
Diff-BBO: Diffusion-Based Inverse Modeling for Black-Box Optimization

Dongxia Wu

University of California San Diego
La Jolla, CA
dowu@ucsd.edu

Nikki Lijing Kuang

University of California San Diego
La Jolla, CA
11kuang@ucsd.edu

Ruijia Niu

University of California San Diego
La Jolla, CA
rniu@ucsd.edu

Yi-An Ma

University of California San Diego
La Jolla, CA
yianma@ucsd.edu

Rose Yu

University of California San Diego
La Jolla, CA
roseyu@ucsd.edu

Abstract

Black-box optimization (BBO) aims to optimize an objective function by iteratively querying a black-box oracle in a sample-efficient way. While prior studies focus on forward approaches to learn surrogates for the unknown objective function, they struggle with steering clear of out-of-distribution and invalid inputs. Recently, inverse modeling approaches that map objective space to the design space with conditional diffusion models have demonstrated impressive capability in learning the data manifold. They have shown promising performance in offline BBO tasks. However, these approaches require a pre-collected dataset. How to design the acquisition function for inverse modeling to *actively* query new data remains an open question. In this work, we propose *diffusion-based inverse modeling for black-box optimization* (Diff-BBO), an *inverse* approach leveraging diffusion models for online BBO problem. Instead of proposing candidates in the design space, Diff-BBO employs a novel acquisition function *Uncertainty-aware Exploration* (UaE) to propose objective function values. Subsequently, we employ a conditional diffusion model to generate samples based on these proposed values within the design space. We demonstrate that using UaE results in optimal optimization outcomes, supported by both theoretical and empirical evidence.

1 Introduction

Practical problems in science and engineering often involve optimizing a black-box objective function that is expensive to evaluate, such as robotics (Tesch et al., 2013) and molecular design (Sanchez-Lengeling and Aspuru-Guzik, 2018). How to achieve a near-optimal solution while minimizing function evaluations is thus a major challenge in black-box optimization (BBO). Prior works in BBO have largely focused on the online setting where a model can iteratively query the function during training (Turner et al., 2021; Zhang et al., 2021; Hebbal et al., 2019; Mockus, 1974). Most existing algorithms belong to forward methods, including Bayesian optimization (BO) (Kushner, 1964; Mockus, 1974; Wu et al., 2023; Frazier, 2018), bandit algorithms (Agrawal and Goyal, 2012;

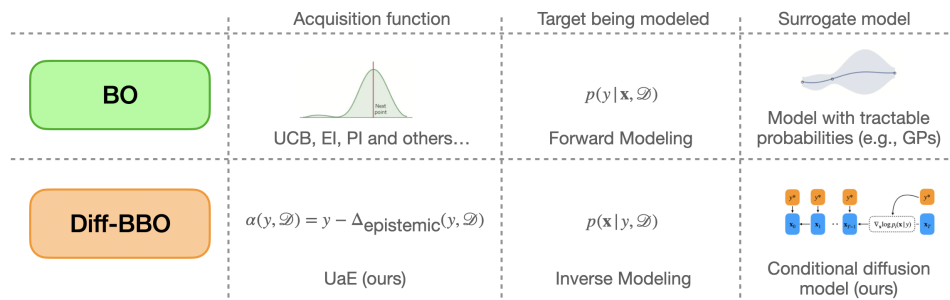


Figure 1: Forward modeling vs inverse modeling for black-box optimization. (*Top*) Forward modeling approach using certain surrogate models (e.g., GPs) for forward modeling and acquisition functions (e.g., UCB, PI, and EI) to select \mathbf{x} . (*Bottom*) Our inverse modeling approach using generative model (e.g., diffusion model) for inverse modeling and acquisition function (e.g., UaE) to select y .

Karbasi et al., 2023), and conditional sampling approaches (Brookes et al., 2019; Gruver et al., 2024; Stanton et al., 2022). They build a surrogate model to approximate the black-box function and optimize sequentially.

However, these approaches may face difficulties in scenarios where valid inputs represent a small subspace, such as valid protein sequences or molecular structures. Such optimization problems become exceptionally challenging, as the optimizer must navigate and avoid out-of-distribution and invalid inputs (Kumar and Levine, 2020). Recently, a novel set of methods, termed *inverse approaches*, have been proposed to address this issue. These methods (Kumar and Levine, 2020; Krishnamoorthy et al., 2023; Kim et al., 2023; Fu and Levine, 2021) break the traditional paradigm by learning an inverse mapping from objective space back to the input (design) space. Leveraging the state-of-the-art generative models, such as diffusion models (Sohl-Dickstein et al., 2015; Song et al., 2020), these approaches effectively capture data distributions in high-dimensional input space and facilitate optimization within the data manifold (Kong et al., 2024; Li et al., 2024). They achieve high performance in offline optimization settings (Kumar and Levine, 2020; Lu et al., 2023; Wang et al., 2018), assuming access to a fixed pre-collected dataset.

Despite these advancements, the online setting of inverse modeling, particularly how to capture the uncertainty of the inverse model and design an acquisition function for data-efficient querying, remains an open question. In this paper, we propose Diff-BBO, an inverse approach for online black-box optimization. Our approach consists of a novel acquisition function designed through the *uncertainty quantification* (UQ) of conditional diffusion model, which proposes the desired objective function values to strategically sample the design space and query the oracle function efficiently. We summarize our main contributions as follows:

- We present Diff-BBO, an inverse modeling approach for efficient online black-box optimization (BBO) leveraging uncertainty of conditional diffusion models.
- We design a novel acquisition function for BBO based on the uncertainty of conditional diffusion models. Theoretically, we prove that the balance between targeting higher objective values and minimizing epistemic uncertainty lead to optimal optimization outcomes.
- We demonstrate that Diff-BBO achieves state-of-the-art performance with superior sample efficiency on Design-Bench and molecular discovery task in the online BBO setting.

