COOL-BO: CONSTRAINT-ALIGNED LATENT BAYESIAN OPTIMIZATION FOR HIGH-DIMENSIONAL PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent methods employing variational autoencoders (VAEs) to map discrete, high-dimensional input spaces to continuous latent representations, have achieved remarkable success in high-dimensional Bayesian optimization (BO) in unconstrained settings. However, real-world problems (e.g., molecule design, robotic control tasks) are often subject to practical constraints (e.g., drug likeliness of proposed molecules, safety constraints) that are not explicitly captured by standard VAE training. Consequently, the trained VAE may map feasible and infeasible inputs to overlapping latent regions. As a result, nearby latent points can decode to candidates with different feasibility, producing an irregular feasible region that complicates acquisition-function optimization and reduces sampling efficiency. In this paper, we propose CooL-BO, which explicitly shapes the learned latent manifold with respect to input feasibility. By introducing a discriminator and a dynamic mechanism for learning priors, we enforce separability in the latent domain during pre-training of the VAE, thereby facilitating the selection of promising feasible candidates for latent space BO. Consequently, CooL-BO achieves a higher BO objective than state-of-the-art latent space BO methods, while reducing computation time by up to 10 times across various high-dimensional benchmark tasks. Our results highlight the importance of leveraging both feasible and infeasible inputs in representation learning for latent-space BO, motivating the need for learning representations tailored for latent BO tasks.

1 INTRODUCTION

Many real-world problems, such as molecular design and robotic manipulation, require optimizing expensive black-box functions over large, structured, and often discrete spaces [42]. Bayesian optimization (BO) is a principled and sample-efficient approach for solving such problems [35]. However, its direct application to high-dimensional settings is challenging as the computational cost of the optimization process becomes prohibitive. State-of-the-art approaches address this challenge by learning a low-dimensional continuous latent representation of the input domain using generative models such as variational autoencoders (VAEs), and then applying BO in this latent space [49, 9].

Yet, in many practical settings, the optimization problem is constrained. For example, the synthesizability of molecular structures and safety requirements for trajectories visited in robotics are constraints that need to be satisfied while optimizing the objective. In some scenarios, these constraints are also expensive and unknown, similar to the objective function [31]. However, in many cases, feasibility can be verified cheaply, specified a priori, or estimated from prior data. For example, in molecular design, feasibility may be determined by drug-likeness rules or synthetic-accessibility scores [29, 12], or in robotic control tasks, the safety constraints may be specified from domain knowledge or learned using prior expert demonstrations [22].

Given known constraints, a natural approach to perform latent space BO is to first learn a latent domain restricting the VAE only to the feasible inputs, and then perform unconstrained BO in the latent space [47]. This can be achieved by filtering the dataset or using constrained sampling methods such as rejection sampling. However, training the VAE solely on feasible data may make the representation more prone to collapse or poor coverage, and does not guarantee the feasibility of decoded samples due to the inherent noise of VAE decoders [27, 7].

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Alternatively, constraints can be handled during BO through penalty-based objectives [30] or learned latent surrogate models for constraints based on the encoded inputs [51]. While these methods can guide candidate selection, they typically leave the VAE training objective unchanged. As a result, feasibility is learned or enforced only after the latent space has been constructed, rather than shaping the geometry of the latent representation itself. Consequently, feasible and infeasible candidates may become entangled in latent space, potentially resulting in a fragmented or irregular feasible region, which in turn increases the computational cost of maximizing the acquisition function.

In this paper, we propose Constraint-aligned Latent BO (CooL-BO), which enables efficient Bayesian optimization with known constraints in high-dimensional spaces by learning a latent representation that encodes feasibility during VAE pre-training. In particular, it introduces a discriminator that induces separability between feasible and infeasible candidates in the latent space, together with a dynamic prior-learning mechanism that adapts the latent prior to the resulting geometry. Our contributions can be summarized as follows:

- We propose CooL-BO, a framework that incorporates known input-space constraints into latent space Bayesian optimization by explicitly encoding feasibility structure in the learned latent representation.
- We introduce a discriminative VAE pretraining objective that leverages feasibility information to induce separation between feasible and infeasible regions in the latent space. In particular, we show that a linear discriminator suffices to produce a structured latent geometry while enabling computationally efficient candidate selection for latent BO.
- We show that shaping the latent space with feasibility information leads to more efficient candidate selection, enabling BO to identify high-quality feasible solutions with negligible computational overhead.
- We evaluate CooL-BO on various high-dimensional benchmark tasks; our results show that CooL-BO improves sampling efficiency compared to state-of-the art latent-space BO methods, achieving higher best-observed objective values under a fixed evaluation budget and reaching given performance levels with fewer black-box evaluations.

2 Related Work

The problem of learning separable and disentangled latent spaces is closely related to latent clustering and representation learning via contrastive learning. Deep clustering [41, 53, 36, 21] aims to cluster the input data in a low-dimensional latent space in an unsupervised way, which differs from our setup where the labels (corresponding to the feasibility of the inputs) are known. Recent methods have applied contrastive learning to VAEs to perform latent clustering when the input data comprises multiple classes [38, 1, 52]. Methods to learn dynamic priors for VAEs have also been proposed [45, 3]. These methods introduce additional learnable components, such as a deep neural network to learn a multi-modal prior for the VAE. However, such methods lack an explicit decision boundary between the clusters in the latent domain. Discriminative approaches to representation learning on supervised and semi-supervised data have been explored in the literature. For example, [32] use a discriminator to circumvent the KL loss in imposing a multi-modal Gaussian prior. However, a key distinction is that the priors of the respective classes are preselected. On the contrary, our approach dynamically learns the centroids of the priors of the feasible and infeasible classes.

Latent space BO [49, 9, 47, 28] has been predominantly used to optimize high-dimensional functions without constraints. In the presence of unknown constraints, safe Bayesian optimization methods [43, 50, 51, 16] use additional constraint Gaussian processes (GPs) to model feasibility in the latent domain. These methods may be adapted to a BO problem with known constraints by pretraining the GPs prior to the BO task.

Few works have considered BO with known constraints. Penalty-based approaches [30, 44, 39] attempt to incorporate known constraints in the BO objective by imposing a penalty for constraint violations in the input domain. Surrogates have been used to model constraint feasibility in the latent domain in [17]. A limitation of these methods is that feasibility is modeled during the BO task, impacting the sampling efficiency. Recent methods have focused on reducing the reconstruction gap of the VAE for improving BO performance [17, 27, 7]. These methods, however, do not consider a constrained problem setting like ours. Another line of research developed methods for constrained

Bayesian optimization [19, 33, 15, 20] applied to control tasks. These methods do not attempt to subdue the infeasibility of the proposed candidates during BO, thus restricting their applicability to tasks with a limited evaluation budget. Methods that incorporate the constraint in the expected improvement (EI) acquisition function have also been proposed [40, 13, 54]. However, all of the aforementioned methods are restricted to low-dimensional problems and have not been applied to latent space BO.

3 PRELIMINARIES

We consider a constrained Bayesian optimization problem with known constraints,

$$\text{(P1):} \qquad \mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{1}$$

$$\text{s.t:} \quad g_i(\mathbf{x}) \geq 0, \quad i = 1, \dots I, \tag{2}$$

where $\mathcal{X} \subset \mathbb{R}^d$ is the domain of function $f : \mathcal{X} \to \mathbb{R}$. f is a black-box, unknown function that is expensive to evaluate, while the constraints $g_i : \mathcal{X} \to \mathbb{R}, \forall i \in \{1, \dots, I\}$ are known, e.g., g_i may be a safety constraint imposed by a pretrained neural network [22]. We consider the case $d \gg 1$, which makes it computationally inefficient to solve P1 in the input domain $\mathcal{X}$.

3.1 BAYESIAN OPTIMIZATION

In BO, an initial dataset $\mathcal{D} = \{(\mathbf{x_n}, y_n)\}_{n=1}^N$ is used to fit a surrogate model, typically a Gaussian process. An acquisition function computed using the surrogate is then optimized to select the next candidate $\hat{\mathbf{x}}$ to be evaluated [42, 35, 37]. The selected point $\hat{\mathbf{x}}$ is then evaluated using the oracle (the function f) and the point together with the observation is appended to $\mathcal{D}$ to update the surrogate in an iterative manner. When constraints are unknown or expensive to evaluate, they are often modeled with separate surrogate models and handled via constrained acquisition functions (e.g., based on the probability of feasibility) [15]. Alternatively, penalty-based formulations can be used when constraints are known explicitly [30, 44, 39], as discussed in Section 2.

3.2 LATENT SPACE BAYESIAN OPTIMIZATION

For high dimensional unconstrained BO, the structured high-dimensional input space is first transformed to a lower dimensional continuous domain $\mathcal{Z} \subset \mathbb{R}^l$, with $l \leq d$, and then the surrogate $\tilde{f} \sim \mathcal{GP}(m(\mathbf{z}), \kappa(\mathbf{z}, \mathbf{z}'))$ is optimized in the latent domain [34, 26, 49, 9, 6, 47].

A deep autoencoder, comprising an encoder $\Phi : \mathcal{X} \to \mathcal{Z}$ and a decoder $\Gamma : \mathcal{Z} \to \mathcal{X}$, is typically trained in an unsupervised manner to learn a continuous representation $\mathcal{Z}$ of the input space. In practice, variational autoencoders (VAE) are frequently employed to induce smooth latent spaces, which are well suited for Bayesian optimization. The parameters $\theta_\Phi, \theta_\Gamma$ of the VAE are trained by maximizing the evidence lower bound (ELBO)

$$\mathcal{L}_{\text{VAE}}(\theta_\Phi, \theta_\Gamma; \mathbf{X}) = \mathbb{E}_{\Phi(\mathbf{Z}|\mathbf{X})}[\Gamma(\mathbf{X}|\mathbf{Z})] - \text{KL}(\Phi(\mathbf{Z}|\mathbf{X}) \;||\; p(\mathbf{Z})), \tag{3}$$

where $\Phi(\mathbf{Z}|\mathbf{X})$ and $\Gamma(\mathbf{X}|\mathbf{Z})$ are the variational posterior distributions of the encoder and decoder, respectively, and $p(\mathbf{Z})$ is the prior distribution. The distributions $\Phi(\mathbf{Z}|\mathbf{X})$, $\Gamma(\mathbf{X}|\mathbf{Z})$, and $p(\mathbf{Z})$ are often assumed to be Gaussian for analytical tractability. Importantly, the parameters of the Gaussian prior $p(\mathbf{Z})$ in equation 3 are typically chosen arbitrarily prior to the training. A common choice is to let $p(\mathbf{Z})$ follow a unit Gaussian ($\mathcal{N}(\mathbf{0}, \mathbf{I})$). We discuss the implications of the choice of the prior in solving a constrained Bayesian optimization problem in Section 4.

3.3 CONSTRAINED BAYESIAN OPTIMIZATION IN LATENT SPACE

One approach to addressing the high-dimensional constrained BO problem P1 via latent space BO would be to train the VAE on the feasible dataset only. One could use constrained sampling techniques such as rejection sampling or Markov chain Monte Carlo (MCMC) sampling, thus transforming the constrained optimization problem (P1) into an unconstrained one in latent space. Such an approach, however, reduces, the diversity of the data presented to the VAE, making the VAE more susceptible to representation collapse. Furthermore, the output of the decoder of a VAE is a random variable and may be infeasible, hence the constraints (2) would have to be imposed in the target domain $\mathcal{X}$.

