HIGH-DIMENSIONAL BLACK-BOX OPTIMIZATION VIA GUIDANCE SAMPLING OF DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many real-world optimization problems involve optimizing black-box objectives for which only a limited number of evaluations are available. Bayesian optimization (BO) is a powerful framework for such problems, but its effectiveness often deteriorates in high-dimensional spaces. Recent work leverages diffusion models for black-box optimization, often through pre-training a reward-conditioned score network to steer generation towards high-reward samples. Alternatively, gradient-based guidance methods assume access to a differentiable reward function from which guidance gradients are queried. However, these approaches either require large amounts of labeled data for offline training or access to exact reward gradients, limiting their applicability in black-box settings. We propose BREAD (**B**lack-box **Re**ward-guided S**a**mpling of **D**iffusion Model), a guidance sampling framework for optimizing black-box rewards without assuming access to a labeled dataset or reward gradients. The proposed method uses a Gaussian process (GP) as a differentiable surrogate for the unknown objective and actively queries labels for the generated samples. We provide a theoretical analysis of the total-variation distance between the generated distribution and the target distribution. Using Upper Confidence Bound (UCB) as guidance signals, our method balances exploration and exploitation during generation and enables the discovery of high-reward regions in the data space. Numerical experiments on scientific discovery and image generation tasks demonstrate that our framework can effectively optimize black-box rewards in low-data scenarios.

1 INTRODUCTION

Optimizing expensive black-box functions arises in many real-world applications such as robotics (Tesch et al., 2013), material design (Khatamsaz et al., 2023), and drug discovery (Pyzer-Knapp, 2018). In the black-box setting, the objective function is unknown and can only be evaluated through costly queries. Bayesian optimization (Snoek et al., 2012; Shahriari et al., 2016; Frazier, 2018; Schneider, 2016; Berkenkamp et al., 2023, BO) has emerged as a powerful framework for optimizing such black-box functions. Despite its success, BO often struggles to scale to high-dimensional spaces, where modeling the objective and searching for promising query points become increasingly challenging (Malu et al., 2021; Binois & Wycoff, 2022).

Meanwhile, diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) have demonstrated remarkable capabilities in modeling complex data distributions. Controlled generation methods (Dhariwal & Nichol, 2021; Ho & Salimans, 2021; Chung et al., 2023; Bansal et al., 2024) further enable sample generation with desired properties, which motivates recent work on diffusion model-based methods for data-driven black-box optimization (Krishnamoorthy et al., 2023; Yuan et al., 2023; Li et al., 2024; Wu et al., 2024; Kong et al., 2025; Yun et al., 2025). Due to limited space, we defer a detailed discussion of related work to Appendix A. Existing approaches typically require large amounts of labeled data to learn a reward model or a reward-conditioned score network, making them challenging to apply when only a small number of reward evaluations, e.g., on the order of 50, are available. For example, a user may wish to personalize image generation to their preferences after providing only a few rounds of feedback (Figure 1). In scientific discovery applications such as nuclear fusion and protein design, labeled data can be scarce and costly to obtain. This motivates the following question:

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.

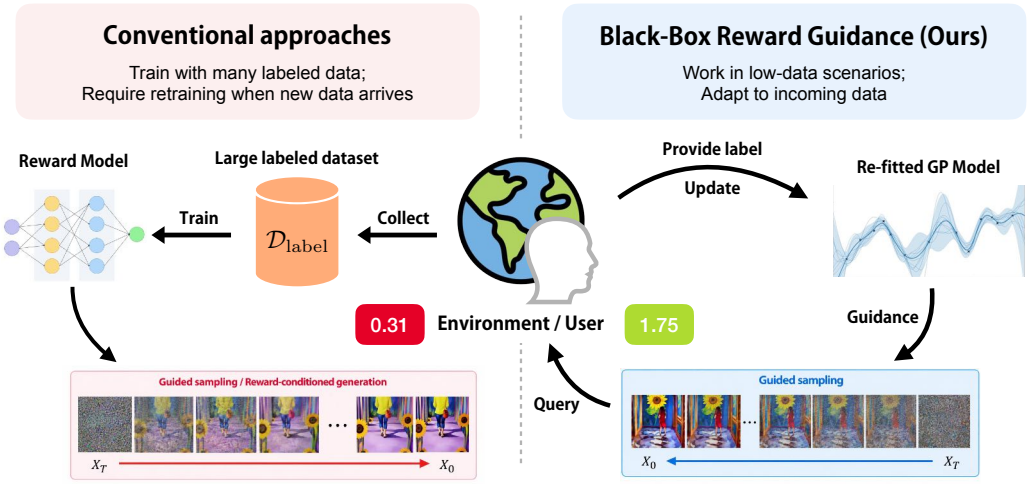


Figure 1: **Black-box Reward-guided Sampling of Diffusion Models (BREAD).** By actively exploring and querying reward labels in promising regions of the (embedded) sample space, BREAD can effectively optimize black-box rewards in high-dimensional but low-data regimes.

Can we optimize black-box rewards in high-dimensional space using only a few evaluations?

In this paper, we propose BREAD, shorthand for **B**lack-box **Re**ward-guided S**a**mpling of **D**iffusion Model, a framework for optimizing black-box reward functions in high-dimensional data spaces. BREAD combines a diffusion model that is pre-trained on an *unlabeled* dataset with a Gaussian process (GP) that models the unknown reward function from limited labeled data. We introduce two mechanisms for guided generation from the pre-trained diffusion model. The first, *reward-directed sampling*, uses reward-conditioned gradients derived from the fitted GP to guide generation toward samples that yield a target reward. We provide a theoretical analysis that bounds the total-variation distance (TV) between the resulting guided distribution and the target distribution. The second, *uncertainty-guided sampling*, incorporates Upper Confidence Bound (UCB) into guidance to encourage exploration toward sample space with both high predicted rewards or high uncertainty. To effectively model the black-box objectives in high-dimensional data spaces, we leverage deep kernel learning (Wilson et al., 2015) to construct expressive GP kernels over learned low-dimensional representations. We evaluate BREAD on several image generation tasks and simulated fusion plasma control. Our results demonstrate that the proposed framework can effectively optimize black-box rewards with only a small number of labeled samples, achieving performance comparable to guidance methods with access to the exact reward gradient. Moreover, UCB-based exploration improves sample efficiency and enables more effective discovery of high-reward regions.

Our contributions are summarized as follows.

- We propose a guidance sampling framework for black-box optimization in high-dimensional spaces. By combining a pre-trained diffusion model with a GP that models the black-box objective, our method effectively generates high-fidelity data with desired properties using only a few rounds of evaluations.
- We introduce two subroutines for guiding generation toward high-reward samples. *Reward-directed sampling* uses the GP likelihood conditioned on a target reward as the guidance signal. *Uncertainty-guided sampling* uses the UCB as the guidance signal to balance exploitation and exploration in the sample space.
- We provide a theoretical analysis of the total-variation distance between the generated distribution and the target distribution.
- Numerical experiments on image generation tasks and simulated fusion plasma control show that our method can effectively optimize black-box rewards with only a small number of reward evaluations and achieve performance comparable to guidance methods that have access to the exact reward gradients.

2 PRELIMINARIES

2.1 SCORE-BASED GENERATIVE MODELS

Given samples from a data distribution $q_{\text{data}} \in \Delta(\mathcal{X})$, where $\mathcal{X} \subset \mathbb{R}^d$ is the sample space, we aim to learn a model distribution p_θ that approximates q_{data} and is easy to sample from. Score-based generative models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021, SGMs) define a forward SDE from time 0 to T

$$\mathrm{d}X_t = -g^2(t)X_t\mathrm{d}t + \sqrt{2}g(t)\mathrm{d}B_t, \text{ where } X_0 \sim q_{\text{data}}. \tag{1}$$

Here, $g : \mathbb{R}_+ \to \mathbb{R}$ is a positive smooth function, and $(B_t)_{t\geq 0}$ is the standard Wiener process in $\mathbb{R}^d$. Let $q_t := \text{law}(X_t)$. We have that $q_t(X_t|X_0) = \mathcal{N}(X_t; \sqrt{\bar{\alpha}_t}X_0, (1-\bar{\alpha}_t)I)$, where we define $\bar{\alpha}_t := \exp(-2\int_0^t g^2(s)\mathrm{d}s)$. Under regularity conditions, the forward process has an equivalent backward SDE (Anderson, 1982) from time 0 to T

$$\mathrm{d}X_t^{\leftarrow} = [g^2(T-t)X_t^{\leftarrow} + 2g^2(T-t)\nabla \log q_{T-t}(X_t^{\leftarrow})]\mathrm{d}t + \sqrt{2}g(T-t)\mathrm{d}B_t^{\leftarrow}, \text{ where } X_0^{\leftarrow} \sim q_T, \tag{2}$$

where $(B_t^{\leftarrow})_{t\in[0,T]}$ is the reversed Brownian motion, and $\nabla \log q_t$ is the *score function* for q_t. Particularly, the marginal distribution of $X_t^{\leftarrow}$ follows q_{T-t} for any $t \in [0,T]$. By Tweedie's formula (Efron, 2011), the posterior mean of X_0 conditioned on $X_t^{\leftarrow}$ is given by

$$m_{T-t}(X_t^{\leftarrow}) := \mathbb{E}_{q_0(\cdot|X_t)}[X_0] = \frac{1}{\sqrt{\bar{\alpha}_{T-t}}}(X_t^{\leftarrow} + (1-\bar{\alpha}_{T-t})\nabla \log q_{T-t}(X_t^{\leftarrow})) \tag{3}$$

To estimate the score function, denoising score matching minimizes the loss

$$\min_\theta \mathbb{E}_{t\sim\mathcal{U}([0,T]),X_0\sim q_{\text{data}},X_t\sim q_t(\cdot|X_0)}[w(t)\|s_t^\theta(X_t) - \nabla_{X_t}\log q_t(X_t|X_0)\|^2], \tag{4}$$

where $w : \mathbb{R}_+ \to \mathbb{R}_+$ is a weighting function. We define the predicted posterior-mean

$$\hat{x}_0^\theta(x_t^{\leftarrow}, T-t) = \frac{x_t^{\leftarrow} + (1-\bar{\alpha}_{T-t})s_{T-t}^\theta(x_t^{\leftarrow})}{\sqrt{\bar{\alpha}_{T-t}}}. \tag{5}$$

2.2 GUIDED GENERATION FROM DIFFUSION MODELS

Given a pre-trained SGM, guided generation aims to generate a sample X with a desired property $Y = y$ from the distribution $P(X|Y=y)$. To this end, we need to learn the conditional score function $\nabla_{x_t}\log q_t(x_t|y)$. By Bayes' rule, we have

$$\nabla_{x_t}\log q_t(x_t|y) = \nabla_{x_t}\log q_t(x_t) + \nabla_{x_t}\log q_t(y|x_t), \tag{6}$$

where the unconditional score can be estimated by the pre-trained score network, i.e., $s_t^\theta(x_t) \approx \nabla_{x_t}\log q_t(x_t)$. To estimate $\nabla_{x_t}\log q_t(y|x_t)$, classifier guidance (Dhariwal & Nichol, 2021) trains auxiliary classifiers on noisy samples x_t with discrete labels and then uses the gradient of of its log conditional probability as the guidance score. Since such classifiers are domain-specific and generally not readily available, classifier-free guidance (Ho & Salimans, 2021, CFG) instead jointly trains conditional and unconditional diffusion models and combines their score estimates during generation. However, CFG requires the diffusion model to be trained with the desired conditioning information and does not readily accommodate new guidance modalities without retraining. To address this issue, diffusion posterior sampling Chung et al. (2023, DPS) and forward universal guidance (Bansal et al., 2024, Algorithm 1) approximate the conditional score by

$$\nabla_{x_t}\log q_t(y|x_t) \approx \nabla_{x_t}\log q_0(y|m_t(x_t)). \tag{7}$$

2.3 BAYESIAN OPTIMIZATION

Let $\mathcal{X} \subset \mathbb{R}^d$ be a compact domain. Bayesian optimization (Frazier, 2018, BO) aims to solve $\max_{x\in\mathcal{X}} f(x)$, where $f : \mathcal{X} \to \mathbb{R}$ is the objective function. BO consists of two main components: a Bayesian statistical model for modeling the objective function, and an acquisition function for deciding where to sample next. One common choice for the Bayesian statistical model is the Gaussian

Algorithm 1 FORWARDUNIVERSALGUIDANCE (Bansal et al., 2024)

Require: guidance function $f : \mathcal{X} \to \mathbb{R}$, guidance strength $\eta : \mathbb{R}_+ \to \mathbb{R}_+$

1: Sample $X_0^{\leftarrow} \sim \mathcal{N}(0, I)$.
2: **for** denoising step $k = 1, \cdots, \lfloor \frac{T}{h} \rfloor$ **do**
3: Sample

$$\mathrm{d}X_t^{\leftarrow} = [g^2(T-kh)X_{kh}^{\leftarrow} + 2g^2(T-kh)\tilde{s}_{T-kh}(X_{kh}^{\leftarrow})]\mathrm{d}t + \sqrt{2}g(T-kh)\mathrm{d}B_{kh}^{\leftarrow},$$

for $t \in [kh, (k+1)h)$, where

$$\tilde{s}_{T-t}(x_t^{\leftarrow}) = s_{T-t}^{\theta}(x_t^{\leftarrow}) + \eta(t)\nabla_{x_t^{\leftarrow}} f\left(\hat{x}_0^{\theta}(x_t^{\leftarrow}, T-t)\right). \tag{8}$$

4: **end for**
5: **return** $X_T^{\leftarrow}$

Process (Rasmussen & Williams, 2005, GP). A $\mathcal{GP}(\mu(x), \kappa(x, x'))$ is specified by its mean function $\mu(x) = \mathbb{E}[f(x)]$ and covariance (or kernel) function $\kappa(x, x') = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x')]$. Without loss of generality, we assume that $\mu(x) = 0$ for GPs not conditioned on data. Given noisy evaluations $\{y_n\}_{n=1}^N$ at a collection of points $\{x_n\}_{n=1}^N$, where $y_n = f(x_n) + \varepsilon_n$ with i.i.d. Gaussian noise $\varepsilon_n \sim \mathcal{N}(0, \tau^2)$, the conditional distribution of $f(x)$ is a Gaussian distribution $\mathcal{N}(\mu_N(x), \sigma_N^2(x))$, where

$$\mu_N(x) = \kappa(x, x_{1:N})\left(\kappa(x_{1:N}, x_{1:N}) + \tau^2 I\right)^{-1} y_{1:N}, \tag{9}$$

$$\sigma_N^2(x) = \kappa(x, x) - \kappa(x, x_{1:N})\left(\kappa(x_{1:N}, x_{1:N}) + \tau^2 I\right)^{-1} \kappa(x_{1:N}, x). \tag{10}$$

To sample the next point x_{N+1} for evaluation, the GP-UCB algorithm (Srinivas et al., 2010) proposes to use the Upper Confidence Bound (UCB) as an acquisition function to select the candidate

$$x_{N+1} = \arg\max_{x \in \mathcal{X}} \mu_N(x) + c_N \sigma_N(x), \tag{11}$$

where c_N is a hyperparameter. The UCB acquisition function prefers points whose objective value is uncertain (large $\sigma_N(\cdot)$) and expected to be high (large $\mu_N(\cdot)$). GP-UCB has demonstrated strong performance in low- to moderate-dimensional optimization problems. However, directly optimizing the UCB acquisition function becomes challenging in high-dimensional data spaces. Instead, our method uses UCB to guide exploration toward promising regions of the data space, while leveraging a pre-trained diffusion model to generate high-fidelity samples.