2 Methodology

Let $f : \mathcal{X} \rightarrow \mathbb{R}$ denote the unknown ground-truth black-box function that evaluates the quality of any data point \mathbf{x} , with $\mathcal{X} \subseteq \mathbb{R}^d$. Our goal is to find the optimal point \mathbf{x}^* that maximizes f , $\mathbf{x}^* \in \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$. In the batch online BBO setting, we iteratively query f with batch size N and a fixed query iteration K and update the model based on observed outputs. At each iteration, the acquisition function guides the data selection of new query points by balancing exploration and exploitation. In Diff-BBO, we model the conditional distribution of $p(\mathbf{x}|y, \mathcal{D})$ with training data \mathcal{D} . The function value y to condition on is proposed by an acquisition function, which quantifies the

quality of the generated \mathbf{x} . The objective of the above optimization becomes:

$$\max_{y_k \in \mathbb{R}} \sum_{k=1}^K f(\mathbf{x}_k), \quad \mathbf{x}_k \sim p_\theta(\cdot | y_k, \mathcal{D}), \quad \theta \in \Theta. \quad (1)$$

To solve this optimization problem, we introduce Diff-BBO algorithm in [Algorithm 1](#). At each iteration k , we train a conditional diffusion model and compute the optimal y_k^* with the designed acquisition function. In practice, y is selected from a constructed candidate set \mathcal{Y} based on the acquisition function scores $\alpha(y)$. The range of \mathcal{Y} is determined by $w \cdot \phi_k$, where w is a positive scalar and ϕ_k is the maximum function values being queried in the current training dataset \mathcal{D} . Conditioning on y_k^* , we generate N samples $\{\mathbf{x}_j\}_{j=1}^N$, where $\mathbf{x}_j \sim p_\theta(\mathbf{x} | y_k^*, \mathcal{D})$. By querying the oracle to evaluate \mathbf{x}_j , we obtain the best possible reconstructed value ϕ_k for the current iteration, and append all queried data pairs $\{\mathbf{x}_j, f(\mathbf{x}_j)\}_{j=1}^N$ to the training dataset \mathcal{D} . [Figure 1](#) summarizes the difference between the prior forward BBO methods and our proposed inverse modeling approach.

2.1 Acquisition Function Design

In this section, we analyze the uncertainty of Diff-BBO, decomposing its uncertainty into aleatoric and epistemic uncertainty. Then we propose an acquisition function called Uncertainty-aware Exploration (UaE) based on it. We prove that by achieving a balance between high objective values and low epistemic uncertainty, UaE provides a near-optimal solution to the online BBO problem.

Uncertainty Decomposition. We resort to the tools of Bayesian inference to solve the optimization problem defined in [Equation \(1\)](#). Given an observed value y of a sample \mathbf{x} , the objective of Bayesian inference is to estimate the predictive distribution:

$$p(\mathbf{x} | y, \mathcal{D}) = \mathbb{E}_\theta[p_\theta(\mathbf{x} | y)] = \int_\theta p_\theta(\mathbf{x} | y)p(\theta | \mathcal{D})d\theta. \quad (2)$$

By [Equation \(2\)](#), we recognize that the uncertainty arises from two sources: uncertainty in deciding parameter θ from its posterior $p(\theta | \mathcal{D})$ and uncertainty in generating sample \mathbf{x} from a fixed diffusion model $p_\theta(\mathbf{x} | y)$ after θ is chosen. We further provide a decomposition in terms of the aleatoric uncertainty and its epistemic counterpart.

To estimate the aleatoric uncertainty, we can Monte Carlo (MC) sample \mathbf{x} for N times from a learned likelihood function $p_\theta(\mathbf{x} | y)$ for fixed y, θ . To estimate the epistemic uncertainty, we use ensemble techniques. During the inference time, by initializing the trained ensemble models with different random seeds, we first sample M model parameters $\{\theta_i\}_{i=1}^M$ to simulate M conditional diffusion models. Then we generate N samples $\{\mathbf{x}_j\}_{j=1}^N$ for each diffusion model with corresponding parameter $\theta_i, \forall i \in [M]$. Combining the above gives a practical way to decompose and estimate the two types of uncertainty, which is formally described in [Proposition 1](#).

Proposition 1 (Uncertainty Decomposition). *At each iteration $k \in [K]$, the overall uncertainty in inverse modeling can be split into aleatoric and epistemic components, measured empirically as:*

$$\begin{aligned} \Delta_{\text{aleatoric}}(y, \mathcal{D}) &= \mathbb{E}_{\theta_i \sim p(\cdot | \mathcal{D})} \left[\text{Var}_{\mathbf{x}_{i,j} \sim p_{\theta_i}(\cdot | y)} (\|\mathbf{x}_{i,j}\|) \right], \quad \forall i \in [M], j \in [N]; \\ \Delta_{\text{epistemic}}(y, \mathcal{D}) &= \text{Var}_{\theta_i \sim p(\cdot | \mathcal{D})} \left(\mathbb{E}_{\mathbf{x}_{i,j} \sim p_{\theta_i}(\cdot | y)} [\|\mathbf{x}_{i,j}\|] \right), \quad \forall i \in [M], j \in [N]. \end{aligned} \quad (3)$$

Uncertainty-aware Exploration. At each iteration $k \in [K]$ of Dif-BBO algorithm, the acquisition function proposes an optimal scalar value y_k^* as follows: $y_k^* = \text{argmax}_y \alpha(y, \mathcal{D})$, which is used to generate \mathbf{x} in the design space using conditional diffusion model.

Note that to design an effective acquisition function for inverse modeling, we need to achieve a balance between high objective values y and low epistemic uncertainty. On the one hand, it is advantageous to focus on the regions in \mathcal{X} whose corresponding y is of high values. As function evaluations are expensive to perform, we prefer to generate samples \mathbf{x} conditioned on higher y , and only query the oracle for such promising samples to solve the black-box optimization task. On the other hand, we employ the epistemic uncertainty to gauge the error in the trained diffusion model. Specifically, it helps reduce the approximation error between y_k^* and the reconstructed function value $\max_{j \in [N]} f(\mathbf{x}_j)$, where $f(\cdot)$ is the black-box oracle, and $\mathbf{x}_j \sim p_\theta(\cdot | y_k^*, \mathcal{D}), \forall j \in [N]$.