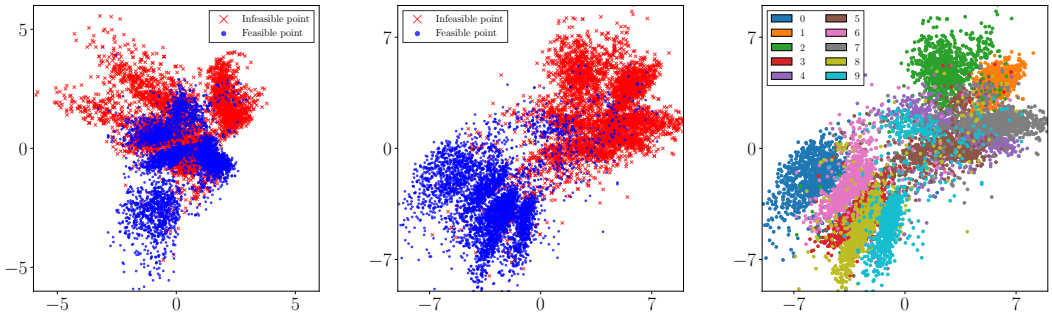


Figure 1: Latent vectors learned by a standard variational autoencoder and latent vectors learned by CooL-BO on MNIST. The feasible set includes the digits $0, 3, 6, 8, 9$. The VAE uses a Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ in (a) and dynamic priors using a linear discriminator to learn separable latent vectors (CooL-BO) in (b). (c) Shows the latent vectors learned by CooL-BO, colored by the respective classes.

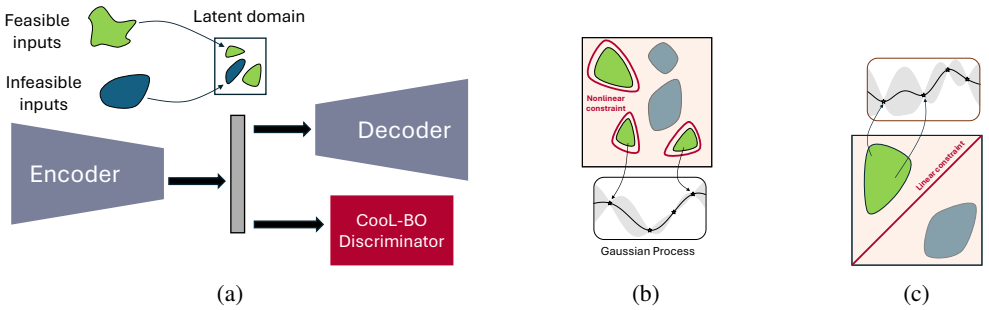


Figure 2: Overview of CooL-BO. (a) VAE and the discriminator are jointly trained to learn separable latent representations. Illustration of a latent space learned by CooL-BO with a nonlinear discriminator (b) and a separable latent space learned by CooL-BO with a linear discriminator (c).

The alternative is to model feasibility in latent space, as done in high dimensional safe BO, where an additional constraint GP $\tilde{g}_i \sim \mathcal{GP}(m(\mathbf{z}), \kappa(\mathbf{z}, \mathbf{z}'))$ is learned for each of the constraint functions in the latent domain. The feasibility in the latent domain is then determined by the GP constraints $\tilde{g}_i(\mathbf{z}) \geq 0, \forall\, i = 1, \dots I$, and is used for constraining the acquisition function. Given the known constraints in (2), such latent space constraints $g_i(\mathbf{z})$ can be learned, and used as in safe BO. Nonetheless, following this approach, the training of the VAE and thus the learned latent domain are oblivious to the feasibility of the latent vectors, i.e., the VAE training objective does not attempt to incorporate feasibility when learning the latent representations. Consequently, the learned latent domain may not be separable with respect to feasibility, as shown in Figure 1a, which makes it computationally expensive to sample feasible points for the maximization of the acquisition function, as we show in Section 5.

No. infeasible regions	**Best BO objective**
5	-2.2×10^{-6}
50	-1.4×10^{-3}
75	-5.6×10^{-2}
100	-6.1×10^{-2}

Table 1: BO optimization results on a 2D toy example $\max_{\mathbf{x}} -||\mathbf{x} - \frac{1}{2}||_2^2$ as a function of no. infeasible regions (circles, radius 0.02), under a computation budget of 100 oracle evaluations (details in Appendix B.1).

4 CooL-BO

CooL-BO addresses the limitations of existing methods in handling known constraints within latent space BO. The intuition behind CooL-BO is that the geometry of the infeasible regions in the latent domain would affect the BO exploration overhead, in turn impacting the attained BO objective under

a limited budget (Table 1). Therefore, we explicitly shape the latent space by encoding feasibility information into the learned representations, thus inducing a geometry that separates feasible and infeasible regions. In particular, we promote linear separability, which enables efficient optimization of the acquisition function over the latent domain (Figure 1b). We then formulate an analogous constrained optimization problem in the latent domain and perform BO directly in this domain (Figure 2a).

4.1 Learning Separable Latent Representations

While learning disentangled representations has been studied in the domain of contrastive learning [45, 38], such approaches do not provide an explicit model of the decision boundary between feasible and infeasible regions. The distinction is crucial in our setting, where the boundary defines a notion of feasibility (or safety) in the latent space. To address this, we introduce a discriminator that guides the VAE encoder to learn separable latent representations.

Let $\mathbf{T} : \mathcal{X} \to \{0, 1\}$ be a r.v. denoting the feasibility of the input $\mathbf{x}$. $\mathbf{T}(\mathbf{x}) = 0$ implies that the input $\mathbf{x}$ is infeasible with respect to any of the constraints g_i, and $\mathbf{T}(\mathbf{x}) = 1$ means that the input $\mathbf{x}$ satisfies all the constraints. Our objective is to derive an evidence lower bound (ELBO) to maximize the joint probability $p(\mathbf{X}, \mathbf{T})$. Assuming conditional independence between $\mathbf{X}$ and $\mathbf{T}$ given the latent vector $\mathbf{Z}$, we can write

$$\begin{aligned}\log p(\mathbf{X}, \mathbf{T}) &= \log \int p(\mathbf{X}, \mathbf{T}, \mathbf{Z}) d\mathbf{Z} = \log \int \frac{\Gamma(\mathbf{X}|\mathbf{Z})\Omega(\mathbf{T}|\mathbf{Z})}{\Phi(\mathbf{Z}|\mathbf{X})} \Phi(\mathbf{Z}|\mathbf{X}) p(\mathbf{Z}) d\mathbf{Z} \\ &= \log \mathbb{E}_{\Phi(\mathbf{Z}|\mathbf{X})} \left[\frac{\Gamma(\mathbf{X}|\mathbf{Z})\Omega(\mathbf{T}|\mathbf{Z})}{\Phi(\mathbf{Z}|\mathbf{X})} p(\mathbf{Z}) \right] \\ &\geq \mathbb{E}_{\Phi(\mathbf{Z}|\mathbf{X})} \left[\log \Gamma(\mathbf{X}|\mathbf{Z}) + \log \Omega(\mathbf{T}|\mathbf{Z}) \right] - \mathrm{KL}(\Phi(\mathbf{Z}|\mathbf{X}) \;||\; p(\mathbf{Z})), \end{aligned} \tag{4}$$

where we have introduced the discriminator $\Omega(\mathbf{T}|\mathbf{Z})$ to approximate the probability $p(\mathbf{T}|\mathbf{Z})$, and applied Jensen's inequality in the last step. Observe that the choice of the prior distribution $p(\mathbf{Z})$ impacts the discriminator objective in (4). Choosing the same prior (such as $\mathcal{N}(\mathbf{0}, \mathbf{I})$) for both feasible and infeasible latent vectors would lead to close clustering of the latent vectors, thus making it hard for the discriminator to distinguish between them.

Instead, we propose to introduce class-specific priors p_0 and p_1 for the feasible and infeasible inputs, and learn these priors dynamically. Since $\mathbf{T}$ is a binary random variable, we can rewrite (4) as

$$\begin{aligned}\log p(\mathbf{X}, \mathbf{T}) \geq \mathbb{E}_{\Phi(\mathbf{Z}|\mathbf{X})}[\log \Gamma(\mathbf{X}|\mathbf{Z})] - (1 - \mathbf{T})\, \mathrm{KL}(\Phi(\mathbf{Z}|\mathbf{X}) \;||\; p_0(\mathbf{Z})) \\ - \mathbf{T}\, \mathrm{KL}(\Phi(\mathbf{Z}|\mathbf{X}) \;||\; p_1(\mathbf{Z})) + \mathbb{E}_{\Phi(\mathbf{Z}|\mathbf{X})} \left[\mathbf{T} \log \Omega(\mathbf{T}|\mathbf{Z}) \right], \end{aligned} \tag{5}$$

where we let $p_t(\mathbf{Z}) = \mathcal{N}(\mu_t, \mathbf{I})$, $t \in \{0, 1\}$, and compute the centroids μ_t after each training iteration. By grouping the terms in (5), we can express the log-likelihood as

$$\mathcal{L}(\theta_\Phi, \theta_\Gamma, \theta_\Omega; \mathbf{X}, \mathbf{T}) = \mathcal{L}_{\text{VAE}}(\theta_\Phi, \theta_\Gamma; \mathbf{X}) + \mathbb{E}_{\Phi(\mathbf{Z}|\mathbf{X})} \left[\mathcal{L}_{\text{dis}}(\theta_\Omega; \mathbf{T}, \mathbf{Z}) \right], \tag{6}$$

where $\mathcal{L}_{\text{dis}}$ is the log-likelihood of the discriminator. Note that the training objective for both the VAE and the discriminator is to maximize the ELBO in (6), as opposed to an adversarial setting, thus improving training stability. Note that it is possible to introduce the term $\mathbb{E}_{\Phi(\mathbf{Z}|\mathbf{X})} \left[\mathcal{L}_{\text{GP}}(\theta_{\mathcal{GP}}, \theta_\Phi; \mathbf{Z}, \mathbf{y}) \right]$ in (6), making it amenable to joint training of the GP as in [34].

Relation to prior work. Our proposed method to learn a separable latent space has key differences from facet clustering in VAEs. For instance, the prior in [45] is decomposed as $p(\mathbf{Z}|\mathbf{T})$ and parameterized using a deep neural network. Such an approach relies on the inherent dissimilarities between the inputs from different classes to enforce separability. Conditional VAEs [23, 8] condition the learned representations on known attributes during pretraining, but they do not enforce separability with respect to the known attributes (constraints) in the latent domain. On the contrary, our decomposition in (5), uses the discriminator $\Omega(\mathbf{T}|\mathbf{Z})$ to explicitly force the encoder to learn separable representations for feasible and infeasible inputs. We provide a comparison of VAE architectures in Appendix A.

4.2 CONSTRAINED LATENT SPACE BAYESIAN OPTIMIZATION WITH COOL-BO

Having shaped the latent space during VAE pretraining, we reformulate P1 as a constrained Bayesian optimization problem in the latent domain,

$$\text{(P2):} \qquad \mathbf{z}^* = \arg\max_{\mathbf{z}} f(\Gamma(\mathbf{z})) \tag{7}$$

$$\text{s.t:} \quad \Omega(\mathbf{z}) \geq \delta, \tag{8}$$

where we use $\Omega(\mathbf{z})$ as a short-hand for $\Omega(\mathbf{t}|\mathbf{z})$, and $\delta \in [0, 1]$ serves as a feasibility threshold. Note that (8) implicitly imposes feasibility with respect to all the constraints $g_i(\mathbf{x}),\ \forall i = 1, \ldots, I$ by the design of the discriminator (see Subsection 4.1) to propose latent candidates deemed feasible with confidence δ.

4.3 COOL-BO ALGORITHM

The above formulation is sufficient to incorporate feasibility into the latent space. However, to ensure computational efficiency in downstream BO, additional structure is required. In particular, the geometry induced by the discriminator should enable efficient optimization of the acquisition function.

While the discriminator can, in principle, be parameterized by a nonlinear model (e.g., quadratic, or a deep neural network) [45, 46], such a choice may induce highly nonconvex and entangled feasible regions in the latent space (Figure 2b), thereby complicating the optimization of the acquisition function (c.f. Appendix B.1). To address this, we restrict the discriminator to a class of linear models with a sigmoid output activation, which induces a linearly separable (and hence geometrically structured) latent space by leveraging the capacity of the deep encoder (Figure 2c). This structure yields a simple characterization of the feasible region and facilitates efficient optimization in the latent domain. By letting $\Delta = \log\left(\frac{\delta}{1-\delta}\right)$, we can equivalently express the feasibility constraint as

$$\theta_\Omega^\top \mathbf{z} \geq \Delta. \tag{9}$$

Note that feasibility in the latent domain is now characterized by the linear constraint (9), allowing seamless integration with trust-region Bayesian optimization methods that impose box constraints.