3 GUIDANCE SAMPLING FOR HIGH-DIMENSIONAL BLACK-BOX OPTIMIZATION

Existing diffusion model-based black-box optimization methods often rely on a sufficiently large reward-labeled dataset to train a reward model or a reward-conditioned score network. In many real-world applications, however, one often has access to abundant unlabeled data while reward labels must be actively acquired. In this section, we propose a guidance sampling framework for high-dimensional black-box optimization.

3.1 PROBLEM SETUP

The objective is to optimize a black-box function

$$\arg\max_{x \in \mathcal{X}} f^\star(x), \text{ where } f^\star : \mathcal{X} \to [0, 1]. \tag{12}$$

We consider a high-dimensional data space $\mathcal{X}$, such as the space of images. Moreover, we assume access to a large unlabeled dataset sampled from the data distribution q_{data}. Importantly, we do *not* assume access to an offline reward-labeled dataset. Rather, reward labels are actively acquired through an online interaction protocol (see the RHS of Figure 1). At each iteration k, the algorithm generates a sample x_k, queries its reward from the environment, and uses the newly acquired pair

Algorithm 2 Black-box **Re**ward-guided S**a**mpling of **D**iffusion Model (BREAD)

Require: Unlabeled dataset $\mathcal{D}$, reward-labeled dataset $\mathcal{D}^{\text{reward}} = \emptyset$, black-box objective $f^\star : \mathcal{X} \to [0,1]$, Gaussian process prior $\mathcal{GP}(0, \kappa)$, target reward $y^\star \in [0,1]$, UCB parameter $\lambda \in \mathbb{R}$, guidance strength $\eta : \mathbb{R}_+ \to \mathbb{R}_+$

1: **for** iteration $k = 1, 2, \cdots$ **do**
2: Fit the GP posterior $f^\star \mid \mathcal{D}^{\text{reward}} \sim \mathcal{GP}(\mu_k(\cdot), \kappa_k(\cdot,\cdot))$.
3: **if** Reward-directed Guidance (SUBROUTINE A) **then**
4: Set $f_k(x) = \log p^k_{\text{GP}}(y^\star|x)$.
5: **else if** Uncertainty-aware sampling (SUBROUTINE B) **then**
6: Set $f_k(x) = \mu_k(x) + \lambda\sigma_k(x)$, where μ_k and σ_k are given by Equations (9-10).
7: **end if**
8: Sample $x_k = \text{FORWARDUNIVERSALGUIDANCE}(f_k, \eta)$.
9: Observe reward $y_k \sim \mathcal{N}(f^\star(x_k), \tau^2)$ for the generated sample x_k.
10: Update the reward-labeled dataset $\mathcal{D}^{\text{reward}} = \mathcal{D}^{\text{reward}} \cup \{(x_k, y_k)\}$.
11: **end for**

(x_k, y_k) to update its reward estimator and subsequent sampling strategy. This setting differs from conventional Bayesian optimization (BO) problems (Frazier, 2018), which typically operate in low- to moderate-dimensional search spaces. Directly applying BO methods can be challenging, as modeling the black-box objective and optimizing the acquisition function become increasingly difficult as the dimensionality grows. Our setting also differs from existing diffusion model-based black-box optimization frameworks (Krishnamoorthy et al., 2023; Yuan et al., 2023; Li et al., 2024; Wu et al., 2024; Kong et al., 2025), which assume access to a pre-collected labeled dataset for offline training.

3.2 GUIDED SAMPLING OF DIFFUSION MODELS VIA BLACK-BOX REWARDS

In this section, we introduce the framework of **B**lack-box **Re**ward-guided S**a**mpling of **D**iffusion Model (BREAD, Algorithm 2). Our approach combines a diffusion model that is pre-trained on an unlabeled dataset with a Gaussian process (GP) that models the black-box objective. The GP helps discover high-reward regions of the data space, and the pre-trained diffusion model serves as prior distribution and generates valid and high-fidelity sample. At each iteration k, BREAD refits the GP posterior using the reward-labeled samples observed so far (Line 2), generates a new sample x_k via guided diffusion sampling (Lines 3-8), queries its reward $y_k \sim \mathcal{N}(f^\star(x_k), \tau^2)$ from the environment (Line 9), and augments the reward-labeled dataset with the newly observed sample-reward pair (Line 10).

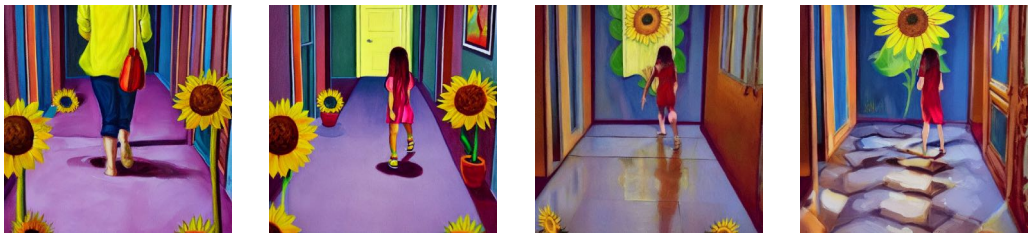
We consider two complementary strategies for steering generation toward high-reward samples. The first is motivated by reward-conditioned generation (Krishnamoorthy et al., 2023), which generates samples conditioned on a target reward value. During optimization, the conditioning value is iteratively updated to the best reward observed so far. The second is motivated by gradient-based guidance (Dhariwal & Nichol, 2021; Bansal et al., 2024), which uses gradients of a differentiable function as the guidance signal. Since the true gradients are unavailable in our black-box setting, we use the fitted GP as a differentiable surrogate to construct the guidance functions in both strategies. Accordingly, we introduce two subroutines that use forward universal guidance (Bansal et al., 2024, Algorithm 2) with different guidance functions derived from the GP.

SUB-ROUTINE A. Reward-conditioned guidance via GP-likelihood. This sub-routine aims to generate samples whose rewards are close to a pre-specified target value $y^\star$. Ideally, we would sample from the reward-conditioned target distribution

$$q_{y^\star}(x) \propto q_{\text{data}}(x)\mathcal{N}(y^\star; f^\star(x), \tau^2), \tag{13}$$

where q_{data} is the data distribution and τ^2 is the observation-noise variance. Given the GP fitted at iteration k, we approximate the target distribution by $q^k_{\text{GP},y^\star}(x) \propto q_{\text{data}}(x)p^k_{\text{GP}}(y^\star|x)$, where we denote by $p^k_{\text{GP}}(y^\star|x)$ the GP predictive likelihood of observing the target reward $y^\star$ for sample x. Specifically, let μ_k and σ^2_k be the mean (9) and the variance (10) of the conditional distribution of $f^\star$, we have $p^k_{\text{GP}}(y^\star|x) := \mathcal{N}(y^\star; \mu_k(x), \sigma^2_k(x) + \tau^2)$. To approximately sample from this distribution, we use the guided score

$$\tilde{s}^k_{\text{GP},T-t}(x^{\leftarrow}_t) = s^\theta_{T-t}(x^{\leftarrow}_t) + \eta(t)\nabla_{x^{\leftarrow}_t} \log p^k_{\text{GP}}\left(y^\star|\hat{x}^\theta_0(x^{\leftarrow}_t, T-t)\right), \tag{14}$$



(a) Unguided (ImageReward: 0.31)

(b) Forward universal guidance by exact gradients (ImageReward: 1.61)

(c) BReaD-SubA **(Round 21, ImageReward: 1.29)**

(d) BReaD-SubB **(Round 11, ImageReward: 1.75)**

Figure 2: **Human preference-guided generation (Task 2, Section 4.1).** *A painting of a girl walking in a hallway and suddenly finds a giant sunflower on the floor blocking her way*. BREAD generates high-score samples with only a few evaluations without accessing reward gradients. Images are generated with the same seed.

at each time t of the reverse process, where s^{θ}_{T-t} is the unconditional score estimated by the pre-trained diffusion model, $\hat{x}^{\theta}_0(x^{\leftarrow}_t, T-t)$ is the predicted posterior mean (5). We analyze the total-variation (TV) distance between the generated distribution and the target reward-conditioned distribution. Our analysis decomposes the distribution mismatch into five sources, including GP approximation, score estimation, discretization, posterior averaging, and initialization errors. The GP approximation error arises from the discrepancy between the GP surrogate and the true black-box objective, which decreases as the GP surrogate becomes more accurate with additional reward observations. The remaining errors arise from approximating the reward-conditioned sampling process using forward universal guidance. The complete bound is provided in Appendix B.3.

Theorem 1 (Distribution mismatch, simplified)**.** *Let $\hat{q}^k_{y^\star}$ denote the sample distribution generated by* SUBROUTINE A *of Algorithm 2 at iteration k. Under Assumptions 1-8 in Appendix B.2, the total-variation distance between $\hat{q}^k_{y^\star}$ and the target distribution (13) satisfies*

$$D_{\mathrm{TV}}(q_{y^\star}, \hat{q}^k_{y^\star}) \lesssim \varepsilon_{\mathrm{GP},k} + \varepsilon_{\mathrm{init}} + \varepsilon_{\mathrm{score}} + \varepsilon_{\mathrm{disc}} + \varepsilon_{\mathrm{post\text{-}avg}}. \tag{15}$$

Proof. See Appendix B. □

SUB-ROUTINE B. Uncertainty-aware guidance via UCB. To encourage exploration in the data space, this subroutine incorporates the Upper Confidence Bound (Auer et al., 2002, UCB) into guidance. We use the guided score

$$\tilde{s}^k_{\mathrm{UCB},T-t}(x^{\leftarrow}_t) = s^{\theta}_{T-t}(x^{\leftarrow}_t) + \eta(t)\nabla_{x^{\leftarrow}_t}\mathrm{UCB}_k(x^{\leftarrow}_t) \tag{16}$$

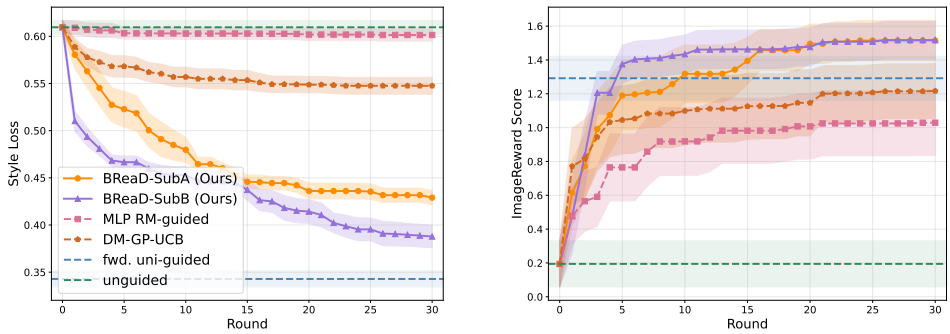
at each time t of the reverse process, where $\mathrm{UCB}_k(x^{\leftarrow}_t) := [\mu_k + \lambda\sigma_k]\,(\hat{x}^{\theta}_0(x^{\leftarrow}_t, T-t))$ is the (predicted) UCB value for the noisy sample, and λ controls the trade-off between exploitation and exploration. The guided process (16) aims to approximate the data distribution reweighted by the UCB value, i.e., $q^k_{\mathrm{UCB}}(x) \propto q_{\mathrm{data}}(x)\exp(\mathrm{UCB}_k(x))$. In the GP bandit literature (Srinivas et al., 2010), UCB provides an optimistic estimate of the black-box objective $f^\star$. By steering generation toward the high-UCB region of the data space and querying reward labels for these samples, the GP acquires informative observations that improve its estimation. This further enables the discovery of the promising high-reward region. In numerical experiments, we find that UCB guidance often leads to more effective optimization.

Extension to Deep Kernel Learning. To construct expressive kernel functions in high-dimensional spaces, we utilize deep kernel learning (Wilson et al., 2015, DKL), which uses a deep neural network to transform data into a low-dimensional representation, i.e., $g(\cdot; w) : \mathcal{X} \to \mathbb{R}^d$. We then define the GP kernel over the learned representations as

$$\kappa_{\boldsymbol{w}}(x, x') := \kappa(g(x; \boldsymbol{w}), g(x'; \boldsymbol{w})),$$

where κ is a base kernel function, e.g., the radial basis function (RBF) kernel. The weights can be learned by maximizing the marginal likelihood of the observed data

$$\max_{\boldsymbol{w}} \left\{ \log p(y_{1:N}|x_{1:N}; \boldsymbol{w}) \propto -y_{1:N}^{\top}\left(K^N_{\boldsymbol{w}} + \tau^2 I\right)^{-1} y_{1:N} - \log\det(K^N_{\boldsymbol{w}} + \tau^2 I) \right\},$$



(a) Style guidance (Task 1): Matching the text prompt, *A portrait of a woman*, guided by Pablo Picasso's painting, *Brick Factory at Tortosa*

(b) Human-preference guidance (Task 2): *A painting of a girl walking in a hallway and suddenly finds a giant sunflower on the floor blocking her way*

Figure 3: **Image generation guided by black-box rewards (Section 4.1).** Optimization for style loss (Left) and ImageReward score (Right). Results are averaged over 10 trials.

where $K_{\boldsymbol{w}} = [\kappa_{\boldsymbol{w}}(x_i, x_j)]$ is the kernel matrix.

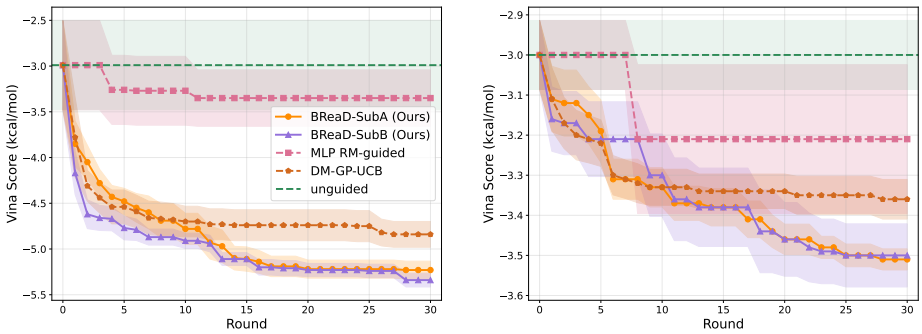
4 Numerical Results

Baselines. In this section, we report the performance of the proposed framework across image generation, protein design, and a simulated nuclear fusion control task. Particularly, we compare BReaD-SubA (BReaD with Subroutine A) and BReaD-SubB (BReaD with Subroutine B) against the following baselines: 1) MLP RM-guided, which guides the generation through a three-layer multi-layer perceptron reward model trained at each iteration to predict the black-box objective using all observations collected so far; 2) DM-GP-UCB, GP-UCB (Srinivas et al., 2010) using the diffusion model (DM) as a priori for searching; it generates 10 candidate samples from the DM at each iteration and selects the one with the highest UCB value for evaluation; 3) fwd. uni-guided, which applies Forward universal guidance (Bansal et al., 2024) using the exact reward gradients of the black-box objective; 4) unguided, which generates samples from the pre-trained diffusion model without reward guidance (while retaining prompt guidance in image generation)). We use the labels for 20 unconditionally generated samples to initialize the GP in BReaD and DM-GP-UCB methods and the reward model in the MLP RM-guided baseline. We report both the mean and the standard error. When reporting the results, we plot the curve for the best-so-far observed loss/reward for BReaD-SubA, BReaD-SubB, MLP-guided, GP-UCB, and random search. Since the unguided and fwd. uni-guided baselines do not adapt to new observations, we report the best loss/reward among 30 generated images.