We introduce the *Uncertainty-aware Exploration* (UaE) as our designed acquisition function:

$$\alpha(y, \mathcal{D}) = y - \Delta_{\text{epistemic}}(y, \mathcal{D}), \tag{4}$$

which utilizes the uncertainty estimation on conditional diffusion model as in [Proposition 1](#). By balancing the exploration-exploitation trade-off, UaE effectively solve the online BBO problem.

Performance Analyses of UaE. We prove in [Theorem 1](#) that by adopting UaE for inverse modeling to guide the selection of generated samples for solving BBO problems, we can obtain a near-optimal solution for the optimization problem defined in [Equation \(1\)](#). Detailed discussions and proofs can be found in [Appendix C](#), [Appendix D](#), and [Appendix E](#).

Theorem 1. *Let \mathcal{Y} be the constructed candidate set at each iteration $k \in [K]$ in [Algorithm 1](#). By adopting UaE as the acquisition function to guide the sample generation process in conditional diffusion model, Diff-BBO ([Algorithm 1](#)) achieves a near-optimal solution for the online BBO problem defined in [Equation \(1\)](#):*

$$\max_{y_k \in \mathbb{R}} \sum_{k=1}^K f(\mathbf{x}_k), \quad \mathbf{x}_k \sim p_{\theta}(\cdot | y_k, \mathcal{D}), \quad \theta \in \Theta \Rightarrow \max_{y_k \in \mathcal{Y}} \sum_{k=1}^K \alpha(y_k, \mathcal{D}).$$

As a result, equipped with the novel design of UaE, Diff-BBO is a theoretically sound approach utilizing inverse modeling to effectively solve the online BBO problem.

3 Experiments

To validate the efficacy of Diff-BBO, we conduct experiments on six real-world online black-box optimization tasks for both continuous and discrete optimization tasks.

Dataset. We restructured 5 real-world tasks from Design-Bench including 3 continuous and 2 discrete tasks. In **D’Kitty** and **Ant Morphology**, the goal is to optimize for the morphology of robots. In **Superconductor**, the aim is to optimize superconducting material with a high critical temperature. **TFBind8** and **TFBind10** are discrete tasks to find a DNA sequence with maximum affinity to bind with a specified transcription factor. We also include a **Molecular Discovery** task to optimize compound’s activity against a biological target with therapeutic value. For each task, we arrange the offline dataset from [Krishnamoorthy et al. \(2023\)](#) in ascending order based on objective values and select data from the 25th to the 50th percentile as the initial training dataset. We prioritize data with lower objective scores to better observe performance differences across each baseline. Each optimization iteration is allocated 100 queries to the oracle function (batch size $N = 100$), with a total of 16 iterations conducted. More details of the dataset are provided in [Appendix F.1](#).

Baselines. We compare Diff-BBO with 10 baselines, including Bayesian optimization (BO), trust region BO (TurBO) ([Eriksson et al., 2019](#)), local latent space Bayesian optimization (LOL-BO) ([Maus et al., 2022](#)), likelihood-free BO (LFBO) ([Song et al., 2022](#)), evolutionary algorithms ([Brindle, 1980](#); [Real et al., 2019](#)), conditioning by adaptive sampling (CbAS) ([Brookes et al., 2019](#)), and random sampling. For BO approaches, we include Gaussian Processes (GP) with Monte Carlo (MC)-based batch expected improvement (EI), MC-based batch upper confidence bound (UCB) ([Wilson et al., 2017](#)), and joint entropy search (JES ([Hvarfner et al., 2022](#)) as the acquisition functions. For LFBO, we use EI and probability of improvement (PI) as the acquisition functions.

Results. [Figure 2](#) illustrates the performance across six datasets for all baselines and our proposed algorithm. Notably, Diff-BBO consistently outperforms other baselines in both discrete and continuous settings. Specifically, in the Ant and Dkitty tasks, Diff-BBO demonstrates a significant lead over all baseline methods, starting from the very first iteration of the online optimization process. This remarkable performance can be attributed to Diff-BBO’s diffusion model-based inverse modeling approach, which effectively learns the data manifold in the design space from the initial dataset, even when the initial dataset lacks data with high objective function values.

4 Conclusion

In this paper, we introduced Diff-BBO, an inverse modeling approach for online black-box optimization that leverages the uncertainty of conditional diffusion models. By utilizing the novel acquisition function UaE, Diff-BBO strategically proposes objective function values to improve sample efficiency. We did extensive empirical evaluations to show the superior performance of Diff-BBO. Theoretically, we prove that using UaE leads to optimal optimization solutions.

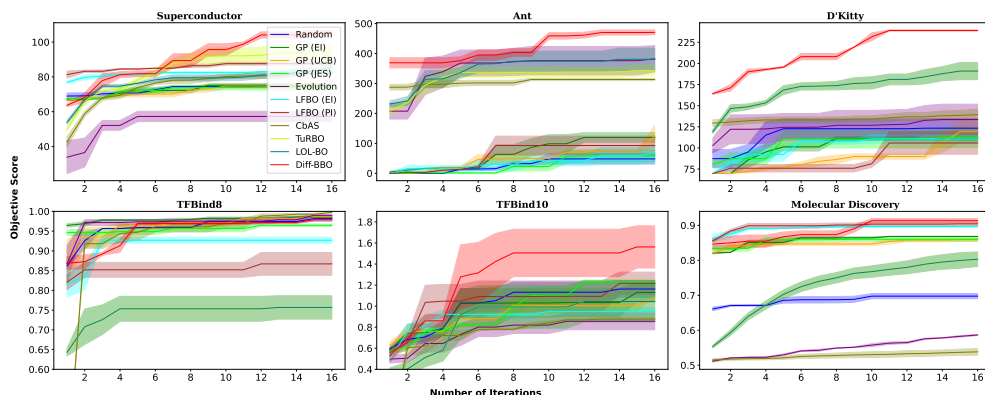


Figure 2: Comparison of Diff-BBO with baselines for online black-box optimization. We plot the mean and standard deviation across three random seeds.

