that this bound is trivially true given the range of divergences, and this covers VSD as a special case. However, this bound is tight if and only if p_0 concentrates as a Dirac delta at $\mathbf{x}_t^*$ with an appropriate choice of $q(\mathbf{x}|\phi)$. In any case, the lower bound remains valid for any choice of informative prior p_0 or even a uninformed prior, which allows us to maintain the framework flexible to incorporate existing prior information whenever that is available.