Summary of results. We summarize the numerical results as follows. The proposed methods achieve performance comparable to the fwd. uni-guided baseline, despite not having access to the exact reward gradients. Meanwhile, they also consistently outperform the MLP-guided baseline. This demonstrates the benefit of GP-based surrogate modeling and guidance in low-data settings. Compared with GP-UCB, which follows a generate-then-select strategy over candidates sampled unconditionally from the diffusion model, our methods consistently generate higher-reward samples, especially when such samples may have low probability under the unconditional data distribution. Finally, incorporating UCB into the guidance process further promotes exploration of promising regions and often leads to more effective optimization performance. Additional experimental results and implementation details are postponed to Appendix E.

4.1 Results for Image Generation

Task 1. Style-guided generation. We perform style-guided image generation using Stable Diffusion v1.5 (Rombach et al., 2022). The *style loss* is defined as the cosine distance between the CLIP embeddings (Radford et al., 2021) of the generated and reference images, i.e., $\ell_{\text{style}}(x_{\text{gen}}) =$



(a) Dipeptidyl peptidase-4 (DPP4) (b) p38 MAP kinase (MAPK)

Figure 4: **Ligand optimization for Vina scores.** DPP4 and p38-MAPK. Results are averaged over 10 trials.

$1 - \cos\langle CLIP(x_{\text{ref}}), CLIP(x_{\text{gen}})\rangle$, where x_{ref} is the image in the reference style and x_{gen} is the generated image. Figure (3a) reports results for generating images that match the text prompt, *A portrait of a woman*, and the style of Pablo Picasso's painting, *Brick Factory at Tortosa*. Example generated images can be found in Figure 6 in the Appendix. Our methods can effectively produce images that match the reference style and achieve a loss comparable to the fwd. uni-guided baseline that has access to the exact loss gradient. Moreover, incorporating UCB into guidance leads to faster reduction in style loss.

Task 2. Human preference-guided generation. One practical low-data setting is personalized image generation, where a user may wish to generate images with desired properties based on only a few rounds of feedback. To simulate such feedback, we use ImageReward (Xu et al., 2023), a text-to-image human-preference reward model trained on 137k expert comparisons, as the black-box objective. We generate images with Stable Diffusion v1.5 and optimize the ImageReward score through iterative queries. Figures (3b) and 2 show one such example where BREAD effectively discovers images with substantially higher rewards and fidelity using only a small number of evaluations.

4.2 Results for Ligand Optimization

Structure-based drug design aims to identify ligand molecules that bind strongly to a target protein. We consider optimizing the Vina score (Trott & Olson, 2010), a widely used molecular docking score that approximates the binding affinity between a ligand and a protein binding pocket. A lower Vina score indicates favorable predicted binding. Directly applying gradient-based optimization methods can be challenging since the Vina score is non-differentiable. We use the pre-trained DiffSBDD model[1] (Schneuing et al., 2024), which generates three-dimensional ligand structures conditioned on a target protein pocket. Figure 4 reports results on DPP4 and p38-MAPK. We do not include the forward universal-guidance baseline because gradients of the Vina scoring function are unavailable. BREAD effectively improves the Vina score for both protein targets using only a small number of evaluations and outperforms the baselines.

4.3 Results for Controllable Nuclear Fusion

Nuclear fusion is a promising path toward sustainable energy. Our goal is to access the Wide-Pedestal QH mode (Li et al., 2022, WPQH), an advanced plasma scenario that achieves high energy confinement without damaging transient heat pulses. The labeled data for this plasma scenario is scarce. Among approximately 10k shots in the historical dataset of the DIII-D Tokamak in San Diego, we only have the WPQH labels for around 400 shots. We pre-trained a diffusion model on the historical shot data. Specifically, given the plasma state and actuator sequence during the first 200ms of the flat-top phase (10 control steps), the diffusion model models the plasma state and actuator

[1]Github repository: `https://github.com/arneschneuing/DiffSBDD`. We use the conditional full-atom model in the experiment.

sequence over the subsequent 3800ms (190 control steps). In the experiment, we parameterize six controllable actuators, including beam power, beam torque, I_p, $q95$, and ECH, using 21 parameters that determine their control waveforms.

Simulation results. We evaluate the methods in simulation. To simulate the tokamak dynamics, we train an ensemble of 25 recurrent probabilistic neural networks (RPNNs) on the historical DIII-D dataset, following (Char et al., 2024; Sonker et al., 2025). Given an actuator configuration, the RPNN ensemble predicts the resulting plasma trajectories. Additionally, we train a WPQH detector using labeled experimental data to identify whether the plasma is in WPQH mode at each timestep. The trained detector achieves 85.49% accuracy on real experimental trajectories and 72.46% accuracy when applied to trajectories rolled out by the RPNN ensemble. We then compute the WPQH ratio as the fraction of time steps during the flat-top phase classified as WPQH. We provide additional details in Appendix D. Figure 5 reports the best-so-far WPQH ratio over 30 optimization rounds. The results show that the proposed uncertainty-aware guidance method progressively improves the WPQH ratio and matches the performance of the forward universal guidance baseline that has direct access to the objective gradients.

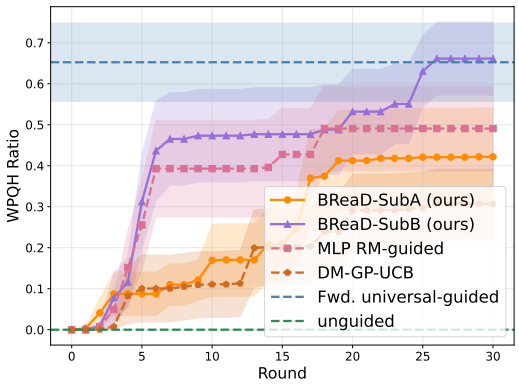


Figure 5: Accessing the Wide-Pedesatal QH in a simulated tokamak environment. Results are averaged over 10 trials.

Retrospective analysis on DIII-D Tokamak data. We further conduct a retrospective analysis using data collected from the experimental session MP 2026-72-01 in the DIII-D Tokamak, San Diego. Specifically, we replay BREAD using only experimental observations available up to a given shot and examine whether the generated candidate is consistent with actuator configurations that were subsequently validated in the experiment. After Shot 207515, BREAD proposes to increase the beam power from 2807 kW to 3855 kW and decrease I_p from 1.29 MA to 1.04 MA. Independently, the experiment subsequently executed Shot 207519 with a beam power of 4131 kW and I_p of 1.10 MA, which closely matches the direction and magnitude of the changes proposed by BREAD. Shot 207519 achieved a substantially higher WPQH ratio of 0.77, compared with 0.44 for Shot 207515.

5 Discussion

We propose BREAD, a framework for optimizing black-box reward functions in high-dimensional data spaces. BREAD uses a Gaussian process as a differentiable surrogate for the unknown reward function and constructs guidance signals for guided generation from a diffusion model that is pre-trained on unlabeled data. Numerical results on image generation, protein design, and simulated fusion plasma control tasks demonstrate that our framework can effectively optimize black-box rewards in low-data scenarios. An important direction for future work is to deploy and evaluate BREAD on real-world systems, such as the DIII-D tokamak. Another promising direction is to extend the framework to multi-objective and constrained optimization problems, which commonly arise in scientific applications.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experiments, assist with implementing methods, assist in checking mathematical arguments and proofs, and assist with interpreting experimental results. We also used generative AI tools for editing code, reviewing relevant literature, providing feedback on experimental design, and improving clarity and readability of the manuscript. All AI-assisted mathematical arguments were independently checked by the authors, and AI-assisted code was verified and tested for correctness. We reviewed all AI-assisted content and take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REFERENCES

Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. ISSN 0304-4149. doi: https://doi.org/10.1016/0304-4149(82)90051-5. URL https://www.sciencedirect.com/science/article/pii/0304414982900515.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2–3):235–256, May 2002. ISSN 0885-6125. doi: 10.1023/A:1013689704352. URL https://doi.org/10.1023/A:1013689704352.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Roni Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pzpWBbnwiJ.

Felix Berkenkamp, Andreas Krause, and Angela P. Schoellig. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, 112(10):3713–3747, 2023.

Mickaël Binois and Nathan Wycoff. A survey on high-dimensional gaussian process modeling with application to bayesian optimization. *ACM Trans. Evol. Learn. Optim.*, 2(2), August 2022. ISSN 2688-299X. doi: 10.1145/3545611. URL https://doi.org/10.1145/3545611.

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

Ian Char, Youngseog Chung, Joseph Abbate, Egemen Kolemen, and Jeff Schneider. Full shot predictions for the diii-d tokamak via deep recurrent networks, 2024. URL https://arxiv.org/abs/2404.12416.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zyLVMgsZ0U_.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

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

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

David Eriksson and Martin Jankowiak. High-dimensional bayesian optimization with sparse axis-aligned subspaces. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 493–503. PMLR, 2021.

David Eriksson, Michael Pearce, Jacob R. Gardner, Ryan Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=FPnUhsQJ5B.

Peter I. Frazier. A tutorial on bayesian optimization, 2018. URL https://arxiv.org/abs/1807.02811.

Nate Gruver, Samuel Stanton, Nathan Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 12489–12517. Curran Associates, Inc., 2023. doi: 10.52202/075280-0547. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/29591f355702c3f4436991335784b503-Paper-Conference.pdf.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Danial Khatamsaz, Raymond Neuberger, Arunabha M. Roy, Sina Hossein Zadeh, Richard Otis, and Raymundo Arróyave. A physics informed bayesian optimization approach for material design: application to niti shape memory alloys. *npj Computational Materials*, 9(1):221, 2023.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Lingkai Kong, Yuanqi Du, Wenhao Mu, Kirill Neklyudov, Valentin De Bortoli, Dongxia Wu, Haorui Wang, Aaron M Ferber, Yian Ma, Carla P Gomes, and Chao Zhang. Diffusion models as constrained samplers for optimization with unknown constraints. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (eds.), *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4582–4590. PMLR, 03–05 May 2025. URL https://proceedings.mlr.press/v258/kong25b.html.

Siddarth Krishnamoorthy, Satvik Mehul Mashkaria, and Aditya Grover. Diffusion models for black-box optimization. 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. 17842–17857. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/krishnamoorthy23a.html.

Aviral Kumar and Sergey Levine. Model inversion networks for model-based optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5126–5137. Curran Associates, Inc.,

2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/373e4c5d8edfa8b74fd4b6791d0cf6dc-Paper.pdf.

Zeyu Li, Xi Chen, C. M. Muscatello, K.H. Burrell, Xueqiao Xu, Ben Zhu, Rongjie Hong, T.H. Osborne, B. A. Grierson, T.L. Rhodes, T.Y. Xia, G.R. McKee, Z. Yan, Guanying Yu, and DIII-D Team. Numerical modeling of pedestal stability and broadband turbulence of wide-pedestal qh-mode plasmas on diii-d. *Nuclear Fusion*, 62(7):076033, may 2022. doi: 10.1088/1741-4326/ac4acf. URL https://doi.org/10.1088/1741-4326/ac4acf.

Zihao Li, Hui Yuan, Kaixuan Huang, Chengzhuo Ni, Yinyu Ye, Minshuo Chen, and Mengdi Wang. Diffusion model for data-driven black-box optimization, 2024. URL https://arxiv.org/abs/2403.13219.

Mohit Malu, Gautam Dasarathy, and Andreas Spanias. Bayesian optimization in high-dimensional spaces: A brief survey. In *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pp. 1–8, 2021. doi: 10.1109/IISA52424.2021.9555522.

Satvik Mehul Mashkaria, Siddarth Krishnamoorthy, and Aditya Grover. Generative pretraining for black-box optimization. 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. 24173–24197. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/mashkaria23a.html.

Leonard Papenmeier, Luigi Nardi, and Matthias Poloczek. Increasing the scope as you learn: Adaptive bayesian optimization in nested subspaces. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=e4Wf6112DI.

E. O. Pyzer-Knapp. Bayesian optimization for accelerated drug discovery. *IBM Journal of Research and Development*, 62(6):2:1–2:7, 2018. doi: 10.1147/JRD.2018.2881731.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Rajalaxmi Rajagopalan, Debottam Dutta, Yu-Lin Wei, and Romit Roy Choudhury. Personalized image generation via human-in-the-loop bayesian optimization. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=sq3JUR4IDd.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01042. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01042.

Jeff Schneider. Bayesian optimization and embedded learning systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 413, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2945367. URL https://doi.org/10.1145/2939672.2945367.

Arne Schneuing, Charles Harris, Yuanqi Du, Kieran Didi, Arian Jamasb, Ilia Igashov, Weitao Du, Carla Gomes, Tom L. Blundell, Pietro Lio, Max Welling, Michael Bronstein, and Bruno Correia. Structure-based drug design with equivariant diffusion models. *Nature Computational Science*, 4 (12):899–909, 2024.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2016. doi: 10.1109/JPROC.2015.2494218.

Jasper Snoek, Hugo Larochelle, and Ryan Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C.J. Burges, L. Bottou, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Rohit Sonker, Alexandre Capone, Andrew Rothstein, Hiro Josep Farre Kaga, Egemen Kolemen, and Jeff Schneider. Multi-timescale dynamics model Bayesian optimization for plasma stabilization in tokamaks. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56465–56479. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/sonker25a.html.

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.

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 Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20459–20478. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/stanton22a.html.

Kim Yong Tan, Yueming Lyu, Ivor W. Tsang, and Yew-Soon Ong. Fast direct: Query-efficient online black-box guidance for diffusion-model target generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Matthew Tesch, Jeff Schneider, and Howie Choset. Expensive multiobjective optimization for robotics. In *2013 IEEE International Conference on Robotics and Automation*, pp. 973–980, 2013. doi: 10.1109/ICRA.2013.6630691.

Oleg Trott and Arthur J. Olson. Autodock vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2010. doi: https://doi.org/10.1002/jcc.21334. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.21334.

Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, and Nando de Freitas. Bayesian optimization in high dimensions via random embeddings. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pp. 1778–1784. AAAI Press, 2013.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning, 2015. URL https://arxiv.org/abs/1511.02222.

Dongxia Wu, Nikki Lijing Kuang, Ruijia Niu, Yian Ma, and Rose Yu. Diff-BBO: Diffusion-based inverse modeling for black-box optimization. In *NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty*, 2024. URL https://openreview.net/forum?id=KizOx8SheQ.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=JVzeOYEx6d.

Hui Yuan, Kaixuan Huang, Chengzhuo Ni, Minshuo Chen, and Mengdi Wang. Reward-directed conditional diffusion: Provable distribution estimation and reward improvement. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=58HwnnEdtF.

Taeyoung Yun, Kiyoung Om, Jaewoo Lee, Sujin Yun, and Jinkyoo Park. Posterior inference with diffusion models for high-dimensional black-box optimization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EXds2NBOoq.

A Related Work

High-dimensional Bayesian optimization. Bayesian optimization (BO) is effective for optimizing expensive black-box objectives but often struggles in high-dimensional spaces. REMBO (Wang et al., 2013) assumes that the objective has low effective dimensionality and performs BO in a randomly embedded low-dimensional subspace. TuRBO (Eriksson et al., 2019) performs BO within local trust regions and adaptively expands or contracts these regions during optimization. SAASBO (Eriksson & Jankowiak, 2021) places a sparsity-inducing prior over input dimensions. BAxUS (Papenmeier et al., 2022) progressively scales up the dimensionality of the target subspace.