References

- Agrawal, S. and Goyal, N. (2012). Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1. JMLR Workshop and Conference Proceedings.
- Brindle, A. (1980). Genetic algorithms for function optimization.
- Brookes, D., Park, H., and Listgarten, J. (2019). Conditioning by adaptive sampling for robust design. In *International conference on machine learning*, pages 773–782. PMLR.
- Eckmann, P., Sun, K., Zhao, B., Feng, M., Gilson, M. K., and Yu, R. (2022). Limo: Latent inceptionism for targeted molecule generation. *arXiv preprint arXiv:2206.09010*.
- Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. (2019). Scalable global optimization via local bayesian optimization. *Advances in neural information processing systems*, 32.
- Frazier, P. I. (2018). A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*.
- Fu, J. and Levine, S. (2021). Offline model-based optimization via normalized maximum likelihood estimation. *arXiv preprint arXiv:2102.07970*.
- Gruver, N., Stanton, S., Frey, N., Rudner, T. G., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., and Wilson, A. G. (2024). Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36.
- Hebbal, A., Brevault, L., Balesdent, M., Talbi, E.-G., and Melab, N. (2019). Bayesian optimization using deep gaussian processes. *arXiv preprint arXiv:1905.03350*.
- Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.
- Ho, J. and Salimans, T. (2022). Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.
- Hvarfner, C., Hutter, F., and Nardi, L. (2022). Joint entropy search for maximally-informed bayesian optimization. *Advances in Neural Information Processing Systems*, 35:11494–11506.
- Jeon, W. and Kim, D. (2020). Autonomous molecule generation using reinforcement learning and docking to develop potential novel inhibitors. *Scientific reports*, 10(1):22104.
- Karbasi, A., Kuang, N. L., Ma, Y., and Mitra, S. (2023). Langevin thompson sampling with logarithmic communication: bandits and reinforcement learning. In *International Conference on Machine Learning*, pages 15828–15860. PMLR.
- Kim, M., Berto, F., Ahn, S., and Park, J. (2023). Bootstrapped training of score-conditioned generator for offline design of biological sequences. *Advances in Neural Information Processing Systems*, 37.

- Kong, L., Du, Y., Mu, W., Neklyudov, K., De Bortol, V., Wang, H., Wu, D., Ferber, A., Ma, Y.-A., Gomes, C. P., et al. (2024). Diffusion models as constrained samplers for optimization with unknown constraints. *arXiv preprint arXiv:2402.18012*.
- Krishnamoorthy, S., Mashkaria, S. M., and Grover, A. (2023). Diffusion models for black-box optimization. In *International Conference on Machine Learning*, pages 17842–17857. PMLR.
- Kumar, A. and Levine, S. (2020). Model inversion networks for model-based optimization. *Advances in neural information processing systems*, 33:5126–5137.
- Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise.
- Lee, S., Jo, J., and Hwang, S. J. (2023). Exploring chemical space with score-based out-of-distribution generation. In *International Conference on Machine Learning*, pages 18872–18892. PMLR.
- Li, Z., Yuan, H., Huang, K., Ni, C., Ye, Y., Chen, M., and Wang, M. (2024). Diffusion model for data-driven black-box optimization. *arXiv preprint arXiv:2403.13219*.
- Lu, H., Qian, H., Wu, Y., Liu, Z., Zhang, Y.-L., Zhou, A., and Yu, Y. (2023). Degradation-resistant offline optimization via accumulative risk control. In *ECAI 2023*, pages 1609–1616. IOS Press.
- Maus, N., Jones, H., Moore, J., Kusner, M. J., Bradshaw, J., and Gardner, J. (2022). Local latent space bayesian optimization over structured inputs. *Advances in neural information processing systems*, 35:34505–34518.
- Mockus, J. (1974). On bayesian methods for seeking the extremum. In *Proceedings of the IFIP Technical Conference*, pages 400–404.
- Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S., and Olson, A. J. (2009). Autodock4 and autodocktools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry*, 30(16):2785–2791.
- Noh, J., Jeong, D.-W., Kim, K., Han, S., Lee, M., Lee, H., and Jung, Y. (2022). Path-aware and structure-preserving generation of synthetically accessible molecules. In *International Conference on Machine Learning*, pages 16952–16968. PMLR.
- Real, E., Aggarwal, A., Huang, Y., and Le, Q. V. (2019). Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789.
- Sanchez-Lengeling, B. and Aspuru-Guzik, A. (2018). Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365.
- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.
- Song, J., Yu, L., Neiswanger, W., and Ermon, S. (2022). A general recipe for likelihood-free bayesian optimization. In *International Conference on Machine Learning*, pages 20384–20404. PMLR.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.
- Stanton, S., Maddox, W., Gruver, N., Maffettone, P., Delaney, E., Greenside, P., and Wilson, A. G. (2022). Accelerating bayesian optimization for biological sequence design with denoising autoencoders. In *International Conference on Machine Learning*, pages 20459–20478. PMLR.
- Tesch, M., Schneider, J., and Choset, H. (2013). Expensive multiobjective optimization for robotics. In *2013 IEEE international conference on robotics and automation*, pages 973–980. IEEE.
- Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE.
- Trabucco, B., Geng, X., Kumar, A., and Levine, S. (2022). Design-bench: Benchmarks for data-driven offline model-based optimization. In *International Conference on Machine Learning*, pages 21658–21676. PMLR.

- Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., and Guyon, I. (2021). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, pages 3–26. PMLR.
- Wang, H., Jin, Y., Sun, C., and Doherty, J. (2018). Offline data-driven evolutionary optimization using selective surrogate ensembles. *IEEE Transactions on Evolutionary Computation*, 23(2):203–216.
- Wilson, J. T., Moriconi, R., Hutter, F., and Deisenroth, M. P. (2017). The reparameterization trick for acquisition functions. *arXiv preprint arXiv:1712.00424*.
- Wu, D., Niu, R., Chinazzi, M., Ma, Y., and Yu, R. (2023). Disentangled multi-fidelity deep bayesian active learning. In *International Conference on Machine Learning*, pages 37624–37634. PMLR.
- Zhang, D., Fu, J., Bengio, Y., and Courville, A. (2021). Unifying likelihood-free inference with black-box optimization and beyond. *arXiv preprint arXiv:2110.03372*.