To further improve separability, observe that the choice of the KL priors in (5) is critical to the effectiveness of the discriminator, as it directly influences the separability of the latent representations. In particular, overlapping priors $p_t(\mathbf{Z}),\ t \in \{0, 1\}$ lead to poor separation between feasible and infeasible regions. While both the discriminator and the VAE are trained to minimize the loss (6), jointly optimizing their parameters can lead to unstable updates of the KL priors, as the underlying latent distributions shift during training. To address this, we adopt an alternating optimization scheme, where the parameters of the VAE and the discriminator are updated in turns, and the centroids of the KL priors are recomputed between iterations, as summarized in Algorithm 1.

Our choice of the estimators of the priors (line 7-8) is motivated by the observation that the aggregated posterior, $p(\mathbf{z}) = \frac{1}{n}\sum_{n=1}^{N} \mathbb{E}\left[\Phi(\mathbf{z}_n|\mathbf{x}_n)\right]$, maximizes the ELBO [46]. We therefore estimate the centroids μ_0 and μ_1 by using the aggregate posteriors of the feasible and infeasible samples, respectively. This avoids introducing additional learnable parameters for modeling the priors and simplifies training.

Owing to the linear structure of the feasibility constraint in the latent space, we can directly leverage trust-region Bayesian optimization methods. In particular, we adopt TuRBO [11], which operates by defining a hyper-rectangular trust region $\mathcal{T}_{\hat{\mathbf{z}}}$ around the current incumbent $\hat{\mathbf{z}}$. The size of the trust region is adaptively adjusted, shrinking upon consecutive failures and expanding upon improvements. Within each trust region, we impose the linear constraint (9) during candidate generation and use Thompson sampling to propose candidates in the latent space. This structured latent representation enables efficient constrained optimization within each trust region, allowing CooL-BO to fully exploit TuRBO.

Algorithm 1 CooL-BO VAE pretraining algorithm

Input: Dataset comprising input vectors and their feasibility $\mathcal{D} = \{\mathbf{x}_n, \mathbf{t}_n\}_{n=1}^N$. The number of updates to the discriminator k per training epoch. We used $k = 1$ in all our experiments.

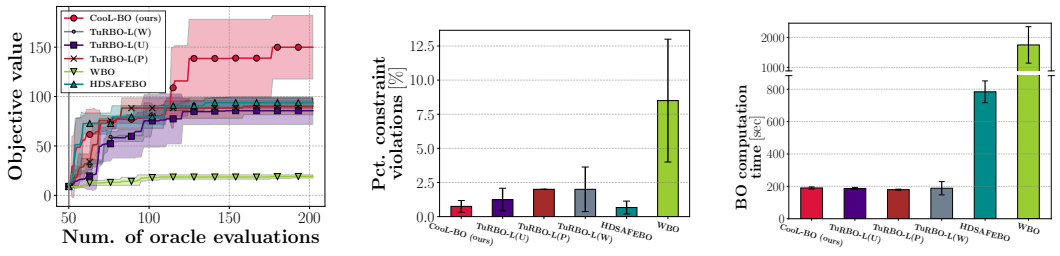
Initialize: Prior centroids $\mu_0, \mu_1 \leftarrow \mathbf{0}$
for number of training iterations **do**
  for k steps **do** ▷ **Discriminator update**
    Sample of M minibatches $\{(\mathbf{X}^{(m)}, \mathbf{T}^{(m)})\}_{m=1}^M$ from the dataset $\mathcal{D}$
    Update the discriminator by descending its stochastic gradient:
      $\nabla_{\theta_\Omega} \mathcal{L}_{\text{BCE}}(\theta_\Omega; \mathbf{T}^{(m)}, \mathbf{Z}^{(m)})$
  end for
  $n_0 \leftarrow \sum_{n=1}^N (1 - \mathbf{t}_n)$, $n_1 \leftarrow \sum_{n=1}^N \mathbf{t}_n$ ▷ **Recompute prior centroids**
  $\mu_0 \leftarrow \frac{1}{n_0} \sum_{n=1}^N (1 - \mathbf{t}_n) \cdot \mathbb{E}\left[\Phi(\mathbf{z}_n | \mathbf{x}_n)\right]$, $\mu_1 \leftarrow \frac{1}{n_1} \sum_{n=1}^N \mathbf{t}_n \cdot \mathbb{E}\left[\Phi(\mathbf{z}_n | \mathbf{x}_n)\right]$
  Sample of M minibatches $\{(\mathbf{X}^{(m)}, \mathbf{T}^{(m)})\}_{m=1}^M$ from the dataset $\mathcal{D}$ ▷ **VAE update**
  Update the VAE by descending its stochastic gradient, imposing KL using μ_0, μ_1
    $\nabla_{\theta_\Phi, \theta_\Gamma} \mathcal{L}_{\text{VAE}}(\theta_\Phi, \theta_\Gamma; \mathbf{X}^{(m)}) + \mathcal{L}_{\text{BCE}}(\mathbf{T}^{(m)}, \mathbf{Z}^{(m)})$
end for

5 Experiments

We evaluate the performance of CooL-BO on eleven high-dimensional constrained BO problems, comprising Guacamol molecule synthesis, arithmetic expression fitting, and synthetic benchmark tasks. For all tasks, we impose novel constraints, something that prior works on latent space BO lack in their evaluation. In this section, we benchmark the performance of CooL-BO with respect to the baselines. In Appendices E.2 and H we ablate the hyperparameters for CooL-BO. We implement CooL-BO using BoTorch [4] and GPyTorch [14].

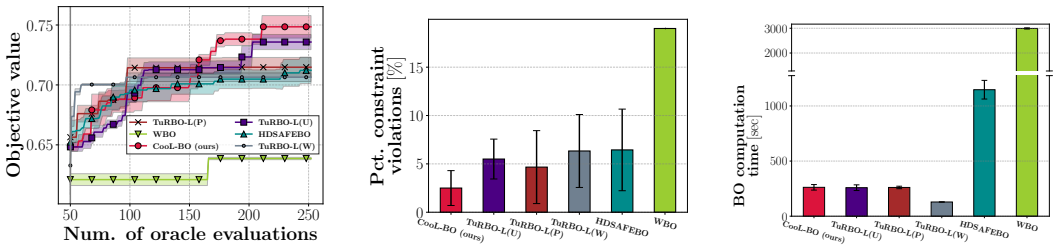
Baselines. As we are not aware of prior works in latent space BO that incorporate known constraints, we adapt existing methods in solving a constrained problem such as ours. The first baseline is `TuRBO-L(U)`, which leverages `TuRBO` in latent space to solve an unconstrained BO problem, as detailed in Section 3.3. The second baseline is `TuRBO-L(P)` (as described in [44, Section. 5.3]) that transforms the constrained optimization problem into an unconstrained one by imposing a penalty for constraint. The third baseline is `HDSAFEBO` [51], which solves a constrained optimization problem in the latent space with GP constraints. The fourth baseline is `TuRBO-L(W)`, which performs weighted retraining to periodically update the VAE during BO [48]. The fifth baseline `WBO` is a seminal work that weighs the expected improvement objective by the constraint feasibility [17]. In case of `HDSAFEBO` and `WBO`, we pretrain the constraint GPs on the training datasets of the respective tasks.

Tasks. We consider $\log$ P and Ranolazine MPO (`Rano`) tasks from the Guacamol dataset [5]. For both tasks, we impose a constraint that the quantitative estimation of drug-likeness (QED) of the proposed molecules during BO can have a value of 0.56, at most. The expressions task [48, 18, 2], `Expr` involves generating a univariate math expression to match the target expression $x^2 + \sin(x) + \sin(3x)$, subject to the constraint that the proposed math expressions are syntactically valid. We also evaluate the performance of CooL-BO on six challenging benchmark synthetic BO tasks [10].



(a) (b) (c)

Figure 3: Optimization results on Guacamol $\log$ P task.



(a) (b) (c)

Figure 4: Optimization results on Guacamol `Rano` task.

Since the input domain of these tasks is orthogonal, we adapt these tasks to our problem of constrained latent space BO by first projecting the problem to a higher-dimensional space, and let the dimensionality of the latent domain of the VAE be less than that of the original problem domain to induce imperfect reconstruction. Importantly, we evaluate the black-box objective and the constraints in the high-dimensional space (details in Appendix B.2). Details on the datasets used for pretraining the VAEs are provided in Appendix C. For all tasks, we benchmark the performance of latent space BO methods under a limited evaluation budget. For all our experiments, we set the feasibility threshold for CooL-BO $\delta = 0.5$. We evaluate the impact of the feasibility threshold in Appendix H.

We observed that for Guacamol Zaleplon MPO and Perindopril MPO tasks, with the added complexity of the QED constraint, none of the approaches managed to improve upon the initial best with an evaluation budget of $3,000$. These tasks typically require a much larger budget and joint training of the VAE and surrogate model [34, 26], which is not the scope of our proposed work. Thus, we do not report the performance on these tasks.

Evaluation metrics. We measure performance in terms of i) the achieved BO objective value as a function of the number of black-box function (oracle) evaluations, ii) the feasibility of the latent candidates proposed during latent space BO, and iii) the computation time for the Bayesian optimization loop (excluding the time to update the objective and constraint surrogate models). All results are an average of ten runs, and the bars indicate the standard deviation.

5.1 Results on Guacamol Benchmarks

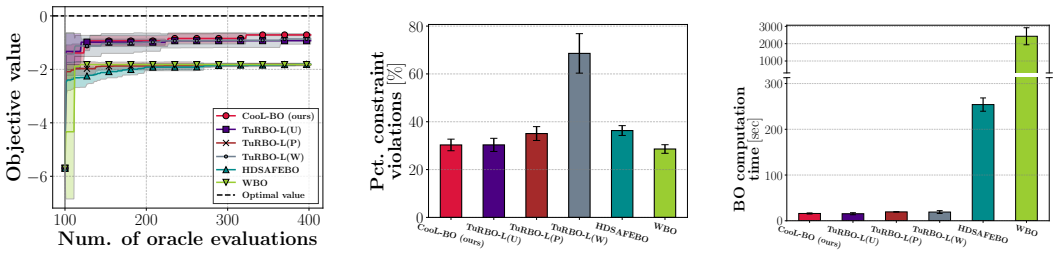
The results on Guacamol tasks are shown in Figures 3 and 4. In Figure 3a, we observe that CooL-BO achieves a mean objective value of 150.01, which is considerably higher than the score of 97.12 achieved by `HDSAFEBO`. In case of the `Rano` task (Figure 4a), CooL-BO still outperforms `TuRBO-L(U)`, while `HDSAFEBO` achieves a lower objective, possibly due to the higher fraction of infeasible candidates proposed during latent BO (Figure 4b). In both tasks, CooL-BO proposes a lower fraction of infeasible candidates during BO (Figures 3b and 4b), owing to its feasibility-aligned latent manifold. From Figures 3c and 4c, we observe that CooL-BO is able to solve a constrained latent space optimization problem at the expense of solving an unconstrained problem (`TuRBO-L(U)`), while solving a constrained latent space BO problem with GP constraints (`HDSAFEBO`) results in about 5 times higher BO computation time than using CooL-BO. Note that the BO computation time of the baseline `WBO` is significantly higher as it optimizes a constraint-weighted EI acquisition function, which becomes computationally expensive in higher dimensions.

5.2 Results on Expressions Task

We plot the results for the `Expr` task in Figure 5. CooL-BO attains the highest objective value of -0.8439 compared to an objective -0.9129 attained by the baseline `TuRBO-L(W)`. Furthermore, CooL-BO achieves up to $10\times$ lower BO computation time than `HDSAFEBO` (Figure 5c).