Diffusion models and controlled generation. A separate line of work uses pre-trained diffusion models to generate samples with desired properties. Classifier guidance (Dhariwal & Nichol, 2021) modifies the diffusion score using gradients from an auxiliary classifier, while classifier-free guidance (Ho & Salimans, 2021) performs conditional generation by jointly learning the estimation of unconditional and conditional scores. Diffusion Posterior Sampling (Chung et al., 2023) extends gradient-guided diffusion to inverse problems by incorporating gradients derived from an observation likelihood. Universal Guidance enables a pre-trained diffusion model to be controlled by arbitrary differentiable guidance functions without retraining task-specific components (Bansal et al., 2024). These approaches assume access to a differentiable function that provides guidance gradients during sampling, which are not readily available in black-box settings.

Generative black-box optimization. Generative models have increasingly been used to optimize black-box objectives in high-dimensional design spaces. Brookes et al. (2019) propose adaptive sampling of generative models that estimates a distribution over the design space conditioned on desired properties. Model inversion networks (Kumar & Levine, 2020, MINs) learn an inverse function that maps values to corresponding inputs. DDOM (Krishnamoorthy et al., 2023) trains a reward-conditioned diffusion model from a large labeled dataset and generates candidates conditioned on a high objective value. Diff-BBO (Wu et al., 2024) further extends this approach to an online setting and uses an uncertainty-aware acquisition strategy to propose the conditioned objective. Reward-directed generation (Yuan et al., 2023) uses a pseudo-labeler trained on labeled data to annotate an unlabeled dataset and then trains a reward-conditioned diffusion model. Recent work integrates Bayesian optimization into generative model-based optimization. GenBO (Mashkaria et al., 2023) approximates a target distribution proportional to BO's acquisition function to propose the next query locations. BONET (Mashkaria et al., 2023) approximates the dynamics of black-box optimizers using an autoregressive sequence model. LaMBO (Stanton et al., 2022) and LaMBO-2 (Gruver et al., 2023) jointly train a diffusion model with a discriminator and perform gradient-based optimization of an acquisition function. Fast Direct (Tan et al., 2025) updates a GP surrogate using batched black-box evaluations and constructs a pseudo-target for guidance. DAS (Kim et al., 2025) performs Sequential Monte Carlo (SMC) during the reverse diffusion process to approximate a reward-tilted distribution at test time. DiBO (Yun et al., 2025) iteratively retrains and fine-tunes a diffusion model to sample from the posterior distribution induced by a deep-ensemble reward surrogate to balance exploration and exploitation. Multi-BO (Rajagopalan et al., 2026) performs BO on the diffusion model's attention features and iteratively uses human feedback to generate new candidates.

B Distibution Mismatch Analysis of Subroutine A

B.1 Notations and Definitions

Notation	Definition
$f^\star$	(Unknown) Reward function
$\mathcal{N}(0,\tau)$	Noise in the observed rewards
$y^\star$	Target score
q_{data}	Data distribution
q_t	Marginal distribution of $\bar{X}_t$ in process (17)

Continued on next page

Notation	Definition
$Q_T^{\leftarrow}$	Path distribution of the backward SDE (18)
$m_t(x_t)$	Posterior mean $\mathbb{E}_{q_0(\cdot\|x_t)}[x_0]$ (3)
$J_m(x_t, t)$	Jacobian matrix $\nabla_{x_t} m_t(x_t)$ of the posterior-mean map
$q_{y^\star}(x) \propto q_{\text{data}}(x)\mathcal{N}(y^\star; f^\star(x), \tau^2)$	Target-conditioned distribution
$Z_{y^\star}$	Normalizer $\int \mathcal{N}(y^\star; f^\star(x_0), \tau^2) q_{\text{data}}(x_0)\mathrm{d}x_0$
$q_{y^\star,t}$	Marginal distribution of $\bar{X}_{y^\star,t}$ in process (19)
$Q^{\leftarrow}_{y^\star,T}$	Path distribution of the backward SDE (20)
$\bar{p}_t(y^\star\|x_t)$	$\mathbb{E}_{x_0\sim q_0(\cdot\|x_t)}[\mathcal{N}(y^\star; f^\star(x_0), \tau^2)]$
$q^k_{\text{GP},y^\star}(x) \propto q_{\text{data}}(x)\mathcal{N}(y^\star; \mu_k(x), \sigma_k^2(x) + \tau^2)$	GP-conditioned distribution at the k-th iteration
Z_k	Normalizer $\int q_{\text{data}}(x)\mathcal{N}(y^\star; \mu_k(x), \sigma_k^2(x) + \tau^2)\mathrm{d}x$
$q^k_{\text{GP},y^\star,t}$	Marginal distribution of $\bar{X}^k_t$ in process (22)
$Q^{k,\leftarrow}_{\text{GP},y^\star,T}$	Path distribution of the backward SDE (23)
$\bar{p}^k_{\text{GP},t}(y^\star\|x_t)$	$\mathbb{E}_{x_0\sim q_0(\cdot\|x_t)}[\mathcal{N}(y^\star; \hat{f}^k(x_0), \tau^2)]$
$\hat{q}^k_{y^\star}$	Generated distribution at the k-th online iteration
$\hat{f}^k$	GP posterior mean of $f^\star$ fitted at the k-th online iteration
$\hat{x}_0^\theta(x_t, t)$	Predicted posterior mean (5)
$J_{\hat{x}_0^\theta}(x_t, t)$	Jacobian matrix $\nabla_{x_t}\hat{x}_0^\theta(x_t, t)$ of the posterior-mean prediction map
$\epsilon^2_{\text{uncond}}$	Bound on the per-step score estimation error (24)
$\bar{\epsilon}^2_{\text{uncond}}$	Bound on the weighted-average score estimation error (39)
$\bar{\epsilon}^2_{\text{Hess}}$	Bound on the weighted-average score Hessian estimation error (40)
$\epsilon^2_{\text{post-avg}}$	Posterior-averaging error (53)
L_τ	(29)
$L_{\tau,\text{Var}}$	(30)
h	Discretization length
$\mathfrak{m}_2^2$	Assumption 6
G_k	Assumption 8
$\hat{J}_{\max}$	(38)
$J_{\max}$	(41)
$B^\star$	Assumption 1
r_k	GP mean-approximation error (27)
$r_{\nabla,k}$	GP gradient-approximation error (42)
v_k^2	GP variance (28)
$v_{\nabla,k}$	Gradient of the GP variance (43)
M_k	(46)
B_k	(45)
c_τ	(31)
$c_{\tau,k}$	(47)
L_k	Assumption 5
$\tilde{L}_k$	Assumption 5

Table 1: Notation and definitions in the proof

B.2 PRELIMINARIES

Unconditional generation processes In the proof, we consider $g(t) \equiv 1$ in the forward SDE (1), that is,

$$\mathrm{d}\bar{X}_t = -\bar{X}_t \mathrm{d}t + \sqrt{2}\mathrm{d}B_t, X_0 \sim q_{\text{data}}, \tag{17}$$

and we have $q_t(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, 1 - \bar{\alpha}_t I)$, where $\bar{\alpha}_t = \exp(-2t)$. The corresponding backward SDE is given by

$$\mathrm{d}\bar{X}_t^{\leftarrow} = \left\{\bar{X}_t^{\leftarrow} + 2\nabla \log q_{T-t}(\bar{X}_t^{\leftarrow})\right\} \mathrm{d}t + \sqrt{2}\mathrm{d}B_t, \bar{X}_0^{\leftarrow} \sim q_T. \tag{18}$$

Sampling from the target distribution Next, we consider the processes to sample from the target distribution, i.e., $q_{y^\star}(x) \propto q_{\text{data}}(x)\mathcal{N}(y^\star; f^\star(x), \tau^2)$. The forward process is given by

$$\mathrm{d}\bar{X}_{y^\star,t} = -\bar{X}_{y^\star,t}\mathrm{d}t + \sqrt{2}\mathrm{d}B_t, X_0 \sim q_{y^\star}. \tag{19}$$

The corresponding backward SDE is given by

$$\mathrm{d}\bar{X}_{y^\star,t}^{\leftarrow} = \left\{\bar{X}_{y^\star,t}^{\leftarrow} + 2\nabla \log q_{y^\star,T-t}(\bar{X}_{y^\star,t}^{\leftarrow})\right\} \mathrm{d}t + \sqrt{2}\mathrm{d}B_t, \bar{X}_{y^\star,0}^{\leftarrow} \sim q_{y^\star,T}. \tag{20}$$

Specifically, we have

$$\nabla \log q_{y^\star,T-t}(\bar{X}_{y^\star,t}^{\leftarrow}) = \nabla \left[\log q_{T-t}(\bar{X}_{y^\star,t}^{\leftarrow}) + \log \bar{p}_{T-t}(y^\star|\bar{X}_{y^\star,t}^{\leftarrow})\right],$$

where $\bar{p}_{T-t}(y^\star|\bar{X}_{y^\star,t}^{\leftarrow}) = \mathbb{E}_{x_0\sim q_0(\cdot|x_t)}[\mathcal{N}(y^\star; f^\star(x_0), \tau^2)]$. To show this, note that

$$q_{y^\star}(x_0) = \frac{q_{\text{data}}(x_0)\mathcal{N}(y^\star; f^\star(x_0), \tau^2)}{Z_{y^\star}}, \tag{21}$$

where $Z_{y^\star} = \int \mathcal{N}(y^\star; f^\star(x_0), \tau^2) q_{\text{data}}(x_0)\mathrm{d}x_0$ is the normalizer. Therefore, we have

$$\begin{aligned}
q_{y^\star,t}(x_t) &= \int q_t(x_t|x_0) q_{y^\star}(x_0)\mathrm{d}x_0 = \frac{1}{Z_{y^\star}} \int q_t(x_t|x_0) q_{\text{data}}(x_0)\mathcal{N}(y^\star; f^\star(x_0), \tau^2)\mathrm{d}x_0 \\
&= \frac{q_t(x_t)}{Z_{y^\star}} \int \frac{q_t(x_t|x_0) q_{\text{data}}(x_0)}{q_t(x_t)} \mathcal{N}(y^\star; f^\star(x_0), \tau^2)\mathrm{d}x_0 \\
&= \frac{q_t(x_t)}{Z_{y^\star}} \mathbb{E}_{x_0\sim q_0(\cdot|x_t)}[\mathcal{N}(y^\star; f^\star(x_0), \tau^2)].
\end{aligned}$$

Taking the logarithm and the derivative on both sides leads to the statement.

(Ideal) Sample distribution Since we model the unknown reward function with a GP, we consider the processes to sample from $q^k_{\text{GP},y^\star}(x) \propto q_{\text{data}}(x)\mathcal{N}(y^\star; \mu_k(x), \sigma_k^2(x) + \tau^2)$.

$$\mathrm{d}\bar{X}_t^k = -\bar{X}_t^k \mathrm{d}t + \sqrt{2}\mathrm{d}B_t, X_0^k \sim q^k_{\text{GP},y^\star}. \tag{22}$$

The corresponding backward SDE transforms noise into samples from $q^k_{\text{GP},y^\star}$

$$\mathrm{d}\bar{X}_t^{\leftarrow,k} = \left\{\bar{X}_t^{\leftarrow,k} + 2\nabla \log q^k_{\text{GP},y^\star,T-t}(\bar{X}_t^{\leftarrow,k})\right\} \mathrm{d}t + \sqrt{2}\mathrm{d}B_t, \bar{X}_0^{\leftarrow,k} \sim q^k_{\text{GP},y^\star,T}. \tag{23}$$

Specifically, we have

$$\nabla \log q^k_{\text{GP},y^\star,T-t}(\bar{X}_t^{\leftarrow,k}) = \nabla \left[\log q_{T-t}(\bar{X}_t^{\leftarrow,k}) + \log \bar{p}^k_{\text{GP},T-t}(y^\star|\bar{X}_t^{\leftarrow,k})\right],$$

where $\bar{p}^k_{\text{GP},T-t}(y^\star|\bar{X}_t^{\leftarrow,k}) = \mathbb{E}_{x_0\sim q_0(\cdot|x_t)}[\mathcal{N}(y^\star; \mu_k(x_0), \sigma_k^2(x_0) + \tau^2)]$.

Our proof is based on the following assumptions.

Assumption 1 (Bounded reward)**.** We have $f^\star(x) \in [0, 1]$ and $\|\nabla f^\star(x)\| \leq B^\star$ for any $x \in \mathcal{X}$.

Assumption 2. The true reward $f^\star : \mathcal{X} \to [0, 1]$ belongs to the RKHS $\mathcal{H}_\kappa$ associated with the kernel $\kappa(\boldsymbol{x}, \boldsymbol{x}')$ and satisfies $\|f^\star\|_{\mathcal{H}_\kappa} \leq B$. At online iteration k, the observed reward is $y^k = f^\star(x^k) + \varepsilon^k$, where the noise ε^k is conditionally zero-mean given the history and is R-sub-Gaussian.

Assumption 3 (Unconditional score estimation error)**.** We have

$$\mathbb{E}_{x_t\sim q_t}\left[\left\|s_t^\theta(x_t)-\nabla\log q_t(x_t)\right\|^2\right]\le\epsilon_{\text{uncond}}^2 \tag{24}$$

Assumption 4 (Bounded kernel)**.** We have $\sup_{x\in\mathcal{X}}\kappa(x,x)\le\kappa_{\max}$.

Assumption 5 (Adapt from (Chen et al., 2023, A1))**.** The score function $\nabla\log q^k_{\text{GP},y^\star,t}$ of the forward process (22) is L_k-Lipschitz, i.e., $\|\nabla\log q^k_{\text{GP},y^\star,t}(x)-\nabla\log q^k_{\text{GP},y^\star,t}(x')\|\le L_k\|x-x'\|$.

Assumption 6 (Bounded second moment (Chen et al., 2023, A2))**.** The second moment of the data distribution q_{data} is finite, i.e., $\mathfrak{m}_2^2:=\mathbb{E}_{q_{\text{data}}}[\|\cdot\|^2]<\infty$.

Assumption 7 ((Chen et al., 2023, A3))**.** The data distribution q_{data} has finite KL divergence w.r.t. the standard Gaussian.

Assumption 8. $\nabla\log p^k_{\text{GP}}(y^\star|x)$ is bounded and $\tilde{L}_k$-Lipschitz, i.e., $\|\nabla\log p^k_{\text{GP}}(y^\star|x)\|\le G_k$ and $\|\nabla\log p^k_{\text{GP}}(y^\star|x)-\nabla\log p^k_{\text{GP}}(y^\star|x')\|\le\tilde{L}_k\|x-x'\|$ for any $x,x'\in\mathcal{X}$.