Appendices

A Diff-BBO algorithm

Algorithm 1: Diff-BBO

Input: Initial dataset $\mathcal{D} = \{\mathbf{x}, y\}$, total number of iterations K , candidate feasible range C , oracle function $f(\cdot)$, batch size N

- 1 **Initialization:** Conditional diffusion model $p_\theta(\mathbf{x}|y)$
- 2 **for** $k = 1, 2, \dots, K$ **do**
- 3 Train the conditional diffusion model with \mathcal{D}
- 4 Construct a candidate set $\mathcal{Y} = \{y : 0 \leq y \leq C\}$
- 5 $y_k^* = \operatorname{argmax}_{y \in \mathcal{Y}} \alpha(y, \mathcal{D})$
- 6 Generate $\{\mathbf{x}_j\}_{j=1}^N$ where $\mathbf{x}_j \sim p_\theta(\mathbf{x} | y_k^*, \mathcal{D})$
- 7 Query the oracle function $f(\cdot)$ with generated samples $\{\mathbf{x}_j\}_{j=1}^N$
- 8 $\mathcal{D} \leftarrow \mathcal{D} \cup \{\mathbf{x}_j, f(\mathbf{x}_j)\}_{j=1}^N$
- 9 $\phi_k \leftarrow \max(f(x))$ s.t. $x \in \mathcal{D}$

Output: Reconstructed $\{\phi_k\}_{k=1}^K$

B Conditional Diffusion Model Training

Diffusion Models (Sohl-Dickstein et al., 2015; Song et al., 2020) are probabilistic generative models that learn distributions through an iterative denoising process. These models consist of three components: a forward diffusion process that produces a series of noisy samples by adding Gaussian noise, a reverse process to reconstruct the original data samples from the noise, and a sampling procedure to generate new data samples from the learned distribution. Let the original sample be \mathbf{x}_0 and t be the diffusion step. For conditional diffusion models, a conditional variable y is added to both the forward process as $q(\mathbf{x}_t | \mathbf{x}_{t-1}, y)$ and reverse process as $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, y)$, $\forall t \in [T]$.

The reverse process begins with the standard Gaussian distribution $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and denoises \mathbf{x}_t to recover \mathbf{x}_0 through the following Markov chain with reverse transitions:

$$p_\theta(\mathbf{x}_{0:T} | y) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, y), \quad \mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, y) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t, y), \Sigma_\theta(\mathbf{x}_t, t, y)).$$

During training, Σ_θ is empirically fixed, and μ_θ is reparametrized by a trainable denoise function $\epsilon_\theta(\mathbf{x}_t, t, y)$, which is used to estimate the noise vector ϵ that was added to input \mathbf{x}_t , and is trained by minimizing a reweighted version of the evidence lower bound (ELBO):

$$\mathcal{L}_{\text{dif}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}), y, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(0, T), \mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_0, y)} \left[w(t) \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, y)\|_2^2 \right]. \quad (5)$$

Note that the loss in Equation (5) (Ho et al., 2020) for ϵ_θ is denoising score matching for all time step t , which estimates the gradient of the log probability density of the noisy data (a.k.a. score function): $\epsilon_\theta(\mathbf{x}_t, t, y) \approx -\sigma_t \nabla_{\mathbf{x}} \log p(\mathbf{x} | y)$. We further denote the score function as $s_\theta(\mathbf{x}_t, y, t) := -\epsilon_\theta(\mathbf{x}_t, t, y) / \sigma_t$.

Instead of learning a fixed deterministic θ from a deterministic neural network, we are interested in learning its Bayesian posterior to further understand and improve the model’s performance as well as its reliability with uncertainty quantification. In Bayesian settings, we consider the model parameters $\theta \in \Theta$, where Θ is the parameter space, and maintain its posterior distribution $p(\theta | \mathcal{D})$, which is learned from training data \mathcal{D} . By choosing θ from its posterior, essentially we sample a score function $\tilde{s}_\theta(\mathbf{x}_t, y, t)$ from the probability distribution $p(s_\theta | \mathbf{x}_t, y, t, \mathcal{D}) = \mathcal{N}(s_\theta(\mathbf{x}_t, y, t), \Sigma_{s_\theta}(\mathbf{x}_t, y, t))$, whose expected value is $s_\theta(\mathbf{x}_t, y, t)$, and variance is a diagonal covariance matrix $\Sigma_{s_\theta}(\mathbf{x}_t, y, t)$.

Specifically, we adopt classifier-free guidance as in (Ho and Salimans, 2022) to eliminate the requirement of training a separate classifier model. We jointly train an unconditional diffusion model $p_\theta(\mathbf{x})$ parameterized by $\epsilon_\theta(\mathbf{x}, t, \emptyset)$ and a conditional diffusion model $p_\theta(\mathbf{x}|y)$ parameterized by $\epsilon_\theta(\mathbf{x}, t, y)$ by minimizing the following loss function:

$$\mathcal{L}_{\text{cdif}} = \mathbb{E}_{\mathbf{x}_0, y, \epsilon, t, \mathbf{x}_t, \lambda} \left[w(t) \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, (1 - \lambda)y + \lambda\emptyset)\|_2^2 \right], \quad (6)$$