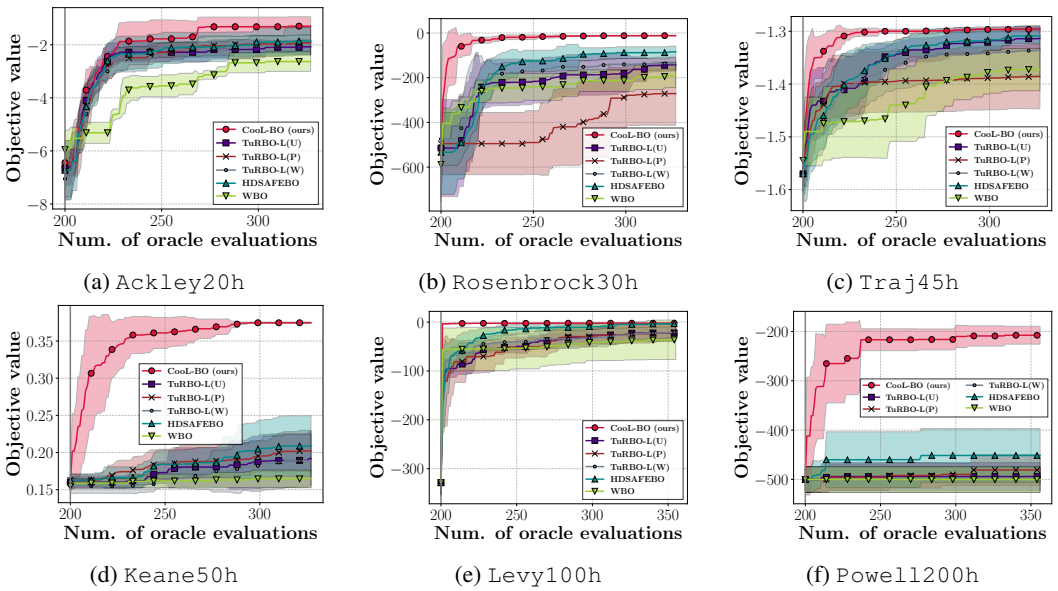
5.3 Results on Synthetic benchmark tasks

We plot the BO objective performance for the synthetic benchmarks in Figure 6. Across all tasks, CooL-BO attains the highest final BO objective and achieves a comparable BO objective with respect



(a) (b) (c)

Figure 5: Optimization results on `Expr` task.



(a) `Ackley20h` (b) `Rosenbrock30h` (c) `Traj45h`

(d) `Keane50h` (e) `Levy100h` (f) `Powell200h`

Figure 6: Optimization results for synthetic benchmark tasks.

to the baselines with up to $1.5\times$ fewer oracle evaluations (Figures 6b and 6c). Additional BO performance metrics for the synthetic tasks are reported in Appendix D.

5.4 Additional Results and Ablation Studies

In Appendices E, and J, we compare the performance of CooL-BO against SOTA VAE architectures. In Appendix G, we provide insights into the latent candidates proposed by CooL-BO. In Appendix I, we benchmark the BO performance for a batch size of 1. Additionally, we introduce two high-dimensional BO tasks on the `MNIST` dataset in Appendices B.3 and F.

6 Conclusion and Discussion

In this paper, we considered the problem of optimizing a high-dimensional black-box function with known constraints via latent space BO. We proposed a discriminative approach for pretraining the VAE to embed the feasibility of the inputs in the latent domain. By restricting ourselves to a linear parameterization of the discriminator to organize the latent geometry, our approach solves a constrained BO problem in the latent domain significantly faster compared to methods that model feasibility during the BO task. Our empirical evaluation indicates that latent representations learned leveraging both feasible and infeasible inputs can indeed impact the objective performance of BO, motivating further research in learning enriched latent representations for latent space BO.

AI USE STATEMENT

The use of generative AI tools was limited to fixing basic grammar and vocabulary. We have not used generative AI tools for the development of the core methodology or as part of the evaluation.

REPRODUCIBILITY STATEMENT

A description of the benchmark tasks and datasets used for evaluation is provided in Appendices B and C, and all hyperparameters have been reported in Appendix L. The source code will be made publicly available upon the acceptance of the paper.

REFERENCES

[1] Abubakar Abid and James Zou. "Contrastive variational autoencoder enhances salient features". In: *arXiv preprint arXiv:1902.04601* (2019).

[2] Sungsoo Ahn et al. "Guiding deep molecular optimization with genetic exploration". In: *Advances in neural information processing systems* 33 (2020), pp. 12008–12021.

[3] Junwen Bai, Shufeng Kong, and Carla P Gomes. "Gaussian mixture variational autoencoder with contrastive learning for multi-label classification". In: *international conference on machine learning*. PMLR. 2022, pp. 1383–1398.

[4] Maximilian Balandat et al. "BoTorch: A framework for efficient Monte-Carlo Bayesian optimization". In: *Advances in neural information processing systems* 33 (2020), pp. 21524–21538.

[5] Nathan Brown et al. "GuacaMol: benchmarking models for de novo molecular design". In: *Journal of chemical information and modeling* 59.3 (2019), pp. 1096–1108.

[6] Jaewon Chu et al. "Inversion-based Latent Bayesian Optimization". In: *Advances in Neural Information Processing Systems*. Ed. by A. Globerson et al. Vol. 37. Curran Associates, Inc., 2024, pp. 68258–68286. DOI: 10.52202/079017-2180. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/7e3491e922bfd199ea34ecafeb7380f0-Paper-Conference.pdf.

[7] Jaewon Chu et al. "Inversion-based latent bayesian optimization". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 68258–68286.

[8] Antonia Creswell et al. "Adversarial information factorization". In: *arXiv preprint arXiv:1711.05175* (2017).

[9] Aryan Deshwal and Jana Doppa. "Combining Latent Space and Structured Kernels for Bayesian Optimization over Combinatorial Spaces". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 8185–8200. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/44e76e99b5e194377e955b13fb12f630-Paper.pdf.

[10] David Eriksson and Matthias Poloczek. "Scalable constrained Bayesian optimization". In: *International conference on artificial intelligence and statistics*. PMLR. 2021, pp. 730–738.

[11] David Eriksson et al. "Scalable Global Optimization via Local Bayesian Optimization". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/6c990b7aca7bc7058f5e98ea909e924b-Paper.pdf.

[12] Peter Ertl and Ansgar Schuffenhauer. "Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions". In: *Journal of Cheminformatics* 1 (2009), p. 8. DOI: 10.1186/1758-2946-1-8.

[13] Paul Feliot, Julien Bect, and Emmanuel Vazquez. "A Bayesian approach to constrained single- and multi-objective optimization". In: *Journal of Global Optimization* 67.1 (2017), pp. 97–133.

[14] Jacob Gardner et al. "Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration". In: *Advances in neural information processing systems* 31 (2018).

[15] Jacob R Gardner et al. "Bayesian optimization with inequality constraints." In: *ICML*. Vol. 2014. 2014, pp. 937–945.

[16] Michael A Gelbart, Jasper Snoek, and Ryan P Adams. "Bayesian optimization with unknown constraints". In: *arXiv preprint arXiv:1403.5607* (2014).

[17] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. "Constrained Bayesian optimization for automatic chemical design using variational autoencoders". In: *Chemical Science* 11.2 (Jan. 2020), pp. 577–586. ISSN: 2041-6520. DOI: 10.1039/c9sc04026a. eprint: https://pubs.rsc.org/sc/article-pdf/11/2/577/6907411/c9sc04026a.pdf. URL: https://doi.org/10.1039/c9sc04026a.

[18] Antoine Grosnit et al. "High-dimensional Bayesian optimisation with variational autoencoders and deep metric learning". In: *arXiv preprint arXiv:2106.03609* (2021).

[19] José Miguel Hernández-Lobato et al. "A general framework for constrained Bayesian optimization using information-based search". In: *Journal of Machine Learning Research* 17.160 (2016), pp. 1–53.

[20] Riley J Hickman et al. "Bayesian optimization with known experimental and design constraints for chemistry applications". In: *Digital Discovery* 1.5 (2022), pp. 732–744.

[21] Ozsel Kilinc and Ismail Uysal. "Learning latent representations in neural networks for clustering through pseudo supervision and graph-based activity regularization". In: *arXiv preprint arXiv:1802.03063* (2018).

[22] Konwoo Kim et al. "Learning Shared Safety Constraints from Multi-task Demonstrations". In: *Advances in Neural Information Processing Systems*. 2023.

[23] Diederik P Kingma et al. "Semi-supervised learning with deep generative models". In: *Advances in neural information processing systems* 27 (2014).

[24] Jack Klys, Jake Snell, and Richard Zemel. "Learning latent subspaces in variational autoencoders". In: *Advances in neural information processing systems* 31 (2018).

[25] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. "Grammar variational autoencoder". In: *International conference on machine learning*. PMLR. 2017, pp. 1945–1954.

[26] Seunghun Lee et al. "Advancing Bayesian Optimization via Learning Correlated Latent Space". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 48906–48917. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/98e967164ae2f6811b975d686dece3eb-Paper-Conference.pdf.

[27] Seunghun Lee et al. "Latent bayesian optimization via autoregressive normalizing flows". In: *International Conference on Learning Representations*. Vol. 2025. 2025, pp. 89789–89811.

[28] Ben Letham et al. "Re-examining linear embeddings for high-dimensional Bayesian optimization". In: *Advances in neural information processing systems* 33 (2020), pp. 1546–1558.

[29] Christopher A. Lipinski et al. "Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings". In: *Advanced Drug Delivery Reviews* 23.1–3 (1997), pp. 3–25. DOI: 10.1016/S0169-409X(96)00423-1.

[30] Congwen Lu and Joel A. Paulson. "No-Regret Bayesian Optimization with Unknown Equality and Inequality Constraints using Exact Penalty Functions". In: *IFAC-PapersOnLine* 55.7 (2022). 13th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2022, pp. 895–902. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2022.07.558. URL: https://www.sciencedirect.com/science/article/pii/S2405896322009648.

[31] Hauke Maathuis, Roeland De Breuker, and Saullo GP Castro. "Autoencoder-enhanced joint dimensionality reduction for constrained Bayesian optimisation". In: *Machine Learning: Science and Technology* 6.4 (2025), p. 045028.

[32] Alireza Makhzani et al. "Adversarial autoencoders". In: *arXiv preprint arXiv:1511.05644* (2015).

[33] Alonso Marco et al. "Robot learning with crash constraints". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1439–1446.

[34] Natalie Maus et al. "Local Latent Space Bayesian Optimization over Structured Inputs". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 34505–34518. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/ded98d28f82342a39f371c013dfb3058-Paper-Conference.pdf.

[35] Jonas Mockus. "The Bayesian approach to global optimization". In: *System Modeling and Optimization: Proceedings of the 10th IFIP Conference New York City, USA, August 31–September 4, 1981*. Springer. 2005, pp. 473–481.

[36] Sudipto Mukherjee et al. "Clustergan: Latent space clustering in generative adversarial networks". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 4610–4617.

[37] Michael A Osborne, Roman Garnett, and Stephen J Roberts. "Gaussian processes for global optimization". In: (2009).

[38] Khalid Oublal et al. "DisCoV: Disentangling Time Series Representations via Contrastive based l-Variational Inference". In: *Proceedings of UniReps: the First Workshop on Unifying Representations in Neural Models*. PMLR. 2024, pp. 223–236.

[39] Victor Picheny et al. "Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/31839b036f63806cba3f47b93af8ccb5-Paper.pdf.

[40] Matthias Schonlau, William J Welch, and Donald R Jones. "Global versus local search in constrained optimization of computer models". In: *Lecture notes-monograph series* (1998), pp. 11–25.

[41] Uri Shaham et al. "Spectralnet: Spectral clustering using deep neural networks". In: *arXiv preprint arXiv:1801.01587* (2018).

[42] Bobak Shahriari et al. "Taking the Human Out of the Loop: A Review of Bayesian Optimization". In: *Proceedings of the IEEE* 104.1 (2016), pp. 148–175. DOI: 10.1109/JPROC.2015.2494218.