B.3 Complete Distribution Mismatch Bound

We state the complete bound in Theorem 1.

$$\begin{aligned}
&D_{\text{TV}}(q_{y^\star},\hat{q}^k_{y^\star})\\
\lesssim&\underbrace{\frac{L_\tau r_k+L_{\tau,\text{Var}}v_k^2}{c_\tau}\left(1+\sqrt{B_k+\frac{M_k^2}{c_{\tau,k}^2}T}\right)+\frac{J_{\max}}{\tau^2}\left[(1+r_k)r_{\nabla,k}+\left(r_k+\frac{v_k^2}{\tau^2}\right)B^\star+\frac{v_{\nabla,k}}{2}\left(\frac{(1+r_k)^2}{\tau^2}+1\right)\right]\sqrt{T}}_{\varepsilon_{\text{GP},k}\text{ (GP approximation error)}}\\
&+\underbrace{\left(1+\frac{v_k^2}{\tau^2}\right)^{1/4}\exp\left(\frac{(1+r_k)^2+2}{4\tau^2}\right)\sqrt{\epsilon^2_{\text{uncond}}T+\hat{J}^2_{\max}\tilde{L}_k^2\bar{\epsilon}^2_{\text{uncond}}+G_k^2\bar{\epsilon}^2_{\text{Hess}}}}_{\varepsilon_{\text{score}}\text{ (score estimation error)}}\\
&+\underbrace{L_k\sqrt{dh}+\left(1+\frac{v_k^2}{\tau^2}\right)^{\frac{1}{4}}\exp\left(\frac{(1+r_k)^2}{4\tau^2}\right)L_kh\mathfrak{m}_2}_{\varepsilon_{\text{disc}}\text{ (discretization error)}}\\
&+\underbrace{\left(1+\frac{v_k^2}{\tau^2}\right)^{\frac{1}{4}}\exp\left(\frac{(1+r_k)^2+1}{4\tau^2}\right)\epsilon_{\text{post-avg}}}_{\varepsilon_{\text{post-avg}}\text{ (posterior-averaging error)}}\\
&+\underbrace{\sqrt{D_{\text{KL}}(q^k_{\text{GP},y^\star}\|\varepsilon)}\exp(-T)}_{\varepsilon_{\text{init}}\text{ (Initialization error)}},
\end{aligned} \tag{25}$$

B.4 Proof of Theorem 1

To this end, we first decompose the suboptimality of $\hat{q}^k_{y^\star}$, i.e., the generated distribution at the k-th iteration. Note that

$$D_{\text{TV}}(q_{y^\star},\hat{q}^k_{y^\star})\le\underbrace{D_{\text{TV}}(q_{y^\star},q^k_{\text{GP},y^\star})}_{\mathcal{E}_1\text{ (GP approximation error)}}+\underbrace{D_{\text{TV}}(q^k_{\text{GP},y^\star},\hat{q}^k_{y^\star})}_{\mathcal{E}_2\text{ (Sampling error)}}, \tag{26}$$

where $q^k_{\text{GP},y^\star}(x)\propto q_{\text{data}}(x)\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)$ is the target distribution induced by the GP.

B.4.1 Bouding GP Approximation Error $\mathcal{E}_1$

Assume

$$\sup_x|\mu_k(x)-f^\star(x)|\le r_k, \tag{27}$$

$$\sup_x \sigma_k^2(x) \le v_k^2. \tag{28}$$

We show that

$$\mathcal{E}_1 = D_{\mathrm{TV}}(q_{y^\star}, q^k_{\mathrm{GP},y^\star}) \le \frac{L_\tau r_k + L_{\tau,\mathrm{Var}} v_k^2}{c_\tau}.$$

Note that by definition of total variation distance, we have

$$\begin{aligned}
D_{\mathrm{TV}}(q_{y^\star}, q^k_{\mathrm{GP},y^\star}) =& \frac{1}{2}\int q_{\text{data}}(x)\left|\frac{\mathcal{N}(y^\star; f^\star(x),\tau^2)}{Z_{y^\star}} - \frac{\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)}{Z_k}\right|\mathrm{d}x \\
\le& \frac{1}{2}\Bigg(\int q_{\text{data}}(x)\left|\frac{\mathcal{N}(y^\star; f^\star(x),\tau^2) - \mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)}{Z_{y^\star}}\right|\mathrm{d}x \\
&+ \left|\frac{1}{Z_{y^\star}} - \frac{1}{Z_k}\right| \underbrace{\int q_{\text{data}}(x)\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)\mathrm{d}x}_{=Z_k}\Bigg) \\
\le& \frac{1}{2}\frac{2}{Z_{y^\star}}\int q_{\text{data}}(x)\left|\mathcal{N}(y^\star; f^\star(x),\tau^2) - \mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)\right|\mathrm{d}x \\
\le& \frac{1}{Z_{y^\star}}\mathbb{E}_{q_{\text{data}}}\left[\left|\mathcal{N}(y^\star; f^\star(x),\tau^2) - \mathcal{N}(y^\star;\mu_k(x),\tau^2)\right|\right] \\
&+ \frac{1}{Z_{y^\star}}\mathbb{E}_{q_{\text{data}}}\left[\left|\mathcal{N}(y^\star;\mu_k(x),\tau^2) - \mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)\right|\right],
\end{aligned}$$

where the second inequality uses the fact that $|Z_k - Z_{y^\star}| \le \mathbb{E}_{q_{\text{data}}}[|\mathcal{N}(y^\star; f^\star(x),\tau^2) - \mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)|]$. By the Lipschitz property of the Gaussian likelihood in its mean, there exists a constant

$$L_\tau := \frac{1}{\sqrt{2\pi e}\tau^2} \ge \sup_{\mu\in\mathbb{R}}\left|\frac{\partial}{\partial\mu}\mathcal{N}(y^\star;\mu,\tau^2)\right|, \tag{29}$$

such that $|\mathcal{N}(y^\star; f^\star(x),\tau^2) - \mathcal{N}(y^\star;\mu_k(x),\tau^2)| \le L_\tau|\mu_k(x) - f^\star(x)| \le L_\tau r_k$. On the other hand, since

$$\begin{aligned}
&\sup_{\mu\in\mathbb{R},v\ge\tau^2}\left|\frac{\partial}{\partial v}\mathcal{N}(y^\star;\mu,v)\right| \\
=& \sup_{\mu\in\mathbb{R},v\ge\tau^2}\left|\frac{1}{2\sqrt{2\pi}v^{3/2}}\exp\left(-\frac{(y^\star-\mu)^2}{2v}\right)\left(\frac{(y^\star-\mu)^2}{v} - 1\right)\right| \\
\le& \frac{1}{2\sqrt{2\pi}\tau^3} =: L_{\tau,\mathrm{Var}},
\end{aligned} \tag{30}$$

where the last inequality uses the fact that $e^{-z/2}|z-1| \le 1$ for $z \ge 0$. Hence, we have $|\mathcal{N}(y^\star;\mu_k(x),\tau^2) - \mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)| \le L_{\tau,\mathrm{Var}} v_k^2$. Furthermore, since $y^\star, f^\star(x) \in [0,1]$ by Assumption 1, we have

$$Z_{y^\star} = \int \mathcal{N}(y^\star; f^\star(x),\tau^2) q_{\text{data}}(x)\mathrm{d}x \ge \frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{1}{2\tau^2}\right) =: c_\tau, \tag{31}$$

which concludes the proof.

B.4.2 Bounding Sampling Error $\mathcal{E}_2$

First, note that

$$\begin{aligned}
\mathcal{E}_2 = D_{\mathrm{TV}}(q^k_{\mathrm{GP},y^\star}, \hat{q}^k_{y^\star}) \le& D_{\mathrm{TV}}(q^k_{\mathrm{GP},y^\star}, \tilde{q}^k_{y^\star}) + D_{\mathrm{TV}}(\tilde{q}^k_{y^\star}, \hat{q}^k_{y^\star}) \\
\le& D_{\mathrm{TV}}(q^k_{\mathrm{GP},y^\star}, \tilde{q}^k_{y^\star}) + D_{\mathrm{TV}}(q^k_{\mathrm{GP},y^\star,T}, \varepsilon),
\end{aligned}$$

where $\tilde{q}^k_{y^\star}$ is the (discretized) reverse sampling process (23), and the last inequality holds by the data processing inequality. For the second term, we have

$$D_{\mathrm{TV}}(q^k_{\mathrm{GP},y^\star,T}, \varepsilon) \lesssim \sqrt{D_{\mathrm{KL}}(q^k_{\mathrm{GP},y^\star}\|\varepsilon)}\exp(-T)$$

Next, we bound the total variation distance between $q^k_{\mathrm{GP},y^\star}$ and $\tilde{q}^k_{y^\star}$, i.e., $D_{\mathrm{TV}}(q^k_{\mathrm{GP},y^\star}, \tilde{q}^k_{y^\star})$. Let $Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}$ denote the law of the ideal reverse path (23), which produces $q^k_{\mathrm{GP},y^\star}$. Define $0 = t_0 < t_1 < \cdots < t_J = T$ as the time grid for the discretized backward SDE, where $h = \frac{T}{J} \lesssim L_k^{-1}$ and $t_j = jh$. We denote by $\tilde{Q}^{k,\leftarrow}_{\mathrm{GP},y^\star,T}$ the path law of the corresponding discretized backward SDE that produces $\tilde{q}^k_{y^\star}$, which uses $\tilde{s}_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})$ for $t \in [t_{j-1}, t_j]$. Therefore, by the data processing inequality for total variation and Pinsker's inequality, we derive

$$D_{\mathrm{TV}}(q^k_{\mathrm{GP},y^\star}, \tilde{q}^k_{y^\star}) \le D_{\mathrm{TV}}(Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}, \tilde{Q}^{k,\leftarrow}_{\mathrm{GP},y^\star,T}) \le \sqrt{\frac{1}{2} D_{\mathrm{KL}}(Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T} \| \tilde{Q}^{k,\leftarrow}_{\mathrm{GP},y^\star,T})}.$$

To this end, we first show

$$\begin{aligned}
&\sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \int_{t_{j-1}}^{t_j} \left\| \tilde{s}_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q^k_{\mathrm{GP},y^\star,T-t}(X^{\leftarrow}_t) \right\|^2 \mathrm{d}t \\
\lesssim & L_k^2 dh + \sqrt{1 + \frac{v_k^2}{\tau^2}} \exp\left(\frac{(1+r_k)^2}{2\tau^2} \right) L_k^2 h^2 \mathfrak{m}_2^2 \\
&+ \sqrt{1 + \frac{v_k^2}{\tau^2}} \exp\left(\frac{(1+r_k)^2 + 2}{2\tau^2} \right) \left(\epsilon^2_{\text{uncond}} T + \hat{J}^2_{\max} \tilde{L}_k^2 \bar{\epsilon}^2_{\text{uncond}} + G_k^2 \bar{\epsilon}^2_{\text{Hess}} \right) \\
&+ J^2_{\max} \left[\frac{(1+r_k) r_{\nabla,k}}{\tau^2} + \left(\frac{r_k}{\tau^2} + \frac{v_k^2}{\tau^4} \right) B^\star + \frac{1}{2} \left(\frac{(1+r_k)^2}{\tau^4} + \frac{1}{\tau^2} \right) v_{\nabla,k} \right]^2 T \\
&+ \frac{(L_\tau r_k + L_{\tau,\mathrm{Var}} v_k^2)^2}{c_\tau^2} \left(B_k + \frac{M_k^2}{c_{\tau,k}^2} T \right) + \sqrt{1 + \frac{v_k^2}{\tau^2}} \exp\left(\frac{(1+r_k)^2 + 1}{2\tau^2} \right) \epsilon^2_{\text{post-avg}}.
\end{aligned} \tag{32}$$

Note that

$$\begin{aligned}
&\sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \int_{t_{j-1}}^{t_j} \left[\left\| \tilde{s}_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q^k_{\mathrm{GP},y^\star,T-t}(X^{\leftarrow}_t) \right\|^2 \right] \mathrm{d}t \\
\lesssim & \sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| \tilde{s}_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q^k_{\mathrm{GP},y^\star,T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right\|^2 \mathrm{d}t \right] && (33) \\
&+ \sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| \nabla \log q^k_{\mathrm{GP},y^\star,T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q^k_{\mathrm{GP},y^\star,T-t}(X^{\leftarrow}_{t_{j-1}}) \right\|^2 \mathrm{d}t \right] && (34) \\
&+ \sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| \nabla \log q^k_{\mathrm{GP},y^\star,T-t}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q^k_{\mathrm{GP},y^\star,T-t}(X^{\leftarrow}_t) \right\|^2 \mathrm{d}t \right]. && (35)
\end{aligned}$$

B.3.2.1. Bounding term (34) For the discretization error, we derive

$$\begin{aligned}
&\mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\left\| \nabla \log \frac{q^k_{\mathrm{GP},y^\star,T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})}{q^k_{\mathrm{GP},y^\star,T-t}} \right\|^2 \right] \\
\lesssim & \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[L_k^2 dh + L_k^2 h^2 \|X^{\leftarrow}_{t_{j-1}}\|^2 + L_k^2 h^2 \|\nabla \log q^k_{\mathrm{GP},y^\star,T-t}(X^{\leftarrow}_{t_{j-1}})\|^2 \right] \\
\lesssim & \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[L_k^2 dh + L_k^2 h^2 \|X^{\leftarrow}_{t_{j-1}}\|^2 + L_k^2 h^2 (\|\nabla \log q^k_{\mathrm{GP},y^\star,T-t}(X^{\leftarrow}_t)\|^2 + L_k^2 \|X^{\leftarrow}_{t_{j-1}} - X^{\leftarrow}_t\|^2) \right] \\
\lesssim & L_k^2 dh + L_k^2 h^2 \sqrt{1 + \frac{v_k^2}{\tau^2}} \exp\left(\frac{(1+r_k)^2}{2\tau^2} \right) \mathfrak{m}_2^2,
\end{aligned} \tag{36}$$

where the last inequality holds by $h \lesssim L_k^{-1}$, (Chen et al., 2023, Lemmas 10, 11), and Proposition 4.

B.3.2.2. Bounding term (35) We have

$$\mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\left\| \nabla \log q^k_{\mathrm{GP},y^\star,T-t}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q^k_{\mathrm{GP},y^\star,T-t}(X^{\leftarrow}_t) \right\|^2 \right]$$

$$\leq L_k^2 \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\left\| X^{\leftarrow}_{t_{j-1}} - X^{\leftarrow}_t \right\|^2 \right] \lesssim L_k^2 (dh + h^2 \sqrt{1 + v_k^2/\tau^2} e^{(1+r_k)^2/2\tau^2} \mathfrak{m}_2^2), \tag{37}$$

where the last inequality holds by (Chen et al., 2023, Lemma 11) and Proposition 4.

B.3.2.3. Bounding term (33) For any $t \in [0, T]$, let

$$\hat{x}_0^\theta(x_t^{\leftarrow}, T-t) = \frac{1}{\sqrt{\bar{\alpha}_{T-t}}} \left(x_t^{\leftarrow} + (1 - \bar{\alpha}_{T-t}) s^\theta_{T-t}(x_t^{\leftarrow}) \right).$$

We derive

$$\begin{aligned}
&\sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| \tilde{s}_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q^k_{\mathrm{GP},y^\star,T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right\|^2 \mathrm{d}t \right] \\
\lesssim &\underbrace{\sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| s^\theta_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right\|^2 \mathrm{d}t \right]}_{E_0 \text{ (Unconditional score estimation error)}} \\
&+ \underbrace{\sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| \nabla \log p^k_{\mathrm{GP}} \left(y^\star \mid \hat{x}_0^\theta(X^{\leftarrow}_{t_{j-1}}, T-t_{j-1}) \right) - \nabla \log p^k_{\mathrm{GP}} \left(y^\star \mid m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right) \right\|^2 \mathrm{d}t \right]}_{E_1 \text{ (Posterior-mean estimation error)}} \\
&+ \underbrace{\sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| \nabla \log p^k_{\mathrm{GP}} \left(y^\star \mid m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right) - \nabla \log \bar{p}^k_{\mathrm{GP},T-t_{j-1}} \left(y^\star \mid X^{\leftarrow}_{t_{j-1}} \right) \right\|^2 \mathrm{d}t \right]}_{E_2 \text{ (Posterior-averaging error)}}.
\end{aligned}$$

Note that the gradients are taken w.r.t. $X^{\leftarrow}_{t_{j-1}}$.