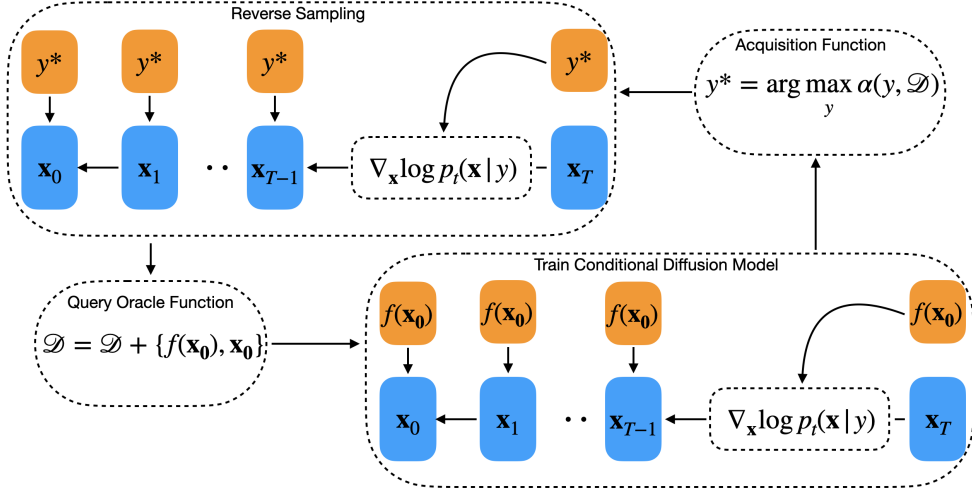


Figure 3: Black-box optimization framework using the conditional diffusion model as the inverse model. The overall framework includes 4 stages. 1. Train the conditional diffusion model given the current training dataset. 2. Compute the acquisition function and select the optimal y^* to condition on. 3. Generate samples $\{\mathbf{x}_0\}$ conditioned on y^* . 4. Query the oracle given generated samples $\{\mathbf{x}_0\}$ and update the training dataset.

where $\mathbf{x}_0 \sim q(\mathbf{x})$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, $t \sim \mathcal{U}(0, T)$, $\mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_0)$, $\lambda \sim \text{Bernoulli}(p_{\text{uncond}})$, and p_{uncond} is the probability of setting y to the unconditional information \emptyset . The overall Diff-BBO framework with the conditional diffusion model included is shown in Figure 3.

C Uncertainty Quantification on Conditional Diffusion Model

Here, let us first consider the problem of how to capture the uncertainty for a fixed diffusion model. In fact, the uncertainty in generating \mathbf{x} can be explicitly traced through the denoising process. More specifically, Theorem 2 provides analytical solutions to compute the uncertainty on a single denoising process of general score-based conditional diffusional models. It provides theoretical insights of how uncertainty is being propagated through the reverse denoising process both in discrete time and continuous time, which is characterized through the lens of stochastic differential equations (SDEs) of the Ornstein–Uhlenbeck (OU) process.

Theorem 2. (*Uncertainty propagation*) Let $t \in [T]$ be the diffusion step, $s_\theta(\mathbf{x}, y, t)$ be the score function of the corresponding diffusion model $p_\theta(\mathbf{x} | y)$. For a single conditional diffusional model $p_\theta(\mathbf{x} | y)$, the uncertainty in generating a sample \mathbf{x} can be analytically traced through the discrete-time reverse denoising process as follows:

$$\begin{aligned} \text{Var}(\mathbf{x}_{t-1}) &= \frac{1}{4} \text{Var}(\mathbf{x}_t) + \text{Var}(s_\theta(\mathbf{x}, y, t)) + \frac{1}{2} (\mathbb{E}[\mathbf{x}_t \circ s_\theta(\mathbf{x}, y, t)] - \mathbb{E}[\mathbf{x}_t] \circ \mathbb{E}[s_\theta(\mathbf{x}, y, t)]) + \mathbf{I}, \\ \mathbb{E}(\mathbf{x}_{t-1}) &= \frac{1}{2} \mathbb{E}(\mathbf{x}_t) + \mathbb{E}(s_\theta(\mathbf{x}, y, t)), \end{aligned}$$

where \circ is the Hadamard product, and \mathbf{I} is the identity matrix. Similarly, in continuous-time process, the uncertainty can be captured as follows:

$$\text{Var}(\mathbf{x}_0) = (T + 1)\mathbf{I} + \text{Var} \left(\int_{t=0}^T \left(\frac{1}{2} \mathbf{x}_t + s_\theta(\mathbf{x}, y, t) \right) dt \right). \quad (7)$$

Nevertheless, performing exact Bayesian inference for uncertainty quantification when training diffusion models requires non-trivial efforts and can be computationally demanding. Hence, we introduce a practically-efficient uncertainty decomposition based on Equation (2).

C.1 Conditional Diffusion SDE

It can be shown that the conditional diffusion model can be represented by the Ornstein–Uhlenbeck (OU) process, which is a time-homogeneous continuous-time Markov process:

$$d\mathbf{x}_t = -\gamma \mathbf{x}_t dt + \sigma d\mathbf{w}_t, \quad (8)$$

where γ is the relaxation rate, σ is the strength of fluctuation, and \mathbf{w}_t is the standard Wiener process (a.k.a., Brownian motion). Both γ and σ are time-invariant. In particular, setting $\gamma = 1$ and $\sigma = \sqrt{2}$, we are able to establish that Denoising Diffusion Probabilistic Model (DDPM) is equivalent to OU process observed at discrete times. In the remaining text, we consider SDEs for general score-based diffusion models. The SDE of the forward process in conditional diffusion model can then be written as:

$$d\mathbf{x}_t = -\frac{1}{2}g(t)\mathbf{x}_t dt + \sqrt{g(t)} d\mathbf{w}_t, \quad \mathbf{x}_0 \sim q(\mathbf{x}|y) \quad (9)$$

where $g(t)$ is a nondecreasing weighting function that controls the speed of diffusion in the forward process and $g(t) > 0$. For simplicity of analysis, we fix $g(t) = 1$ for all $t \in [T]$.

The generation process of a conditional score-based diffusion model can be viewed as a particular discretization of the following reverse-time SDE:

$$d\mathbf{x}_t = \left(\frac{1}{2}\mathbf{x}_t - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y) \right) dt + d\mathbf{w}_t, \quad \mathbf{x}_0 \sim p(\mathbf{x}_T|y). \quad (10)$$

In practice, the unknown ground truth conditional score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y)$ needs to be estimated with score networks. Let such estimator denoted by $s_\theta(\mathbf{x}, y, t)$, then the conditional sample generation is to simulate the following backward SDE:

$$d\mathbf{x}_t = \left(\frac{1}{2}\mathbf{x}_t - s_\theta(\mathbf{x}, y, t) \right) dt + d\mathbf{w}_t, \quad \mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (11)$$

In Bayesian settings, we sample a score function $\tilde{s}_\theta(\mathbf{x}_t, y, t)$ from the probability distribution $p(s_\theta|\mathbf{x}_t, y, t, \mathcal{D}) = \mathcal{N}(s_\theta(\mathbf{x}_t, y, t), \Sigma_\theta(\mathbf{x}_t, y, t))$ with expected value $s_\theta(\mathbf{x}_t, y, t)$, and diagonal covariance $\Sigma_\theta(\mathbf{x}_t, y, t)$.