[43] Yanan Sui et al. "Safe Exploration for Optimization with Gaussian Processes". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 997–1005. URL: https://proceedings.mlr.press/v37/sui15.html.

[44] Alexander Thebelt et al. "Tree ensemble kernels for Bayesian optimization with known constraints over mixed-feature spaces". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 37401–37415. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/f3398b76d17792893ce6d4f660546353-Paper-Conference.pdf.

[45] Jakub Tomczak and Max Welling. "VAE with a VampPrior". In: *International conference on artificial intelligence and statistics*. PMLR. 2018, pp. 1214–1223.

[46] Jakub Tomczak and Max Welling. "VAE with a VampPrior". In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*. Ed. by Amos Storkey and Fernando Perez-Cruz. Vol. 84. Proceedings of Machine Learning Research. PMLR, Sept. 2018, pp. 1214–1223. URL: https://proceedings.mlr.press/v84/tomczak18a.html.

[47] Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. "Sample-Efficient Optimization in the Latent Space of Deep Generative Models via Weighted Retraining". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 11259–11272. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/81e3225c6ad49623167a4309eb4b2e75-Paper.pdf.

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

[49] Ekansh Verma and Souradip Chakraborty. "Uncertainty-aware labelled augmentations for high dimensional latent space Bayesian optimization". In: *NeurIPS 2021 workshop on deep generative models and downstream applications*. 2021.

[50] Hongxuan Wang et al. "Safe Bayesian Optimization for Complex Control Systems via Additive Gaussian Processes". In: *IEEE Robotics and Automation Letters* 10.11 (2025), pp. 11538–11545. DOI: 10.1109/LRA.2025.3612756.

[51] Yunyue Wei et al. "Safe Bayesian optimization for the control of high-dimensional embodied systems". In: *arXiv preprint arXiv:2412.20350* (2024).

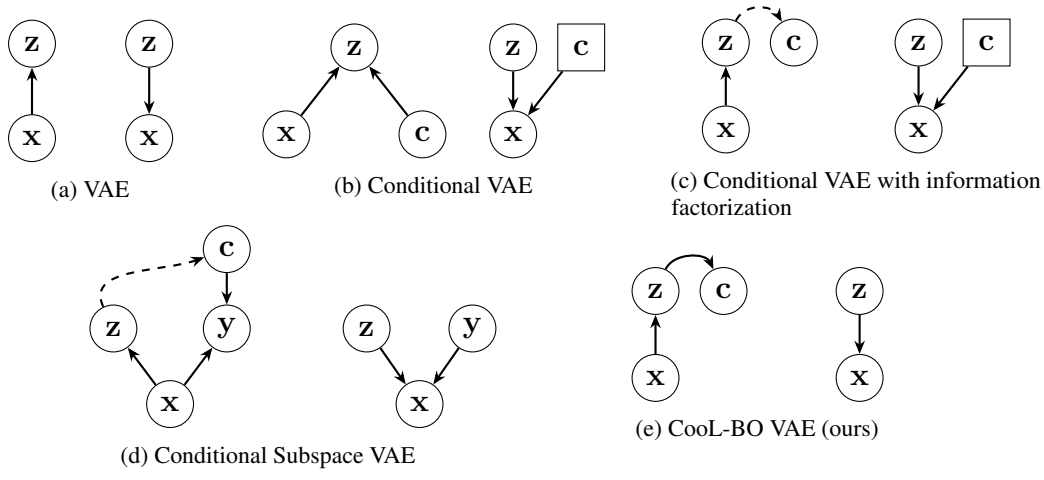
[52] Zhe Xie et al. "Adversarial and contrastive variational autoencoder for sequential recommendation". In: *Proceedings of the web conference 2021*. 2021, pp. 449–459.

[53] Xu Yang et al. "Deep spectral clustering using dual autoencoder network". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4066–4075.

[54] Yunxiang Zhang, Xiangyu Zhang, and Peter Frazier. "Constrained Two-step Look-Ahead Bayesian Optimization". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 12563–12575. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/685217557383cd194b4f10ae4b39eebf-Paper.pdf.

APPENDIX

A COMPARISON OF VAE ARCHITECTURES



(a) VAE (b) Conditional VAE (c) Conditional VAE with information factorization

(d) Conditional Subspace VAE (e) CooL-BO VAE (ours)

Figure 7: Conditioning variables in the encoder (left) and the decoder (right) in the respective VAE architectures. The variable $\mathbf{c}$ denotes the feasibility of the input $\mathbf{x}$. A square node represents the information unknown prior to decoding the latent vector during latent space BO. The dashed line indicates an adversarial objective with respect to predicting the output variable.

We highlight the conceptual differences of the proposed CooL-BO VAE with respect to the conditional VAEs proposed in the literature. Figure 7 illustrates the conditioning variables for the encoder and decoder blocks of the respective VAEs. In the case of a vanilla VAE (Figure 7a), both the encoder and decoder remain oblivious to the feasibility information, and thus the vanilla VAE does not explicitly encourage separability of the feasible and infeasible inputs in the latent domain.

In the case of the conditional VAE [23] (Figure 7c), both the encoder and the decoder are conditioned on the feasibility information ($\mathbf{c}$) to produce the outputs. However, we conjecture that such an architecture may not necessarily lead to a disentangled latent representation, as the decoder can use the feasibility information to reconstruct the input. Furthermore, as the dimensionality of the inputs increases, the contribution of the feasibility information on the reconstruction error becomes marginal. The information factorization [8] (Figure 7c) uses a discriminator with an adversarial objective in predicting the feasibility information given the latent representation, thus, learning a latent domain that is invariant to the known attributes. A key limitation of the aforementioned architectures with regard to application in latent space BO is that the feasibility information of arbitrary latent candidates proposed during BO is not known prior to decoding the latent representation, which makes these architectures unsuitable for latent space BO.

The conditional subspace VAE [24] (Figure 7d) learns higher-dimensional latent features ($\mathbf{y}$) correlated with the know binary attributes. However, leveraging the VAE for latent space BO would require jointly optimizing over the domain $\mathcal{Z} \times \mathcal{W}$, thus, increasing the computational overhead in proposing the latent candidates.

CooL-BO (Figure 7e) addresses these limitations by employing a discriminator that forces the encoder to learn a separable latent domain with respect to the feasibility of the inputs. The decoder in the case of CooL-BO is conditioned only on the latent vector, thus making it amenable to decoding the candidates proposed during latent BO.

B TASK DESCRIPTIONS

B.1 2D TOY EXPERIMENT

The 2D toy experiment used in Table 1 involves finding the global maximizer $\mathbf{x}^* = (\frac{1}{2}, \frac{1}{2})$ of the function $f(\mathbf{x}) = -||\mathbf{x} - \frac{1}{2}||_2^2$ with Thompson sampling as the acquisition function on the domain $\mathcal{X} = [0, 1]^2$. We emulate an imperfect latent domain of a VAE by randomly placing $N \in \{5, \ldots, 100\}$ infeasible regions in the input domain, always ensuring that the global optimum is feasible. Each infeasible region is a circle of radius 0.02 (Figure 8).

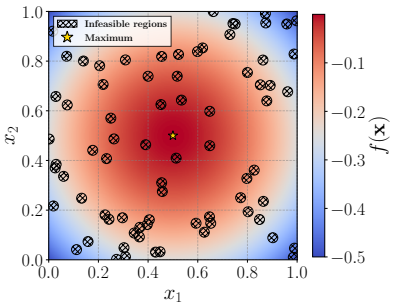


Figure 8: 2D toy example with 75 infeasible regions.

B.2 SYNTHETIC BENCHMARK TASKS

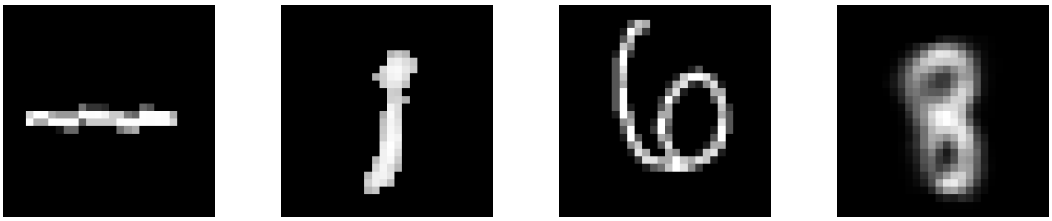
For the synthetic benchmark BO tasks, we project the inputs in the problem domain $\boldsymbol{\omega} \in \mathcal{W} \subset \mathbb{R}^w$ to an augmented input domain $\mathcal{X} \subset \mathbb{R}^d$ $(d > w)$ using a transformation h as $\mathbf{x} = h(\boldsymbol{\omega})$, and let the dimensionality of the latent domain $l = w - 1$.

Table 2: High-dimensional transformation function and constraint functions for the benchmark synthetic tasks. $\mathbf{U}_{m\times n}$ is a block matrix whose elements sampled from a uniform distribution $\mathcal{U}[0, 1]$ such that $\mathbf{U}_{m\times n}\mathbf{1}_{n\times 1} = \mathbf{1}_{m\times 1}$, and f_{NN} and g_{NN} are neural networks.

Task	**Problem domain** (w)	**Input domain** (d)	**Transformation function** (h)	**Constraints** (g_i)
`Ackley20h`	5	20	$h(\omega) = \boldsymbol{\omega} \begin{bmatrix} \mathbf{I}_{w\times w} \\ \mathbf{U}_{(d-w)\times w} \end{bmatrix}^\top$	$g_1(\mathbf{x}) = -\|\|\mathbf{x}\|\|_2^2 + \frac{d}{3}(w^2 + wl + d^2)$ $g_2(\mathbf{x}) = -\sin(20\|\|\mathbf{x}\|\|) + \frac{1}{4}$
`Rosenbrock30h`	10	30		
`Keane50h`	30	50		
`Traj45h`	30	45		
`Levy100h`	50	100	$h(\boldsymbol{\omega}) = [\{\boldsymbol{\omega}^k\}_{k\leq 3}, \sin(\boldsymbol{\omega}), \cos(\boldsymbol{\omega}), \{\omega_i\omega_j\}_{i<j}]^\top$ $h(\boldsymbol{\omega}) = h_{\text{NN}}(\boldsymbol{\omega})$	$g_1(\mathbf{x}) = g_{\text{NN}}(\mathbf{x})$
`Powell200h`	75	200		

We consider both linear and nonlinear transformations for the problem and impose nonconvex constraints in the high-dimensional input domain, as shown in Table 2. Note that we evaluate the black-box objective function (f) and the known constraints $(g_i, i = 1, \ldots, I)$ in the high-dimensional input domain $\mathcal{X}$. We initialize the neural network functions f_{NN} and g_{NN} with uniform weights drawn from $\mathcal{U}[0, 1]$ to avoid clustering of vectors in the output space.

B.3 MNIST TASKS



(a) `MNIST-Rot` task best input in the initial BO dataset

(b) CooL-BO optimization output on `MNIST-Rot` task

(c) `MNIST-Match` task best input in the initial BO dataset

(d) CooL-BO optimization output on `MNIST-Match` task

Figure 9: CooL-BO optimization results on `MNIST-Rot` and `MNIST-Match`.

The `MNIST-Rot` task involves minimizing the angle by which a digit is rotated, subject to the constraint that the digit is of class 1:

$$\max_{\mathbf{x}\in\mathcal{X}} -|f_{\text{rot}}(\mathbf{x})| \tag{10}$$

$$\text{s.t:} \quad f_{\text{clf}}(\mathbf{x}) = 1, \tag{11}$$

where f_{rot} is a neural network that estimates the rotation of the digit (in rads), and f_{clf} is a neural classifier that predicts the class of the digit. We trained f_{rot} on an augmented MNIST dataset by applying random rotations to the images.