Bounding Unconditional Score Estimation Error E_0 By Assumption 3, we have

$$\begin{aligned}
E_0 = &\sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| s^\theta_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right\|^2 \mathrm{d}t \right] \\
\leq &\sqrt{1 + \frac{v_k^2}{\tau^2}} \exp\left(\frac{(1+r_k)^2 + 1}{2\tau^2} \right) \sum_{j=1}^{J} \mathbb{E}_{Q^{\leftarrow}_{y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| s^\theta_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right\|^2 \mathrm{d}t \right] \\
\leq &\sqrt{1 + \frac{v_k^2}{\tau^2}} \exp\left(\frac{(1+r_k)^2 + 2}{2\tau^2} \right) \epsilon^2_{\mathrm{uncond}} T,
\end{aligned}$$

where the second and the third inequalities hold by Proposition 3 and Proposition 2, respectively.

Bounding Posterior-mean Estimation Error E_1 Assume

$$\| J_{\hat{x}_0^\theta}(x_t, t) \|_{\mathrm{op}} \leq \hat{J}_{\max}, \tag{38}$$

where $\| \cdot \|_{\mathrm{op}}$ is the operator norm and $J_{\hat{x}_0^\theta}(x_t, t) := \nabla_{x_t} \hat{x}_0^\theta(x_t, t)$ is the Jacobian matrix of the posterior-mean prediction map. We have

$$\begin{aligned}
&\left\| \nabla \log p^k_{\mathrm{GP}} \left(y^\star \mid \hat{x}_0^\theta(X^{\leftarrow}_{t_{j-1}}, T-t_{j-1}) \right) - \nabla \log p^k_{\mathrm{GP}} \left(y^\star \mid m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right) \right\|^2 \\
\lesssim &\hat{J}^2_{\max} \tilde{L}_k^2 \| m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \hat{x}_0^\theta(X^{\leftarrow}_{t_{j-1}}) \|^2 + G_k^2 \| J_{\hat{x}_0^\theta}(X^{\leftarrow}_{t_{j-1}}, T-t_{j-1}) - J_m(X^{\leftarrow}_{t_{j-1}}, T-t_{j-1}) \|^2_{\mathrm{op}} \\
\lesssim &\frac{\hat{J}^2_{\max} \tilde{L}_k^2 (1 - \bar{\alpha}_{T-t_{j-1}})^2}{\bar{\alpha}_{T-t_{j-1}}} \left\| s^\theta_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right\|^2 \\
&+ \frac{G_k^2 (1 - \bar{\alpha}_{T-t_{j-1}})^2}{\bar{\alpha}_{T-t_{j-1}}} \left\| \nabla s^\theta_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla^2 \log q_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right\|^2_{\mathrm{op}}
\end{aligned}$$

Therefore, we derive

$$
\begin{aligned}
E_1 \lesssim & \hat{J}_{\max}^2 \tilde{L}_k^2 \sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\frac{(1-\bar{\alpha}_{T-t_{j-1}})^2}{\bar{\alpha}_{T-t_{j-1}}} \int_{t_{j-1}}^{t_j} \left\| s^\theta_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right\|^2 \mathrm{d}t \right] \\
& + G_k^2 \sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\frac{(1-\bar{\alpha}_{T-t_{j-1}})^2}{\bar{\alpha}_{T-t_{j-1}}} \int_{t_{j-1}}^{t_j} \left\| \nabla s^\theta_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla^2 \log q_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right\|_{\mathrm{op}}^2 \mathrm{d}t \right] \\
\lesssim & \sqrt{1+\frac{v_k^2}{\tau^2}} \exp\left(\frac{(1+r_k)^2+2}{2\tau^2} \right) \left(\hat{J}_{\max}^2 \tilde{L}_k^2 \bar{\epsilon}^2_{\mathrm{uncond}} + G_k^2 \bar{\epsilon}^2_{\mathrm{Hess}} \right),
\end{aligned}
$$

where the last inequality holds by Propositions 2 and 3, where we define

$$
\bar{\epsilon}^2_{\mathrm{uncond}} := \sum_{j=1}^{J} \mathbb{E}_{q_{t_j}} \left[\frac{(1-\bar{\alpha}_{t_j})^2}{\bar{\alpha}_{t_j}} \int_{t_{j-1}}^{t_j} \left\| s^\theta_{t_j}(x_{t_j}) - \nabla \log q_{t_j}(x_{t_j}) \right\|^2 \mathrm{d}t \right], \tag{39}
$$

$$
\bar{\epsilon}^2_{\mathrm{Hess}} := \sum_{j=1}^{J} \mathbb{E}_{q_{t_j}} \left[\frac{(1-\bar{\alpha}_{t_j})^2}{\bar{\alpha}_{t_j}} \int_{t_{j-1}}^{t_j} \left\| \nabla s^\theta_{t_j}(x_{t_j}) - \nabla^2 \log q_{t_j}(x_{t_j}) \right\|_{\mathrm{op}}^2 \mathrm{d}t \right]. \tag{40}
$$

Bounding Posterior-averaging Error E_2 First, we have

$$
\begin{aligned}
E_2 = & \sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| \nabla \log p^k_{\mathrm{GP}} \left(y^\star \mid m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) \right) - \nabla \log \bar{p}^k_{\mathrm{GP},T-t_{j-1}} \left(y^\star \mid X^{\leftarrow}_{t_{j-1}} \right) \right\|^2 \mathrm{d}t \right] \\
\lesssim & \underbrace{\sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| \nabla \log \frac{\mathcal{N}\left(y^\star; \mu_k(m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})), \sigma_k^2(m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})) + \tau^2 \right)}{\mathcal{N}\left(y^\star; f^\star(m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})), \tau^2 \right)} \right\|^2 \mathrm{d}t \right]}_{E_{2,1}} \\
& + \underbrace{\sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| \nabla \log \bar{p}_{T-t_{j-1}}(y^\star | X^{\leftarrow}_{t_{j-1}}) - \nabla \log \bar{p}^k_{\mathrm{GP},T-t_{j-1}} \left(y^\star \mid X^{\leftarrow}_{t_{j-1}} \right) \right\|^2 \mathrm{d}t \right]}_{E_{2,2}} \\
& + \underbrace{\sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left[\int_{t_{j-1}}^{t_j} \left\| \nabla \log \mathcal{N}\left(y^\star; f^\star(m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})), \tau^2 \right) - \nabla \log \bar{p}_{T-t_{j-1}}(y^\star | X^{\leftarrow}_{t_{j-1}}) \right\|^2 \mathrm{d}t \right]}_{E_{2,3}}.
\end{aligned}
$$

- Step A. Bound term $E_{2,1}$. Let $J_m(x_t,t) := \nabla_{x_t} m_t(x_t)$ denote the Jacobian matrix of the posterior-mean map. Assume

$$
\|J_m(x_t,t)\|_{\mathrm{op}} \le J_{\max}, \tag{41}
$$

$$
\sup_x \|\nabla \mu_k(x) - \nabla f^\star(x)\| \le r_{\nabla,k}, \tag{42}
$$

$$
\sup_x \|\nabla \sigma_k^2(x)\| \le v_{\nabla,k}. \tag{43}
$$

Let $m = m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})$. By the chain rule, we have

$$
\begin{aligned}
& \nabla \log \frac{\mathcal{N}\left(y^\star; \mu_k(m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})), \sigma_k^2(m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})) + \tau^2 \right)}{\mathcal{N}\left(y^\star; f^\star(m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})), \tau^2 \right)} \\
= & J_m(X^{\leftarrow}_{t_{j-1}}, T-t_{j-1})^\top \nabla_m \log \frac{\mathcal{N}\left(y^\star; \mu_k(m), \sigma_k^2(m) + \tau^2 \right)}{\mathcal{N}\left(y^\star; f^\star(m), \tau^2 \right)} \\
= & J_m(X^{\leftarrow}_{t_{j-1}}, T-t_{j-1})^\top \nabla_m \left(\log \frac{\tau}{\sqrt{\sigma_k^2(m)+\tau^2}} + \frac{(y^\star - f^\star(m))^2}{2\tau^2} - \frac{(y^\star - \mu_k(m))^2}{2(\sigma_k^2(m)+\tau^2)} \right)
\end{aligned}
$$

$$=J_m(X^{\leftarrow}_{t_{j-1}}, T-t_{j-1})^\top \left(\frac{y^\star-\mu_k}{\sigma_k^2+\tau^2}\nabla\mu_k - \frac{y^\star-f^\star}{\tau^2}\nabla f^\star + \frac{1}{2}\left(\frac{(y^\star-\mu_k)^2}{(\sigma_k^2+\tau^2)^2} - \frac{1}{\sigma_k^2+\tau^2}\right)\nabla\sigma_k^2 \right).$$

Hence, we bound

$$\begin{aligned}
&\left\| \nabla_m \log \frac{\mathcal{N}\left(y^\star; \mu_k(m), \sigma_k^2(m), \tau^2\right)}{\mathcal{N}\left(y^\star; f^\star(m), \tau^2\right)} \right\| \\
\leq& \frac{(1+r_k)^2}{\tau^2}\left\|\nabla\mu_k - \nabla f^\star\right\| + \left(\frac{r_k}{\tau^2}+\frac{v_k^2}{\tau^4}\right)\left\|\nabla f^\star\right\| + \frac{1}{2}\left(\frac{1+r_k}{\tau^4}+\frac{1}{\tau^2}\right)\left\|\nabla\sigma_k^2\right\| \\
\leq& \frac{(1+r_k)r_{\nabla,k}}{\tau^2} + \left(\frac{r_k}{\tau^2}+\frac{v_k^2}{\tau^4}\right)B^\star + \frac{1}{2}\left(\frac{(1+r_k)^2}{\tau^4}+\frac{1}{\tau^2}\right)v_{\nabla,k}.
\end{aligned}$$

Therefore, we derive

$$E_{2,1} \lesssim J_{\max}^2 \left[\frac{(1+r_k)r_{\nabla,k}}{\tau^2} + \left(\frac{r_k}{\tau^2}+\frac{v_k^2}{\tau^4}\right)B^\star + \frac{1}{2}\left(\frac{(1+r_k)^2}{\tau^4}+\frac{1}{\tau^2}\right)v_{\nabla,k} \right]^2 T. \tag{44}$$

- Step B. Bound term $E_{2,2}$. Assume

$$\sum_{j=1}^{J} \mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}} \left\{ \int_{t_{j-1}}^{t_j} \frac{\bar\alpha_{T-t_{j-1}}}{(1-\bar\alpha_{T-t_{j-1}})^2} \mathrm{Tr}\left[\mathrm{Cov}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}(x_0) \right] \mathrm{d}t \right\} \leq B_k, \tag{45}$$

$$\left\| \nabla \bar p^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right) \right\| \leq M_k. \tag{46}$$

and

$$p^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right) \geq \frac{1}{\sqrt{2\pi(\tau^2+v_k^2)}} \exp\left(-\frac{(1+r_k)^2}{2\tau^2}\right) =: c_{\tau,k}. \tag{47}$$

Note that

$$p_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}}) \geq \frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{1}{2\tau^2}\right) = c_\tau \tag{48}$$

$$\tag{49}$$

We obtain

$$\begin{aligned}
&\left\| \nabla \log \bar p_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}}) - \nabla \log \bar p^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right) \right\|^2 \\
=& \left\| \frac{\nabla \bar p_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}})}{\bar p_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}})} - \frac{\nabla \bar p^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right)}{\bar p^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right)} \right\|^2 \\
=& \left\| \frac{\nabla \bar p_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}}) - \nabla \bar p^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right)}{\bar p_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}})} \right. \\
&\left. + \nabla \bar p^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right)\left(\frac{1}{\bar p_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}})} - \frac{1}{\bar p^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right)} \right) \right\|^2 \\
\lesssim& \frac{1}{c_\tau^2}\left\| \nabla \bar p_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}}) - \nabla \bar p^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right) \right\|^2 \qquad (50) \\
&+ \frac{1}{c_\tau^2 c_{\tau,k}^2}\left\| \nabla \bar p^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right) \right\|^2 \left| \bar p_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}}) - \bar p^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right) \right|^2. \qquad (51)
\end{aligned}$$

Let

$$\phi(x) := \mathcal{N}(y^\star; f^\star(x), \tau^2) - \mathcal{N}(y^\star; \mu_k(x), \sigma_k^2(x)+\tau^2).$$

Sub-step B.1. Bound term (50). We utilize Lemma 1 and derive

$$\begin{aligned}
&\nabla \bar{p}_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}}) - \nabla \bar{p}^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right)\\
=&\nabla \mathbb{E}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}\left[\phi(x_0)\right]\\
=&\mathrm{Cov}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}\left(\phi(x_0), \nabla \log q_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}|x_0)\right)\\
=&\mathrm{Cov}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}\left(\phi(x_0), \frac{\sqrt{\bar{\alpha}_{T-t_{j-1}}}x_0 - X^{\leftarrow}_{t_{j-1}}}{1-\bar{\alpha}_{T-t_{j-1}}}\right)\\
=&\frac{\sqrt{\bar{\alpha}_{T-t_{j-1}}}}{1-\bar{\alpha}_{T-t_{j-1}}}\mathrm{Cov}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}\left(\phi(x_0), x_0\right) \qquad (\mathrm{Cov}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}(\phi(x_0), X^{\leftarrow}_{t_{j-1}}) = 0)\\
=&\frac{\sqrt{\bar{\alpha}_{T-t_{j-1}}}}{1-\bar{\alpha}_{T-t_{j-1}}}\mathbb{E}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}\left[\left(\phi(x_0) - \mathbb{E}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}[\phi(x_0)]\right)\left(x_0 - m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})\right)\right]
\end{aligned}$$

By Cauchy-Schwartz inequality, we further obtain

$$\begin{aligned}
&\left\|\nabla \bar{p}_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}}) - \nabla \bar{p}^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right)\right\|^2\\
\le&\frac{\bar{\alpha}_{T-t_{j-1}}}{(1-\bar{\alpha}_{T-t_{j-1}})^2}\mathrm{Var}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}[\phi(x_0)]\mathrm{Tr}\left[\mathrm{Cov}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}(x_0)\right]\\
\le&\frac{\bar{\alpha}_{T-t_{j-1}}}{(1-\bar{\alpha}_{T-t_{j-1}})^2}\mathrm{Tr}\left[\mathrm{Cov}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}(x_0)\right]\mathbb{E}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}\left[\phi^2(x_0)\right]\\
\le&\frac{\bar{\alpha}_{T-t_{j-1}}(L_\tau r_k + L_{\tau,\mathrm{Var}}v_k^2)^2}{(1-\bar{\alpha}_{T-t_{j-1}})^2}\mathrm{Tr}\left[\mathrm{Cov}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}(x_0)\right],
\end{aligned}$$

where the last inequality holds by the analysis in Appendix B.4.1, that is,

$$\begin{aligned}
|\phi(x)| \le&|\mathcal{N}(y^\star; f^\star(x), \tau^2) - \mathcal{N}(y^\star; \mu_k(x), \tau^2)| + |\mathcal{N}(y^\star; \mu_k(x), \tau^2) - \mathcal{N}(y^\star; \mu_k(x), \sigma_k^2(x)+\tau^2)|\\
\le&L_\tau r_k + L_{\tau,\mathrm{Var}}v_k^2.
\end{aligned}$$

Sub-step B.2. Bound term (51). Note that

$$\begin{aligned}
&\left|\bar{p}_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}}) - \bar{p}^k_{\mathrm{GP},T-t_{j-1}}\left(y^\star \mid X^{\leftarrow}_{t_{j-1}}\right)\right|^2\\
\le&\mathbb{E}_{q_0(\cdot|X^{\leftarrow}_{t_{j-1}})}[\phi^2(x_0)]\\
\le&(L_\tau r_k + L_{\tau,\mathrm{Var}}v_k^2)^2.
\end{aligned}$$