C.2 Estimation of Uncertainty

In this section, we quantify the uncertainty of a single conditional diffusion model in both discrete-time and continuous-time reverse process for [Theorem 2](#).

C.2.1 Uncertainty in Discrete-time Reverse Process

We first proof the first statement of [Theorem 2](#). We consider the Euler discretization of [Equation \(11\)](#), which leads to:

$$\mathbf{x}_{t-1} = \frac{1}{2}\mathbf{x}_t + s_\theta(\mathbf{x}, y, t) + \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (12)$$

We thus have,

$$\text{Var}(\mathbf{x}_{t-1}) = \frac{1}{4}\text{Var}(\mathbf{x}_t) + \text{Var}(s_\theta(\mathbf{x}, y, t)) + \frac{1}{2}\text{Cov}(\mathbf{x}_t, s_\theta(\mathbf{x}, y, t)) + \mathbf{I}. \quad (13)$$

$$\mathbb{E}(\mathbf{x}_{t-1}) = \frac{1}{2}\mathbb{E}(\mathbf{x}_t) + \mathbb{E}(s_\theta(\mathbf{x}, y, t)). \quad (14)$$

Here $\text{Cov}(\mathbf{x}_t, s_\theta(\mathbf{x}, y, t))$ is the element-wise covariance between \mathbf{x}_t and $s_\theta(\mathbf{x}, y, t)$. Note that we only need to consider the correlation between \mathbf{x}_t and $s_\theta(\mathbf{x}, y, t)$ at the same time step. As a result, to estimate $\text{Cov}(\mathbf{x}_t, s_\theta(\mathbf{x}, y, t))$, we have,

$$\begin{aligned} \text{Cov}(\mathbf{x}_t, s_\theta(\mathbf{x}, y, t)) &= \mathbb{E} \left[(\mathbf{x}_t - \mathbb{E}[\mathbf{x}_t]) (s_\theta(\mathbf{x}, y, t) - \mathbb{E}[s_\theta(\mathbf{x}, y, t)])^\top \right] \\ &= \mathbb{E}[\mathbf{x}_t \circ s_\theta(\mathbf{x}, y, t)] - \mathbb{E}[\mathbf{x}_t] \circ \mathbb{E}[s_\theta(\mathbf{x}, y, t)] \\ &= \mathbb{E}_{\mathbf{x}_t}[\mathbf{x}_t \circ s_\theta(\mathbf{x}, y, t)] - \mathbb{E}[\mathbf{x}_t] \circ \mathbb{E}_{\mathbf{x}_t}[s_\theta(\mathbf{x}_t, y, t)] \end{aligned} \quad (15)$$

where \circ is the Hadamard product and the third equality is by tower's rule. Substituting [Equation \(15\)](#) back to [Equation \(13\)](#) completes the proof of the first part of [Theorem 2](#).

C.2.2 Uncertainty in Continuous-time Reverse Process

We now proof the second statement of [Theorem 2](#). To perform the uncertainty quantification for the continuous-time reverse process, we posit the following assumption.

Assumption 1. For valid $t \in [0, T]$, the generating process \mathbf{x}_t in [Equation \(10\)](#) is integrable and has finite second-order moments.

With [Assumption 1](#), integrating [Equation \(10\)](#) with respect to t yields:

$$\mathbf{x}_0 = \mathbf{x}_T - \int_{t=0}^T \left(\frac{1}{2} \mathbf{x}_t + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y) \right) dt + \int_{t=0}^T d\mathbf{w}_t. \quad (16)$$

Applying the variance operator to both sides of

$$\begin{aligned} \text{Var}(\mathbf{x}_0) &= \text{Var}(\mathbf{x}_T) + \text{Var} \left(\int_{t=0}^T \left(\frac{1}{2} \mathbf{x}_t + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y) \right) dt \right) + \text{Var} \left(\int_{t=0}^T d\mathbf{w}_t \right) \\ &= I + \text{Var} \left(\int_{t=0}^T \left(\frac{1}{2} \mathbf{x}_t + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y) \right) dt \right) + \mathbb{E} \left[\left(\int_{t=0}^T d\mathbf{w}_t \right)^2 \right] - \left(\mathbb{E} \left[\int_{t=0}^T d\mathbf{w}_t \right] \right)^2 \\ &= (T+1)I + \underbrace{\text{Var} \left(\int_{t=0}^T \left(\frac{1}{2} \mathbf{x}_t + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y) \right) dt \right)}_{V_1}, \end{aligned} \quad (17)$$

where the last equality follows the properties of Itô Integral and rules of stochastic calculus such that $(d\mathbf{w})^2 = dt$, $\mathbb{E}[\int_{t=0}^T d\mathbf{w}_t] = 0$. Hence, to provide an uncertainty estimate for \mathbf{x}_0 , it remains to estimate the term V_1 . Recall that the true score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|y)$ is approximated by $s_\theta(\mathbf{x}_t, y, t) = -\epsilon_\theta(\mathbf{x}_t, t, y)/\sigma_t$. For ease of notation, let $s_{\theta,t} = s_\theta(\mathbf{x}_t, y, t)$ and $\tilde{s}_{\theta,t} = \tilde{s}_\theta(\mathbf{x}_t, y, t)$, which gives

$$V_1 = \int_{t=0}^T \int_{s=0}^T \left(\frac{1}{4} \text{Cov}(\mathbf{x}_s, \mathbf{x}_t) - \frac{1}{2} \text{Cov}(\mathbf{x}_s, s_{\theta,t}) - \frac{1}{2} \text{Cov}(\mathbf{x}_t, s_{\theta,s}) + \text{Cov}(s_{\theta,t}, s_{\theta,s}) \right) ds dt.$$

When $s \neq t$, score functions $s_{\theta,t}$ and $s_{\theta,s}$ are independent, and similarly, \mathbf{x}_t and $s_{\theta,s}$ are also independent. As a result, the above equation can be further simplified as

$$V_1 = \int_{t=0}^T \int_{s=0}^T \left(\frac{1}{4} \text{Cov}(\mathbf{x}_s, \mathbf{x}_t) - \frac{1}{2} \text{Cov}(\mathbf{x}_s, s_{\theta,t}) \right) ds dt - \int_{t=0}^T (\text{Cov}(\mathbf{x}_t, s_{\theta,t}) + \text{Cov}(s_{\theta,t}, s_{\theta,t})) dt.$$

Combining all the above results together completes the proof of the second statement of [Theorem 2](#).