The `MNIST-Match` task involves synthesizing images to match target images of digits 3 and 8, subject to the constraint that the synthesized image is of class 0, 3, 6, 8, or 9:

$$\max_{\mathbf{x}\in\mathcal{X}} -\frac{||\mathbf{x}-\mathcal{T}_3||_1 + ||\mathbf{x}-\mathcal{T}_8||_1}{2\times 784} \tag{12}$$

$$\text{s.t:} \quad f_{\text{clf}}(\mathbf{x}) \in \{0,3,6,8,9\}, \tag{13}$$

where $\mathcal{T}_k$ is a target image of digit k.

Figure 9 shows sample inputs to BO and the outputs produced by CooL-BO on the respective MNIST tasks.

C VAE Pretraining Details

Table 3: Number of feasible and infeasible inputs in task datasets

Task	Num. of inputs	
	Feasible	Infeasible
`Guacamol`	621,748	651,330
`Expr`	50,651	49,347
`Ackley20h`	50,000	5,000
`Rosenbrock30h`	50,000	5,000
`Traj45h`	50,000	5,000
`Keane50h`	50,000	5,000
`Levy100h`	50,000	5,000
`Powell200h`	50,000	5,000
`MNIST-Rot`	10,000	10,000
`MNIST-Match`	10,000	10,000

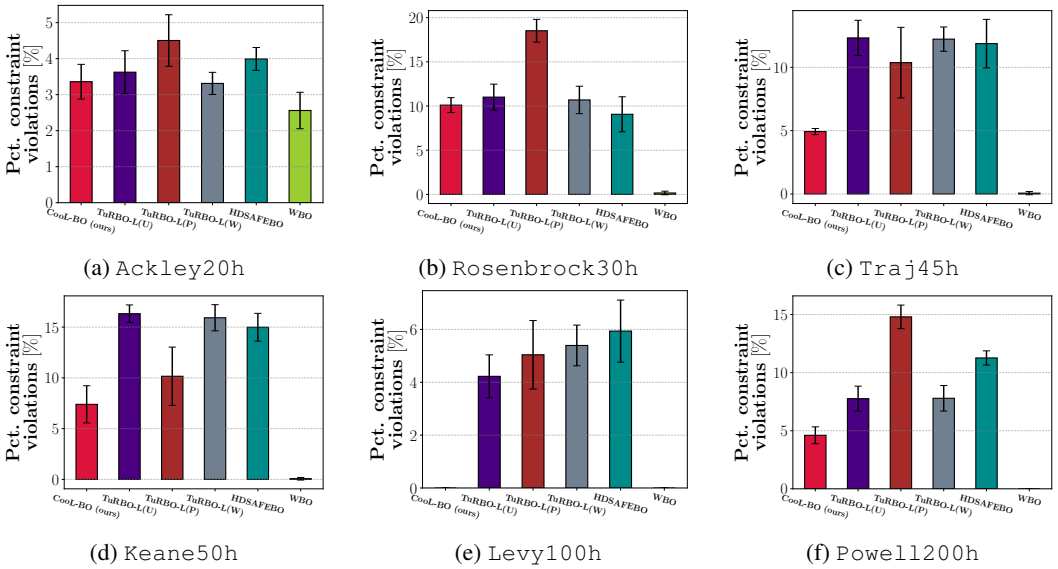
Table 4: Mean reconstruction error of pretrained VAEs used by the respective latent space BO methods. We report the reconstruction accuracy (Acc.) for the `Guacamol` and `Expr` tasks, and the mean squared error (MSE) in the case of all the other tasks.

Task name	TuRBO-L(U), TuRBO-L(W), WBO	HDSAFEBO, TuRBO-L(P)	CooL-BO
`Guacamol`	0.3058	0.3044	0.3055
`Expr`	0.9856	0.9861	0.9777
`Ackley20h`	0.1596	0.1603	0.1608
`Rosenbrock30h`	0.2237	0.2109	0.2544
`Traj45h`	6.8227	6.8387	7.0775
`Keane50h`	0.3309	0.3241	0.3889
`Levy100h`	0.2203	0.2109	0.2366
`Powell200h`	0.0219	0.0227	0.0222
`MNIST-Rot`	0.03144	0.03812	0.03649
`MNIST-Match`	0.04543	0.04721	0.044645

As the VAE training procedure of CooL-BO differs from that of the baselines, we provide details of the datasets used and report the reconstruction error achieved by the VAEs. In Table 3, we report the number of feasible and infeasible inputs in the respective datasets. For the synthetic benchmark tasks, we used an imbalanced dataset with a lower fraction of infeasible inputs; correspondingly, we weigh the infeasible samples by a factor of 10 in the BCE loss (equation 5). The VAE in the case of **TuRBO-L(U)** is trained on the feasible inputs only, whereas in the case of **CooL-BO**, **HDSAFEBO**, and **TuRBO-L(P)**, the datasets comprise both feasible and infeasible inputs. We pretrain the respective VAEs for the same number of training epochs to produce a comparable reconstruction metric, as listed in Table 4. We observe that CooL-BO achieves a comparable reconstruction metric despite the discriminative loss term imposed during the VAE pretraining phase.

VAE architectures. For the Guacamol tasks, we adopt the SELFIES VAE proposed by [34] and use a latent space of 256 dimensions. The `Expr` task uses the Grammar VAE from [25] with a 10-dimensional latent space. For the synthetic benchmark tasks, we use a deep variational autoencoder with three layers and `ReLU` activation function for the hidden layers.

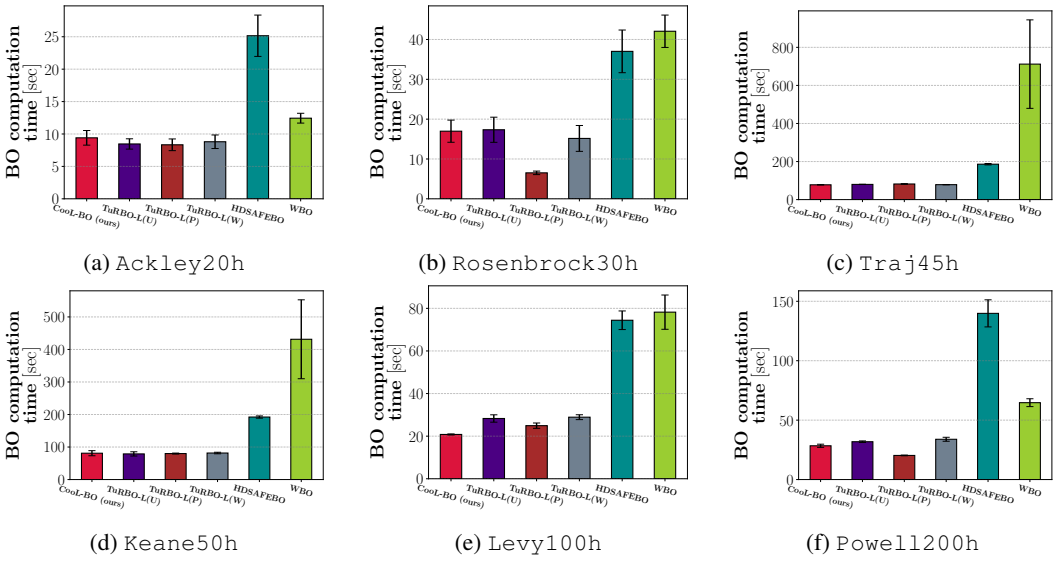
D BO Performance Results on Synthetic Benchmark Tasks



(a) `Ackley20h` (b) `Rosenbrock30h` (c) `Traj45h`

(d) `Keane50h` (e) `Levy100h` (f) `Powell200h`

Figure 10: Percentage of constraint violations during latent space BO in case of synthetic benchmark tasks.



(a) `Ackley20h` (b) `Rosenbrock30h` (c) `Traj45h`

(d) `Keane50h` (e) `Levy100h` (f) `Powell200h`

Figure 11: BO computation time in case of synthetic benchmark tasks.

E Impact of VAE Architecture on Learned Representations and BO Performance

In this section, we ablate the impact of the VAE architecture on the quality of the learned representations and on the performance of latent space BO. We choose a representative set of VAE architectures that incorporate known attributes for representation learning and employ dynamic KL priors. The first baseline is the conditional VAE [23] that conditions the learned latent representation on the feasibility information (Appendix A). The second baseline is the VampPrior [46] that learns the KL

priors dynamically. The third, contrastive VAE [1], imposes a contrastive loss to separate the feasible and infeasible inputs in the latent domain. We trained all the VAEs on the same datasets comprising the feasible and infeasible inputs.

E.1 Probability of Generating a Feasible Candidate via Posterior Sampling

To verify if the learned latent domain facilitates generating feasible candidates, we measure the empirical probability of producing a feasible input from a latent candidate sampled from the posterior of a feasible input in Table 5. Indeed, CooL-BO results in a higher probability of producing feasible candidates, owing to the discriminative pretraining of the VAE that forces the VAE to learn a separable latent domain with regard to the feasibility. Consequently, CooL-BO proposed feasible candidates with a higher probability during latent BO.

Table 5: Empirical probability of generating a feasible candidate from a candidate sampled from the latent posterior of a feasible input.

	Ackley20h	Rosenbrock30h	Traj45h	Keane50h	Levy100h	Powell200h
VAE	0.4076	0.2698	0.3532	0.1070	0.3217	0.1524
Contrastive VAE	0.2338	0.1300	0.1271	0.1610	0.3317	0.1654
Conditional VAE	0.4110	0.2203	0.2071	0.1230	0.2245	0.1559
VampPrior VAE	0.4410	0.1203	0.2521	0.1901	0.2457	0.1705
CooL-BO VAE	**0.4556**	**0.5166**	**0.4532**	**0.2474**	**0.3518**	**0.2231**

E.2 Separability of Latent Space Learned by VAEs

In Subsection 4.1, we imposed a discriminator penalty to shape the latent manifold to encourage the separability of the learned feasible and infeasible latent representations. Since VAEs attempt to preserve the semantic similarity of the input data in the latent space, a natural question arises whether a VAE with enough capacity can inherently learn a separable representation, given that feasible and infeasible sets in the input domain are mutually exclusive. We use Fisher's LDA ratio and accuracy score of a linear classifier to measure the separability between the latent vectors of the two classes. In Table 8, we compare the metrics for the latent domain learned by the standard VAE and the separable latent domain learned by CooL-BO. We observe that across all tasks, VAE training with our proposed approach achieves a better separability score than the baseline. It is worth noting that in the case of the MNIST-Rot task, the simplicity of the task allows a standard VAE to achieve a comparable BCE score to that of CooL-BO, however, with increased data diversity (tasks MNIST-Match, Expr, Guacamol), the latent representations learned by the standard VAE become more tangled, resulting in a lower separability score. A separable latent space impacts the ease of sampling feasible candidates during latent BO, thus impacting the sampling efficiency of BO.

Table 6: Comparison of separability of feasible and infeasible vectors in the latent domain learned by VAE and CooL-BO VAE.

Task	**VAE**			**CooL-BO VAE**		
	Fisher's LDA ratio (↑)	BCE (↓)	Acc. (↑)	Fisher's LDA ratio (↑)	BCE (↓)	Acc. (↑)
Guacamol	2.6378×10^{-5}	0.5525	0.7759	$\mathbf{6.0234 \times 10^{-3}}$	**0.2657**	**0.9327**
Expr	6.0460×10^{-6}	0.3743	0.6724	$\mathbf{8.8159 \times 10^{-4}}$	**0.0943**	**0.9549**
Ackley20h	0.0915	0.0874	0.9990	**0.1601**	**0.0427**	**0.9994**
Rosenbrock30h	1.9046×10^{-3}	0.4152	0.9223	**1.2615**	**0.0269**	**0.9990**
Traj45h	4.7022×10^{-5}	0.4330	0.8850	**0.2060**	**0.1746**	**0.9983**
Keane50h	0.0034	0.0435	0.9920	**1.5728**	**0.0075**	**1.0**
Levy100h	5.0580×10^{-4}	0.3658	0.9145	**0.1653**	**0.2092**	**0.9875**
Powell200h	1.3169×10^{-8}	0.1568	0.9981	$\mathbf{0.2341 \times 10^{-2}}$	**0.0467**	**1.0**
MNIST-Rot	1.1533	0.1223	0.9554	**2.2805**	**0.1006**	**0.9974**
MNIST-Match	0.1637	0.6122	0.9218	**3.3472**	**0.2661**	**0.9884**

Table 7: Comparison of separability of feasible and infeasible vectors in the latent domain learned by condtitional VAE and VampPrior VAE.