Therefore, we derive

$$E_{2,2} \lesssim \frac{(L_\tau r_k + L_{\tau,\mathrm{Var}}v_k^2)^2}{c_\tau^2}B_k + \frac{(L_\tau r_k + L_{\tau,\mathrm{Var}}v_k^2)^2}{c_\tau^2 c_{\tau,k}^2}M_k^2 T \tag{52}$$

Step C. Bound term $E_{2,3}$. Define

$$\epsilon^2_{\text{post-avg}} := \sum_{j=1}^{J}\mathbb{E}_{Q^{\leftarrow}_{y^\star,T}}\left[\int_{t_{j-1}}^{t_j}\left\|\nabla \log \mathcal{N}\left(y^\star; f^\star(m_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}})), \tau^2\right) - \nabla \log \bar{p}_{T-t_{j-1}}(y^\star|X^{\leftarrow}_{t_{j-1}})\right\|^2 \mathrm{d}t\right]. \tag{53}$$

By Proposition 3, we have that

$$E_{2,3} \le \sqrt{1+\frac{v_k^2}{\tau^2}}\exp\left(\frac{(1+r_k)^2+1}{2\tau^2}\right)\epsilon^2_{\text{post-avg}} \tag{54}$$

B.4.3 Putting Everything Together

By the same localization and approximation argument as in the proof of (Chen et al., 2023, Theorem 9), the Girsanov path-space KL bound remains valid without assuming Novikov's condition. Combining this and Inequality (32), we have that

$$D_{\mathrm{KL}}(Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}\|\tilde{Q}^{k,\leftarrow}_{\mathrm{GP},y^\star,T}) \le \sum_{j=1}^{J}\mathbb{E}_{Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T}}\int_{t_{j-1}}^{t_j}\left\|\tilde{s}_{T-t_{j-1}}(X^{\leftarrow}_{t_{j-1}}) - \nabla \log q^k_{\mathrm{GP},y^\star,T-t}(X^{\leftarrow}_t)\right\|^2 \mathrm{d}t.$$

Therefore, we derive

$$
\begin{aligned}
D_{\mathrm{TV}}(q_{y^\star}, \hat{q}^k_{y^\star}) \leq & D_{\mathrm{TV}}(q_{y^\star}, q^k_{\mathrm{GP},y^\star}) + D_{\mathrm{TV}}(q^k_{\mathrm{GP},y^\star}, \tilde{q}^k_{y^\star}) + D_{\mathrm{TV}}(q^k_{\mathrm{GP},y^\star,T}, \varepsilon) \\
\leq & D_{\mathrm{TV}}(q_{y^\star}, q^k_{\mathrm{GP},y^\star}) + \sqrt{D_{\mathrm{KL}}(q^k_{\mathrm{GP},y^\star} \| \varepsilon)}\exp(-T) + \sqrt{D_{\mathrm{KL}}(Q^{k,\leftarrow}_{\mathrm{GP},y^\star,T} \| \tilde{Q}^{k,\leftarrow}_{\mathrm{GP},y^\star,T})} \\
\lesssim & \frac{L_\tau r_k + L_{\tau,\mathrm{Var}} v_k^2}{c_\tau} + \sqrt{D_{\mathrm{KL}}(q^k_{\mathrm{GP},y^\star} \| \varepsilon)}\exp(-T) \\
& + \sqrt{J^2_{\max}\left[\frac{(1+r_k) r_{\nabla,k}}{\tau^2} + \left(\frac{r_k}{\tau^2} + \frac{v_k^2}{\tau^4}\right) B^\star + \frac{1}{2}\left(\frac{(1+r_k)^2}{\tau^4} + \frac{1}{\tau^2}\right) v_{\nabla,k}\right]^2 T} \\
& + \frac{L_\tau r_k + L_{\tau,\mathrm{Var}} v_k^2}{c_\tau}\sqrt{B_k + \frac{M_k^2}{c^2_{\tau,k}} T} \\
& + \left(1 + \frac{v_k^2}{\tau^2}\right)^{1/4} \exp\left(\frac{(1+r_k)^2 + 2}{4\tau^2}\right) \sqrt{\epsilon^2_{\text{uncond}} T + \hat{J}^2_{\max} \tilde{L}^2_k \bar{\epsilon}^2_{\text{uncond}} + G^2_k \bar{\epsilon}^2_{\text{Hess}}} \\
& + \left(1 + \frac{v_k^2}{\tau^2}\right)^{1/4} \exp\left(\frac{(1+r_k)^2 + 1}{4\tau^2}\right) \epsilon_{\text{post-avg}} \\
& + \sqrt{L_k^2 dh + \sqrt{1 + \frac{v_k^2}{\tau^2}} \exp\left(\frac{(1+r_k)^2}{2\tau^2}\right) L_k^2 h^2 \mathfrak{m}_2^2}
\end{aligned}
$$

B.5 Auxiliary Proofs

Proposition 1 (Density ration between target and data distribution)**.** *Under Assumption 1 and the target reward $y^\star \in [0,1]$, we have that*

$$
\left\| \frac{\mathrm{d}q_{y^\star}}{\mathrm{d}q_{\text{data}}} \right\|_\infty, \left\| \frac{\mathrm{d}q_{\text{data}}}{\mathrm{d}q_{y^\star}} \right\|_\infty \leq \exp\left(\frac{1}{2\tau^2}\right). \tag{55}
$$

Proof. By definition, we have

$$
\frac{q_{y^\star}(x)}{q_{\text{data}}(x)} = \frac{\mathcal{N}(y^\star; f^\star(x), \tau^2)}{Z_{y^\star}}.
$$

Since $y^\star, f^\star(x) \in [0,1]$, we have

$$
\begin{aligned}
\frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{1}{2\tau^2}\right) \leq \mathcal{N}(y^\star; f^\star(x), \tau^2) \leq \frac{1}{\sqrt{2\pi}\tau}, \\
\frac{1}{\sqrt{2\pi}\tau}\exp\left(-\frac{1}{2\tau^2}\right) \leq Z_{y^\star} \leq \frac{1}{\sqrt{2\pi}\tau},
\end{aligned}
$$

which concludes the proof. □

Proposition 2. *Under Assumptions 3, we have*

$$
\mathbb{E}_{x_t \sim q_{y^\star,t}}\left[\left\| s^\theta_t(x_t) - \nabla \log q_t(x_t) \right\|^2\right] \leq \exp\left(\frac{1}{2\tau^2}\right)\epsilon^2_{\text{uncond}}
$$

Proposition 3 (Density ratio between target and GP-conditioned distribution)**.** *For any $t \in [0,T]$, we have that*

$$
\left\| \frac{q^k_{\mathrm{GP},y^\star,t}}{q_{y^\star,t}} \right\|_\infty \leq \sqrt{1 + \frac{v_k^2}{\tau^2}} \exp\left(\frac{(1+r_k)^2 + 1}{2\tau^2}\right). \tag{56}
$$

Proof. We first prove the density-ratio bound at time $t = 0$, and then use the fact that the forward noising kernel preserves the upper bounds.

Case 1. $t = 0$. First, we have that

$$\frac{q^k_{\mathrm{GP},y^\star}(x)}{q_{y^\star}(x)} = \frac{Z_{y^\star}}{Z_k}\frac{\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)}{\mathcal{N}(y^\star;f^\star(x),\tau^2)}.$$

Note that

$$\frac{\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)}{\mathcal{N}(y^\star;f^\star(x),\tau^2)} = \frac{\tau}{\sqrt{\tau^2+\sigma_k^2(x)}}\exp\left(\frac{(y^\star-f^\star(x))^2}{2\tau^2} - \frac{(y^\star-\mu_k(x))^2}{2(\tau^2+\sigma_k^2(x))}\right) \le \exp\left(\frac{1}{2\tau^2}\right).$$

On the other hand, we obtain

$$\begin{aligned}\frac{\mathcal{N}(y^\star;f^\star(x),\tau^2)}{\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)} =& \frac{\sqrt{\tau^2+\sigma_k^2(x)}}{\tau}\exp\left(\frac{(y^\star-\mu_k(x))^2}{2(\tau^2+\sigma_k^2(x))} - \frac{(y^\star-f^\star(x))^2}{2\tau^2}\right)\\ \le& \sqrt{1+\frac{v_k^2}{\tau^2}}\exp\left(\frac{(1+r_k)^2}{2\tau^2}\right).\end{aligned}$$

Hence, we derive

$$\begin{aligned}Z_{y^\star} &= \int q_{\text{data}}(x)\mathcal{N}(y^\star;f^\star(x),\tau^2)\mathrm{d}x\\ &= \int q_{\text{data}}(x)\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)\frac{\mathcal{N}(y^\star;f^\star(x),\tau^2)}{\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)}\mathrm{d}x\\ &\le \sqrt{1+\frac{v_k^2}{\tau^2}}\exp\left(\frac{(1+r_k)^2}{2\tau^2}\right)\int q_{\text{data}}(x)\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)\mathrm{d}x = \sqrt{1+\frac{v_k^2}{\tau^2}}\exp\left(\frac{(1+r_k)^2}{2\tau^2}\right)Z_k.\end{aligned}$$

Therefore, we prove

$$\frac{q^k_{\mathrm{GP},y^\star}(x)}{q_{y^\star}(x)} \le \sqrt{1+\frac{v_k^2}{\tau^2}}\exp\left(\frac{(1+r_k)^2+1}{2\tau^2}\right).$$

Case 2. $t \in [0,T]$**.** Note that

$$\begin{aligned}q^k_{\mathrm{GP},y^\star,t}(x_t) &= \int q^k_{\mathrm{GP},y^\star,t}(x_t|x_0)q^k_{\mathrm{GP},y^\star}(x_0)\mathrm{d}x_0\\ &= \int q_{y^\star,t}(x_t|x_0)q_{y^\star}(x_0)\frac{q^k_{\mathrm{GP},y^\star}(x_0)}{q_{y^\star}(x_0)}\mathrm{d}x_0 \qquad (q_{y^\star,t}(x_t|x_0) = q^k_{\mathrm{GP},y^\star,t}(x_t|x_0))\\ &\le \sqrt{1+\frac{v_k^2}{\tau^2}}\exp\left(\frac{(1+r_k)^2+1}{2\tau^2}\right)\int q_{y^\star,t}(x_t|x_0)q_{y^\star}(x_0)\mathrm{d}x_0\\ &= \sqrt{1+\frac{v_k^2}{\tau^2}}\exp\left(\frac{(1+r_k)^2+1}{2\tau^2}\right)q_{y^\star,t}(x_t),\end{aligned}$$

which concludes the proof. □

Proposition 4. *Under Assumptions 1 and 6, $q^k_{\mathrm{GP},y^\star}$ has a finite second moment.*

Proof. Recall that

$$q^k_{\mathrm{GP},y^\star}(x) = \frac{q_{\text{data}}(x)\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)}{Z_k}, \tag{57}$$

where $Z_k = \int q_{\text{data}}(x)\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)\mathrm{d}x$ is the normalizer. Then, we have

$$\mathbb{E}_{q^k_{\mathrm{GP},y^\star}}\left[\|x\|^2\right] = \frac{1}{Z_k}\int \|x\|^2 q_{\text{data}}(x)\mathcal{N}(y^\star;\mu_k(x),\sigma_k^2(x)+\tau^2)\mathrm{d}x \le \frac{1}{Z_k\sqrt{2\pi\tau}}\mathbb{E}_{q_{\text{data}}}\left[\|x\|^2\right].$$

Further, since $|\mu_k(x) - f^\star(x)| \le r_k$ and $\sigma_k^2 \le v_k^2$, we have that $Z_k \ge \frac{1}{\sqrt{2\pi(\tau^2+v_k^2)}}\exp(-(1+r_k)^2/2\tau^2)$. Therefore, we derive

$$\mathbb{E}_{q^k_{\mathrm{GP},y^\star}}\left[\|x\|^2\right] \le \sqrt{1+\frac{v_k^2}{\tau^2}}\exp\left(\frac{(1+r_k)^2}{2\tau^2}\right)\mathbb{E}_{q_{\text{data}}}\left[\|x\|^2\right] \le \sqrt{1+\frac{v_k^2}{\tau^2}}\exp\left(\frac{(1+r_k)^2}{2\tau^2}\right)\mathfrak{m}_2^2,$$

which concludes the proof. □

Lemma 1 (Derivative of posterior mean). *Let $\phi : \mathcal{X} \to \mathbb{R}$ be a function with bounded values. For $t > 0$, we have that*

$$\nabla_{x_t} \mathbb{E}_{q_0(\cdot|x_t)} [\phi(x_0)|x_t] = \mathrm{Cov}_{q_0(\cdot|x_t)} (\phi(x_0), \nabla_{x_t} \log q_t(x_t|x_0)) . \tag{58}$$

Proof. Recall that

$$q_0(x_0|x_t) = \frac{q_{\text{data}}(x_0) q_t(x_t|x_0)}{q_t(x_t)},$$

where $q_t(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I)$. Define $F(x_t) := \mathbb{E}_{q_0(\cdot|x_t)}[\phi(x_0)|x_t] = \int \phi(x_0) q_0(x_0|x_t) \mathrm{d}x_0$. We have that

$$\nabla F(x_t) = \frac{\nabla A(x_t)}{q_t(x_t)} - \frac{A(x_t)}{q_t(x_t)} \nabla \log q_t(x_t),$$

where $A(x_t) := \int \phi(x_0) q_{\text{data}}(x_0) q_t(x_t|x_0) \mathrm{d}x_0$. Since ϕ is bounded and the Gaussian kernel $q_t(x_t|x_0)$ is smooth in x_t, we can interchange the derivative and the integral and obtain that

$$\nabla A(x_t) = \int \phi(x_0) q_{\text{data}}(x_0) \nabla q_t(x_t|x_0) \mathrm{d}x_0 = \int \phi(x_0) q_{\text{data}}(x_0) q_t(x_t|x_0) \nabla \log q_t(x_t|x_0) \mathrm{d}x_0.$$

Hence, we have that

$$\frac{\nabla A(x_t)}{q_t(x_t)} = \int \phi(x_0) \nabla \log q_t(x_t|x_0) q_0(x_0|x_t) \mathrm{d}x_0 = \mathbb{E}_{q_0(\cdot|x_t)}[\phi(x_0) \nabla \log q_t(x_t|x_0)].$$

On the other hand, we have that

$$\frac{A(x_t)}{q_t(x_t)} \nabla \log q_t(x_t) = \nabla \log q_t(x_t) \int \phi(x_0) q_0(x_0|x_t) \mathrm{d}x_0 = \nabla \log q_t(x_t) \mathbb{E}_{q_0(\cdot|x_t)}[\phi(x_0)].$$

Moreover, note that

$$\nabla \log q_t(x_t) = \frac{\nabla q_t(x_t)}{q_t(x_t)} = \frac{\int q_{\text{data}}(x_0) \nabla q_t(x_t|x_0) \mathrm{d}x_0}{q_t(x_t)} = \mathbb{E}_{q_0(\cdot|x_t)}[\nabla \log q_t(x_t|x_0)],$$

where we utilize the fact that $\nabla_{x_t} q_t(x_t|x_0) = q_t(x_t|x_0) \nabla_{x_t} \log q_t(x_t|x_0)$ to derive the last equation. We conclude the proof. □

C Detailed Implementation

BReaD-SubA & BReaD-SubB At each round, BREAD fits a GP/DKL surrogate using the observed labeled samples collected so far. Given the current noisy sample at a denoising step, it first predicts the posterior mean (5) and extracts a differentiable feature representation from the predicted clean sample. The GP posterior is then evaluated on this representation to further compute target-conditioned likelihood in BReaD-SubA or the upper confidence bound (UCB) in BReaD-SubB. The resulting guidance gradient is differentiated with respect to the noisy sample by back-propagating through the feature extractor and the diffusion model.