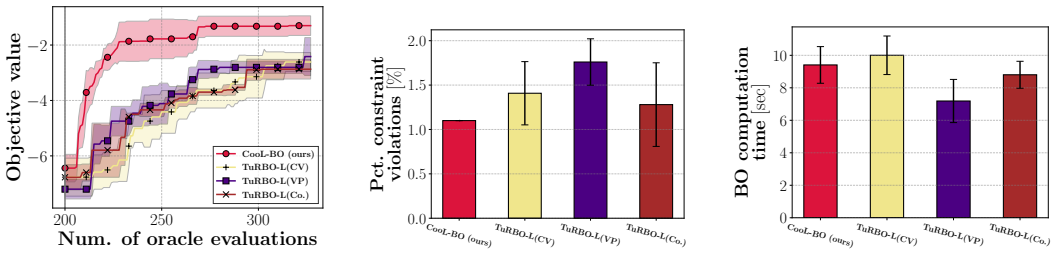
Task	Conditional VAE			VampPrior VAE		
	Fisher's LDA ratio (↑)	BCE (↓)	Acc. (↑)	Fisher's LDA ratio (↑)	BCE (↓)	Acc. (↑)
Guacamol	1.3238×10^{-4}	0.4221	0.7889	6.0234×10^{-4}	0.3612	0.8711
Expr	4.1135×10^{-5}	0.2145	0.7534	8.8159×10^{-4}	0.0943	0.9549
Ackley20h	0.0633	0.3124	0.9980	0.1028	0.1757	0.9984
Rosenbrock30h	1.4276×10^{-3}	0.3671	0.9322	5.34×10^{-3}	0.4226	0.9173
Traj45h	4.7022×10^{-5}	0.4330	0.8850	9.5143×10^{-5}	0.4512	0.8692
Keane50h	0.0034	0.0435	0.9998	1.7501×10^{-4}	0.1515	0.9872
Levy100h	1.2385×10^{-5}	0.02588	0.9998	1.2385×10^{-4}	0.4081	0.8975
Powell200h	3.7299×10^{-4}	0.0814	0.9991	3.3312×10^{-5}	0.2196	0.9968
MNIST-Rot	1.1533	0.1223	0.9554	1.5325	0.2106	0.9612
MNIST-Match	1.1533	0.1223	0.9554	1.7123	0.4711	0.9124

Table 8: Comparison of separability of feasible and infeasible vectors in the latent domain learned by contrastive VAE.

Task	Contrastive VAE		
	Fisher's LDA ratio (↑)	BCE (↓)	Acc. (↑)
Guacamol	2.2214×10^{-5}	1.1657	0.8127
Expr	1.2359×10^{-5}	0.2143	0.8559
Ackley20h	2.2221×10^{-4}	0.3167	0.9881
Rosenbrock30h	1.6743×10^{-4}	0.4240	0.9211
Traj45h	1.3601×10^{-6}	0.5227	0.8337
Keane50h	4.6124×10^{-5}	0.5201	0.8341
Levy100h	1.9642×10^{-5}	0.5212	0.8323
Powell200h	1.0941×10^{-8}	0.5197	0.8337
MNIST-Rot	1.1605	0.3126	0.8214
MNIST-Match	0.7572	0.6671	0.8884

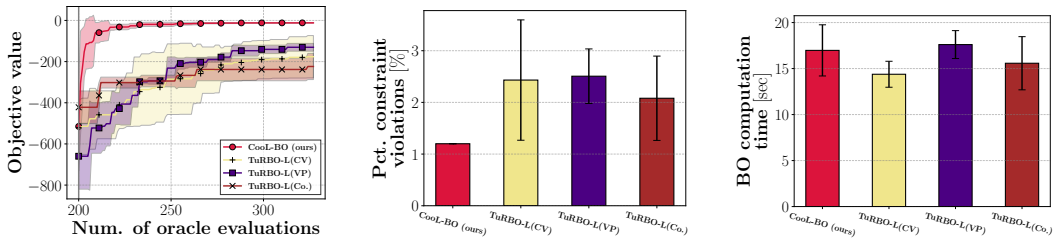
E.3 Impact of VAE Architecture on BO Performance

In this section, we report the impact of the VAE architecture on the attained BO objective during latent space BO. The methods TuRBO(CV), TuRBO(VP), and TuRBO(Co) use the conditional variational VAE, VampPrior, and contrastive VAE architectures, respectively. The baselines use TuRBO for latent space BO (same as CooL-BO) and impose the feasibility constraint in the latent domain using a surrogate GP. We plot the BO optimization results in Figures 12 - 17. We observe that CooL-BO proposes a higher fraction of feasible candidates and attains a higher BO objective across all tasks, owing to the disentangled latent space learned by CooL-BO.



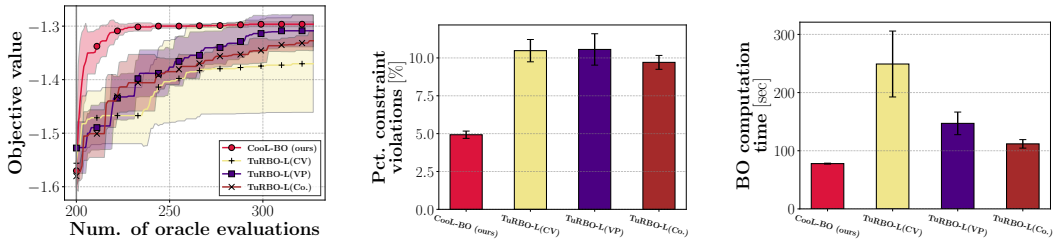
(a) (b) (c)

Figure 12: Optimization results on Ackley20h task.



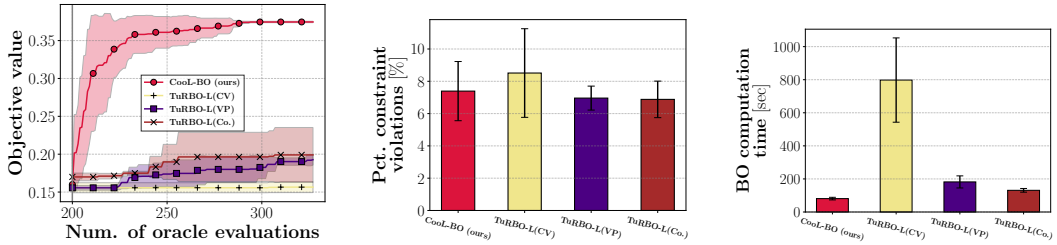
(a) (b) (c)

Figure 13: Optimization results on `Rosenbrock30h` task.



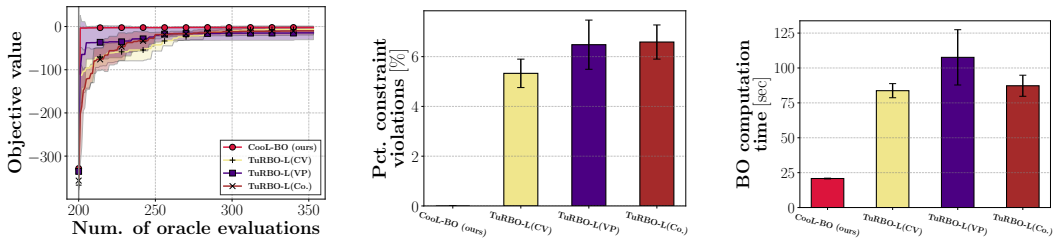
(a) (b) (c)

Figure 14: Optimization results on `Traj45h` task.



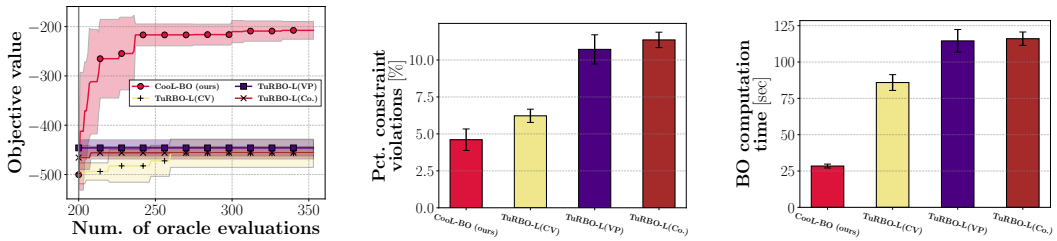
(a) (b) (c)

Figure 15: Optimization results on `Keane50h` task.



(a) (b) (c)

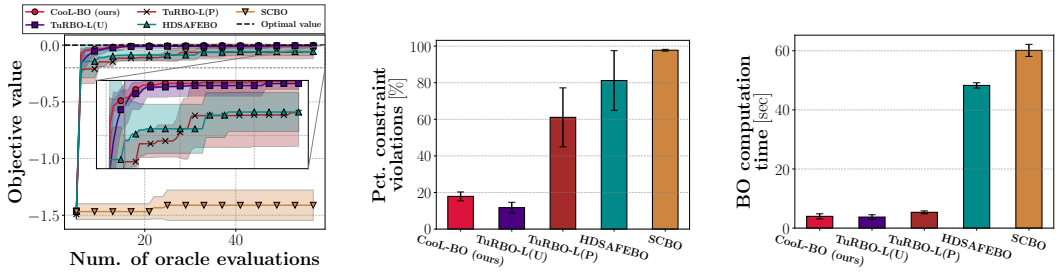
Figure 16: Optimization results on `Levy100h` task.



(a) (b) (c)

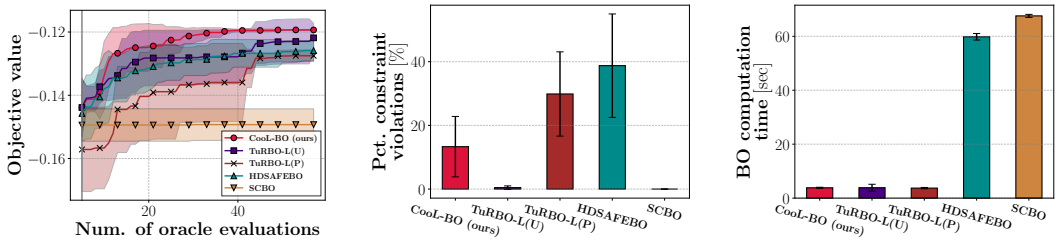
Figure 17: Optimization results on `Powell200h` task.

F RESULTS ON MNIST TASKS



(a) (b) (c)

Figure 18: Optimization results on `MNIST-Rot` task.



(a) (b) (c)

Figure 19: Optimization results on `MNIST-Match` task.

We plot the optimization results for the `MNIST-Rot` and `MNIST-Match` tasks in Figures 18 and 19, respectively. Note that `SCBO` fails to improve upon the initial best, which indicates the complexity of the high-dimensional tasks considered. Again, CooL-BO achieves a higher objective at a comparable computation time of solving an unconstrained optimization problem.

It is worth noting that in case of `MNIST-Rot` and `MNIST-Match` tasks, `TuRBO-L(U)` incurs lower constraint violations during BO, as observed in Figures 18b and 19b. A plausible reason for this is the restrictiveness of the constraints imposed. For instance, in the case of `MNIST-Rot` task, a VAE trained only on the images of the digit 1 is unlikely to generate a digit belonging to a different class, thus resulting in lower constraint violations during latent space BO. Nevertheless, CooL-BO achieves lower constraint violations than the other constrained latent space BO methods, and attains a higher objective value compared to the baselines.

G Analysis of Latent Candidates Proposed by CooL-BO

To investigate whether the objective and the constraint are correlated in the latent domain learned by CooL-BO, we compute the percentile of the distance of the best candidate (with respect to the training dataset) from the boundary of the discriminator in the latent domain, as shown in Tables 9 and 10. The results show that the best candidate is not always farthest from the discriminator hyperplane, i.e., the constraints in the latent domain need not necessarily be aligned with the objective function. Irrespective of the alignment of the objective values and the constraints in the latent domain, CooL-BO is able to search for the best candidate with the default feasibility threshold of $\delta = 0.5$.

Table 9: Percentile of distance of the best latent candidate from the linear discriminator ($\delta = 0.5$) for `Guacamol`, `Expr`, and `MNIST` tasks.

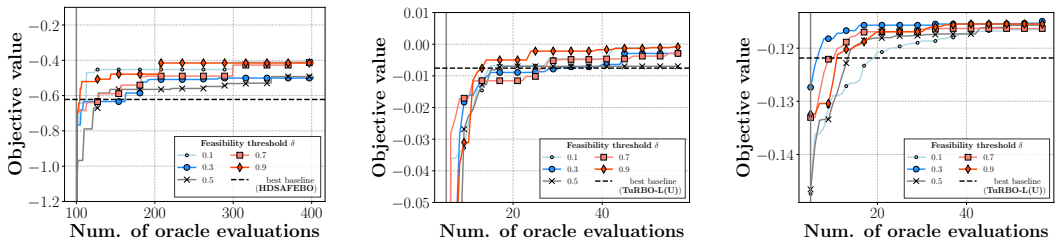
	`Guacamol-logP`	`Guacamol-Rano`	`Expr`	`MNIST-Rot`	`MNIST-Match`
Percentile of distance of the best latent candidate	100	100	65	30	50

Table 10: Percentile of distance of the best latent candidate from the linear discriminator ($\delta = 0.5$) for synthetic benchmark tasks.

	`Ackley20h`	`Rosenbrock30`	`Traj45h`	`Keane50h`	`Levy100h`	`Powell200h`
Percentile of distance of the best latent candidate	85	71	100	99	5	100

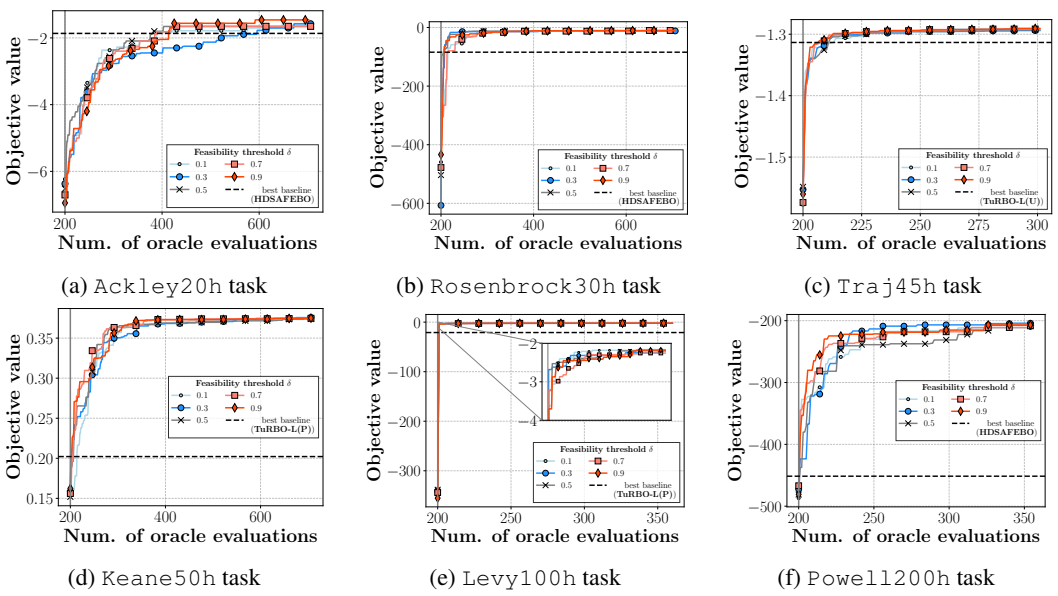
H Impact of Feasibility Threshold on CooL-BO Performance

In Subsection 4.2, we introduced $\delta \in [0, 1]$ as a threshold for feasibility sampling. A higher feasibility threshold for the discriminator in (9) confines the latent domain to a region yielding feasible inputs with a higher probability, while a lower feasibility threshold expands the search space for latent space BO, potentially impacting the sampling efficiency. In Figures 20a - 20c, we empirically evaluate the impact of δ on the objective value attained by CooL-BO. Indeed, the sampling efficiency of CooL-BO is impacted by the feasibility threshold. Nevertheless, our approach achieves a higher objective than the next best baseline.



(a) `Expr` task (b) `MNIST-Rot` task (c) `MNIST-Match` task

Figure 20: Impact of the feasibility threshold δ on the BO objective achieved by CooL-BO for `Expr` and `MNIST` tasks.



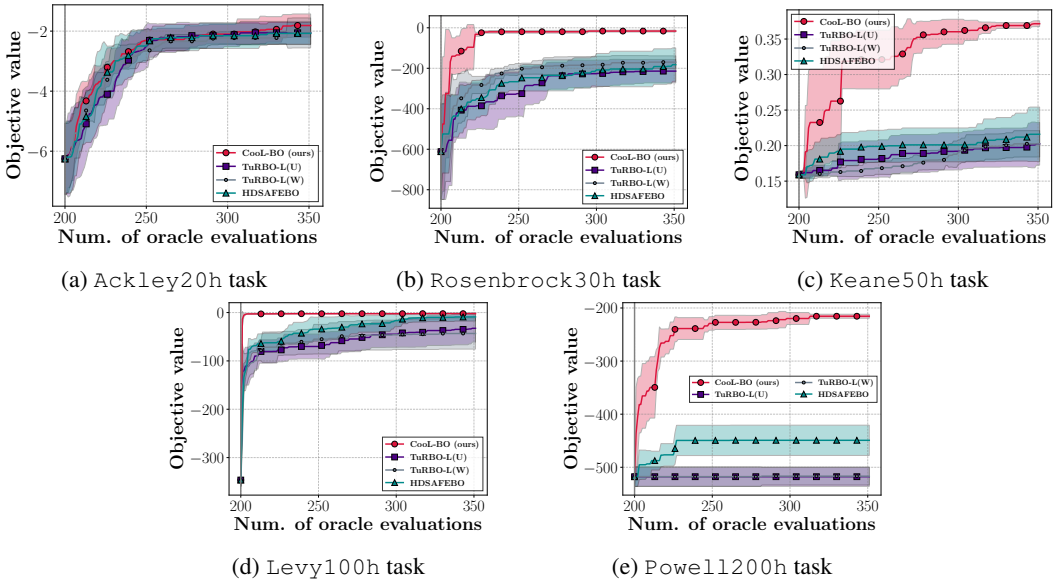
(a) `Ackley20h` task (b) `Rosenbrock30h` task (c) `Traj45h` task

(d) `Keane50h` task (e) `Levy100h` task (f) `Powell200h` task

Figure 21: Impact of feasibility threshold δ on BO objective achieved by CooL-BO for synthetic tasks.

I Impact of Batch Size on BO Performance

The baselines and CooL-BO result in different fractions of infeasible candidates during BO (Figures 10), hence the number of input samples used for the posterior update for the surrogate GP may not be the same. In Figure 22, we evaluate the BO performance by setting the batch size to $q = 1$, ensuring consistency between the surrogate updates for the different methods. We observe that CooL-BO achieves a higher BO objective compared to the baselines across all tasks.



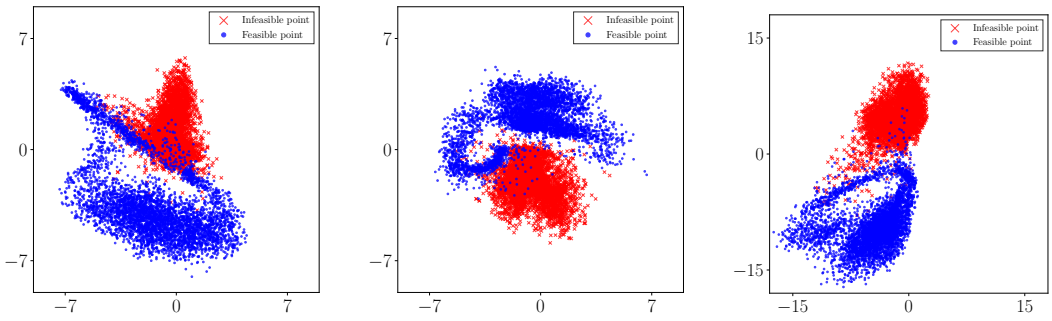
(a) `Ackley20h` task (b) `Rosenbrock30h` task (c) `Keane50h` task

(d) `Levy100h` task (e) `Powell200h` task

Figure 22: BO performance on synthetic benchmark tasks with batch size $q = 1$.

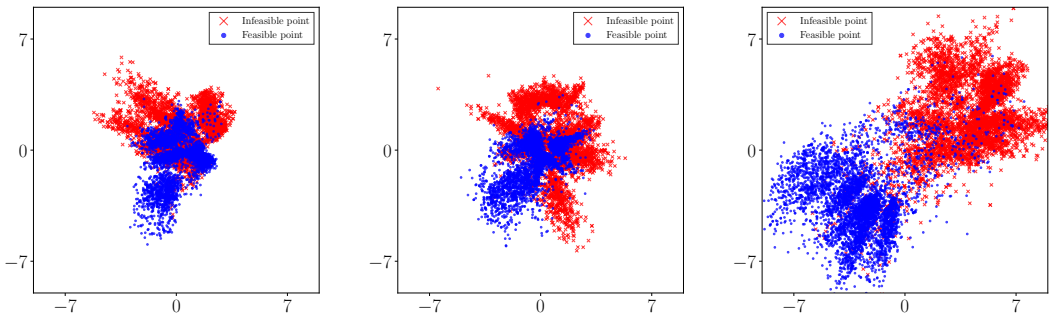
J VISUALIZATION OF LATENT SPACES LEARNED BY VAES AND COOL-BO VAE ON MNIST DATASET

In Figures 23 and 24 we compare the latent representations learned by the VAE architectures. We observe that the conditional VAE learns to map the feasible and infeasible inputs to non-overlapping regions in the case of the `MNIST-Rot` dataset (Figures 23b). However, with increased data-diversity (`MNIST-Match` task), the latent domain learned by the conditional VAE becomes tangled with respect to the feasibility. In contrast, CooL-BO learns a separable latent domain (Figures 23c and 24c).



(a) VAE (b) Conditional VAE (c) CooL-BO VAE

Figure 23: Comparison of latent spaces learned by the respective VAEs for `MNIST-Rot` task.



(a) VAE (b) Conditional VAE (c) CooL-BO VAE

Figure 24: Comparison of latent spaces learned by the respective VAEs for `MNIST-Match` task.

K LIMITATIONS

While CooL-BO introduces a notion of feasibility threshold to constrain the latent space BO optimization, BO tasks that impose a higher feasibility requirement may require careful hyperparameter tuning to trade off sampling of BO for a fraction of feasible candidates during BO. Interesting future work includes leveraging generative modeling, as opposed to the discriminative approach employed by CooL-BO, to simultaneously model and shape the latent domain with regard to feasibility, making it amenable to BO problems with unknown constraints (safe BO).

L HYPERPARAMETERS

BO. In all our experiments, we use an exact GP model with a Matérn-5/2 kernel with adaptive length scaling to fit the surrogate models. For the BO results in Section 5, we use a batch size of 2 in the case of `Guacamol-logP`, `Guacamol-Rano`, and `Expr` tasks, and a batch size of 5 for the synthetic benchmark tasks. All hyperparameters with regards to trust region dynamics are set to `TuRBO` defaults [11] for all methods.

All BO experiments were performed on a Dell PowerEdge R715 equipped with an AMD EPYC 7543P CPU.