MLP RM-guided The MLP-guided baseline uses a lightweight multilayer perceptron (MLP) to predict the black-box objective value from the same feature representation used by BREAD. The MLP shares the same architecture as the feature map within our deep kernel learning (DKL) surrogate, but directly outputs a scalar reward prediction. At each denoising step, it first predicts the posterior mean (5) of the current noisy sample $X_t^{\leftarrow}$. This estimate is then forwarded to extract the feature input to the MLP surrogate. Then, the guidance gradient is computed via backpropagation and used to modify the noise estimate in the forward universal guidance style.

DM-GP-UCB The GP-UCB baseline follows a generate-and-select strategy. At each iteration, it first generates a batch of 10 candidate samples from the pre-trained diffusion model. The UCB values of these samples are computed from a GP surrogate that is fitted on the observed data so far. The candidate with the highest UCB value is selected for evaluation at the current iteration.

fwd. uni-guided The implementation of the forward universal guidance baseline follows[2] (Bansal et al., 2024). At every timestep, the current noisy sample is used to predict the posterior mean 5. This estimate is passed to the true objective function to compute the exact reward gradient for guidance.

Input features for BReaD-SubA, BReaD-SubB, MLP RM-guided, and DM-GP-UCB. For a given experiment, all four methods use the same feature representation to ensure a consistent comparison. For image generation tasks in Section 4.1, the feature is the 768-dimensional CLIP embedding of the predicted clean image. For the ligand optimization task in Section 4.2, we obtain the 256-dimensional feature by first extracting the hidden representation immediately before the EGNN output projection and then mean-pooling the representations over the ligand atoms. For the WPQH experiment in Section 4.3, we parameterize the generated sequence of six actuators into 21 parameters and use these parameters as the surrogate input.

D Additional Details in Section 4.3: Results for Controllable Nuclear Fusion

Controllable nuclear fusion and Wide-Pedestal QH. Nuclear fusion is a promising path toward sustainable energy, and tokamaks are among the leading candidates for producing controlled thermonuclear fusion power. Conventional approaches to plasma control typically combine precomputed actuator trajectories, such as coil-current programs, with feedback controllers designed for specific target quantities. These control systems have demonstrated reliable performance within a wide range of discharges, but their design requires substantial engineering effort and expertise, especially for advanced plasma scenarios. One such example is accessing the Wide-Pedestal QH (WPQH) mode. The WPQH mode is an advanced plasma-confinement regime characterized by a wide edge transport barrier that enables high energy confinement while avoiding damaging edge-localized modes (ELMs).

Modeling DIII-D Tokamak dynamics using an RPNN ensemble. We follow (Char et al., 2024) and model plasma dynamics in the DIII-D Tokamak using an ensemble of recurrent probabilistic neural networks (RPNNs). Each RPNN autoregressively predicts the distribution of the change in plasma state between consecutive timesteps, conditioned on the current plasma state, actuator inputs, and a recurrent hidden state that summarizes the preceding trajectory. The predicted state difference is then added to the current state to obtain the next-state prediction. The network architecture is shown below. We train an ensemble of 25 RPNNs to predict 12 plasma states, consisting of 10 scalar states and two plasma profiles, with each profile represented by four PCA components, from a sequence of 14 actuators parameterized by 19 input variables. To evaluate the predictive accuracy of the learned dynamics model, we report the explained variance (EV) of the predicted timestep differences, i.e., the change from the current state to the next state, for each state variable in Table 2.

Network Architecture:

- **Encoder:**
  - Fully Connected (FC) layer: `input_dim` $\times$ 512
  - FC layer: 512 $\times$ 512
- **Memory Unit:**
  - Gated Recurrent Unit (GRU) block: 512 $\times$ 256
- **Decoder (with residual connections between FC layers):**
  - FC layer: 256 $\times$ 512
  - FC layers: 512 $\times$ 512 (repeated 8 times)
  - FC layer: 512 $\times$ 128
- **Output Heads:**
  - Mean head: 128 $\times$ `output_dim`
  - Log-variance head: 128 $\times$ `output_dim`

[2] https://github.com/arpitbansal297/Universal-Guided-Diffusion

Table 2: Per-variable explained variance (EV) of the RPNN ensemble trained on historical DIII-D experimental data. Higher values indicate better agreement with held-out real trajectories.

Variable	Ensemble EV	Variable	Ensemble EV
β_N	0.7983	denv1	0.4912
β_p	0.8137	temp_pca1	0.6892
W_{MHD}	0.8012	temp_pca2	0.5145
prmtan_tewid	0.3132	temp_pca3	0.3936
prmtan_newid	0.2460	temp_pca4	0.4799
prmtan_teped	0.3803	dens_pca1	0.6918
prmtan_neped	0.4126	dens_pca2	0.5333
FS04_max	0.5999	dens_pca3	0.5380
denr0	0.4538	dens_pca4	0.5692
Average Ensemble EV			**0.5400**

Training WPQH mode detector. To evaluate whether a plasma trajectory accesses the WPQH mode, we train a binary classifier to identify WPQH operation at each timestep. Our labeled dataset contains 477 DIII-D shots that successfully accessed WPQH and 308 shots that did not. After data preprocessing, including removing shots with missing required signals, 434 shots remain for training the detector. The WPQH detector is a multilayer perceptron (MLP) that predicts whether the plasma is in the WPQH mode at a given timestep using a local temporal window of plasma states and actuator signals spanning 60 ms before and after the target timestep. At each timestep, we use five scalar plasma states (β_N, W_{MHD}, β_p, electron-temperature pedestal width (prmtan_tewid), and electron-density pedestal width (prmtan_newid)), the first two PCA components of the electron-temperature (T_e) and electron-density (n_e) profiles, and five actuator signals (NBI power P_{inj}, plasma current (I_p), NBI torque (T_{inj}), q_{95}, and separatrix distance (dR_{sep})). The resulting window is flattened and passed through two fully connected hidden layers with 128 and 64 units and ReLU activations, followed by a binary classification head.

We evaluate the detector under two settings. Real-data detection directly applies the detector to measured plasma trajectories and achieves 85.49% accuracy and 83.98% precision on WPQH predictions. Rollout-data detection first replays the experimental shots using the learned RPNN dynamics model and then applies the same detector to the simulated trajectories. Despite the additional error introduced by the rollouts, the detector achieves 72.46% accuracy and 69.21% precision on simulated trajectories. These results indicate that the detector retains reasonable predictive performance on model-generated rollouts, making it reasonable to use it to estimate the WPQH ratio in our simulation experiments.

Pre-training diffusion model on DIII-D dataset. We pre-train a conditional diffusion model (DM) on historical DIII-D trajectories to generate future plasma states and actuator sequences. Each training sample consists of a trajectory of 200 timesteps (4000ms), where the first 10 timesteps (200ms) are provided as a clean conditioning prefix, and the model generates the remaining 190 timesteps (3800ms). The DM jointly models plasma states and actuator sequences (in Table 3), and it enables both joint trajectory generation and conditional generation, i.e., predicting the corresponding plasma dynamics given the actuator trajectories. We parameterize the diffusion model using a temporal Diffusion Transformer (Esser et al., 2024, DiT). At each timestep, the plasma state and actuator signals are represented by separate tokens with dedicated input projections and learnable type embeddings. The model consists of eight transformer blocks with a hidden dimension of 256 and eight attention heads. Each block contains masked self-attention across the generated trajectory, cross-attention to the clean conditioning prefix, and a feed-forward network. The diffusion timestep is incorporated through adaptive layer normalization (AdaLN). For forward prediction, actuator signals are kept fixed while the DM denoises the plasma-state sequence. We train the model using an exponential noise schedule with 1k diffusion steps and an OU forward process. To evaluate the learned dynamics, we condition the model on the initial plasma state and actuator trajectory and measure the explained variance (EV) between the generated and real plasma trajectories.

E ADDITIONAL NUMERICAL RESULTS

Signal Group	Signals	Actuator	Controllable	State
Scalar States	β_N (Normalized Plasma Pressure)	✗	✗	✓
	β_p (Poloidal Beta)	✗	✗	✓
	MHD Stored Energy (W_{MHD})	✗	✗	✓
	Electron Temperature Pedestal Width (prmtan_tewid)	✗	✗	✓
	Electron Density Pedestal Width (prmtan_newid)	✗	✗	✓
	Electron Temperature Pedestal Height (prmtan_teped)	✗	✗	✓
	Electron Density Pedestal Height (prmtan_neped)	✗	✗	✓
	FilterScope ($\mathrm{FS04}_{\max}$)	✗	✗	✓
	`denr0`	✗	✗	✓
	`denv1`	✗	✗	✓
Profile States	Electron Temperature Profile (T_e)	✗	✗	✓
	Electron Density Profile (n_e)	✗	✗	✓
Plasma Control	NBI Total Power (P_{inj})	✓	✓	✗
	NBI Torque (T_{inj})	✓	✓	✗
	Plasma Current (I_p)	✓	✓	✗
	Safety Factor (q_{95})	✓	✓	✗
Shape Variables	Separatrix Distance (dR_{sep})	✓	✓	✗
	Bottom Triangularity (tribot)	✓	✗	✗
	Top Triangularity (tritop)	✓	✗	✗
	Elongation (κ)	✓	✗	✗
	Minor Radius (a_{minor})	✓	✗	✗
	Radial Positions of Magnetic Axis (R_{axis})	✓	✗	✗
	Vertical Positions of Magnetic Axis (Z_{axis})	✓	✗	✗
Electron Cyclotron Heating (ECH)	`ec_j_eccd_param1`	✓	✓	✗
	`ec_j_eccd_param2`	✓	✓	✗
	`ec_j_eccd_param3`	✓	✓	✗
	`ec_qrfe_param1`	✓	✓	✗
	`ec_qrfe_param2`	✓	✓	✗
	`ec_qrfe_param3`	✓	✓	✗
Other Actuators	Toroidal Field (B_T)	✓	✗	✗
	Gas Puffing (`gasA`)	✓	✗	✗

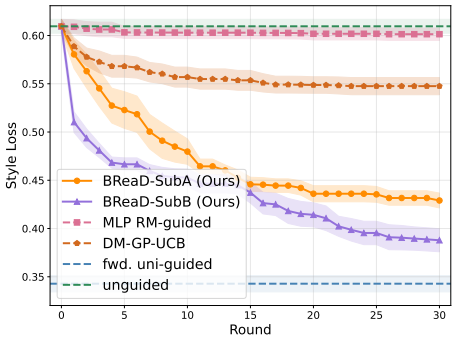
Table 3: Plasma state and actuator signals used for modeling the DIII-D tokamak dynamics in the WPQH experiments. The controllable column indicates actuator signals that can be optimized when searching for WPQH access strategies.



(a) BReaD-SubA



(b) BReaD-SubB



(c) Best-so-far style loss

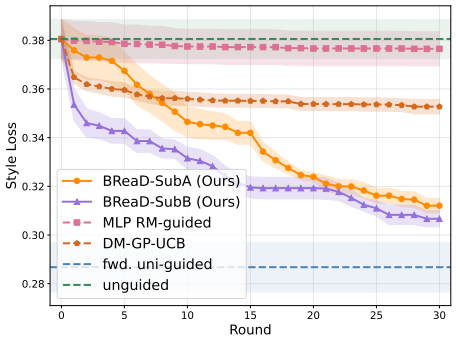
Figure 6: Style guidance on Stable Diffusion v1.5 to match the text prompt, *A portrait of a woman*, and the style of Pablo Picasso's painting, *Brick Factory at Tortosa*. Results are averaged over 10 trials.



(a) BReaD-SubA



(b) BReaD-SubB



(c) Best-so-far style loss

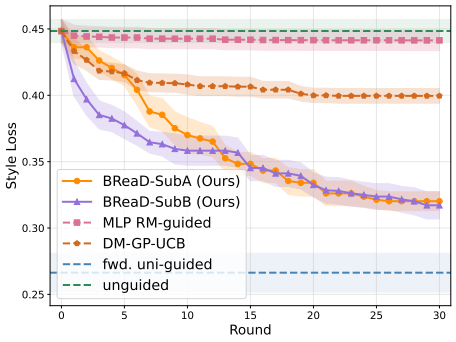
Figure 7: Style guidance on Stable Diffusion v1.5 to match the text prompt, *A colorful photo of an Eiffel Tower*, and the style of Vincent van Gogh's painting, *The Starry Night*. Results are averaged over 10 trials.



(a) BReaD-SubA



(b) BReaD-SubB



(c) Best-so-far style loss

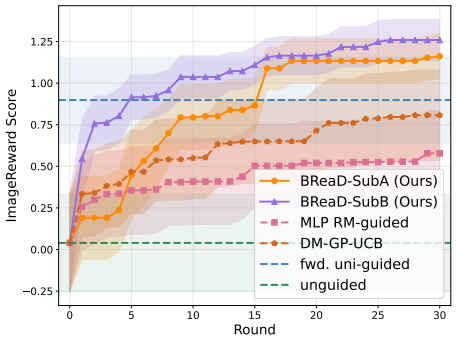
Figure 8: Style guidance on Stable Diffusion v1.5 to match the text prompt, *A fantasy photo of an volcanoes*, and the style of Leonid Afremov's painting, *Melody of The Night*. Results are averaged over 10 trials.



(a) unguided (Reward: −0.94)



(b) BReaD-SubB (Round 11, Reward: 1.37)



(c) Best-so-far ImageReward score

Figure 9: Guidance sampling of Stable Diffusion v1.5 to match the text prompt, *Landscape photography by Marc Adams, mountains with some forests, small lake in the center, fog in the background, sunrays, golden hour, high quality*, where the rewards are given by the ImageReward model. The images demonstrated use the same seed. Results are averaged over 10 trials.



(a) unguided



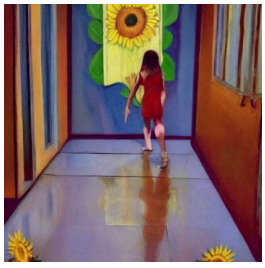
(b) fwd. uni-guided



(c) MLP-guided



(d) DM-GP-UCB

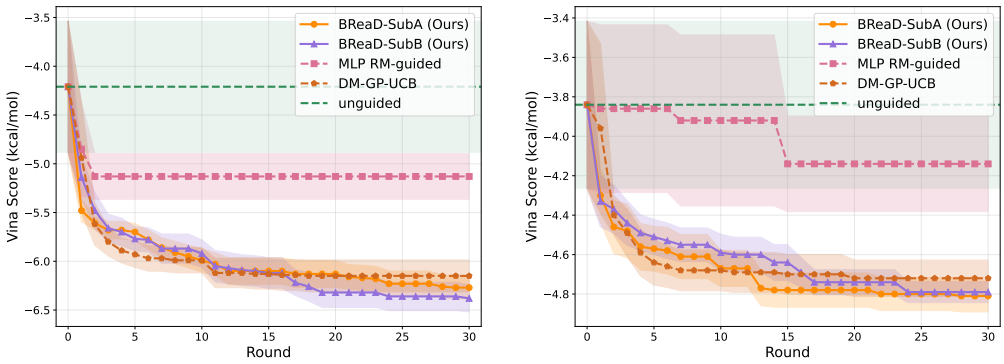


(e) BReaD-SubA



(f) BReaD-SubB

Figure 10: Image generation for optimizing the ImageReward score. *A painting of a girl walking in a hallway and suddenly finds a giant sunflower on the floor blocking her way.*



(a) Vascular Endothelial Growth Factor Receptor 2 (VEGFR2)

(b) Estrogen receptor alpha (ERα)

Figure 11: **Ligand optimization.** VEGFR2 and ERα. Results are averaged over 10 trials.