D Performance Analysis of UaE

To quantify the quality of generated samples, we theoretically analyze the sub-optimality performance gap between y_k^* and reconstructed value at each iteration. In particular, [Theorem 3](#) and [Theorem 4](#) demonstrate that such sub-optimality gap can be effectively handled in inverse modeling, with proofs deferred to [Appendix D.1](#). We first show that by using conditional diffusion model, the expected error of the sub-optimality performance gap is zero.

Theorem 3. *At each iteration $k \in [K]$, define the sub-optimality performance gap as*

$$\Delta(p_\theta, y_k^*) = \left| y_k^* - \max_{j \in [N]} f(\mathbf{x}_j) \right|, \quad \text{where } \mathbf{x}_j \sim p_\theta(\cdot | y_k^*, \mathcal{D}), \quad \forall j \in [N]. \quad (18)$$

Assume that there exists some $\theta^ \sim p(\theta|\mathcal{D})$ that produces a predictive distribution $p_{\theta^*}(\cdot | \mathcal{D})$ such that it is able to generate a sample \mathbf{x}^* that perfectly reconstructs y_k^* . Suppose function f is L -Lipschitz and each sample is σ -subGaussian, it can be shown that*

$$\mathbb{E}[\Delta(p_\theta, y_k^*)] \leq c_1 L \sqrt{d} \sigma,$$

where c_1 is some universal constant and the empirical estimator $\widehat{\mathbb{E}}[\Delta(p_\theta, y_k^)]$ is unbiased.*

[Theorem 3](#) suggests that in expectation, the reconstructed function value $\max_{j \in [N]} f(\mathbf{x}_j)$ is able to accurately recover the provided conditional information y_k^* . Hence, in order to obtain a reasonable estimator for the optimization problem, the remaining concern goes to the variance of the gap defined in [Equation \(18\)](#), which is further evaluated in [Theorem 4](#).

Theorem 4. (Sub-optimality bound) At each iteration $k \in [K]$, suppose M model parameters $\{\theta_i\}_{i=1}^M$ are generated from the ensemble model for some fixed dataset \mathcal{D} . Suppose function f is L -Lipschitz, it can be shown that the variance of the sub-optimality performance gap of each model is bounded by the epidemic uncertainty:

$$\text{Var}(\Delta(p_{\theta_i}, y_k^*)) \leq c_2 L^2 d \sigma^2 + c_2 L^2 \Delta_{\text{epistemic}}(y_k^*, \mathcal{D}), \quad \forall i \in M, \quad (19)$$

where c_2 is some universal positive constant.

[Theorem 4](#) shows that the variance of the sub-optimality performance gap can be upper bounded by the epistemic uncertainty of diffusion model. Therefore, our proposed acquisition function achieves the balance between high objective value and low epistemic uncertainty.

D.1 Analysis of Sub-optimality

In this section, we study the behavior of the sub-optimality gap of our algorithm by proving [Theorem 3](#) and [Theorem 4](#). We first introduce the notation that is used throughout this section and the next section. Then we present the main lemmas along with their proofs. Finally, we combine the lemmas to prove our main results.

At each iteration $k \in [K]$, let y_k^* be the target function value on which the diffusion model conditions, and p_θ be the model learned by the conditional diffusion model. We define the performance metric for online BBO problem, which measures the sub-optimal performance gap between the function value achieved by sample $\mathbf{x} \sim p_\theta(\cdot | y_k^*, \mathcal{D})$ and the target function value y_k^* . Its formal definition is described as follows:

$$\Delta(p_\theta, y_k^*) = \left| y_k^* - \max_{j \in [N]} f(\mathbf{x}_j) \right|, \quad \text{where } \mathbf{x}_j \sim p_\theta(\cdot | y_k^*, \mathcal{D}), \quad \forall j \in [N]. \quad (20)$$

For simplicity of analysis, we consider $N = 1$, and let the generated sample at the k -th iteration be \mathbf{x}_k in the remaining text. We remark that all proofs go through smoothly for general N with more nuanced notations, and do not affect the conclusions being drawn. To proceed with the proofs in this section, we first state the formal assumptions for the black-box function $f(\cdot)$ and sample \mathbf{x} .

Assumption 2. The scalar black-box function f is L -Lipschitz in \mathbf{x} :

$$|f(\mathbf{x}') - f(\mathbf{x})| \leq L \|\mathbf{x}' - \mathbf{x}\|, \quad \forall \mathbf{x}', \mathbf{x} \in \mathbb{R}^d.$$

Assumption 3. Each generated sample $\mathbf{x} \in \mathbb{R}^d$ is σ -subGaussian. That is, there exists $\sigma \in \mathbb{R}$ such that for any $\mathbf{v} \in \mathbb{R}^d$ with $\|\mathbf{v}\| = 1$, $\mathbf{v}^\top(\mathbf{x} - \mathbb{E}[\mathbf{x}])$ is σ -subGaussian, and its moment generating function is bounded by:

$$\mathbb{E}[\exp(\lambda \mathbf{v}^\top(\mathbf{x} - \mathbb{E}[\mathbf{x}]))] \leq \exp\left(\frac{\sigma^2 \lambda^2}{2}\right), \quad \forall \lambda \in \mathbb{R}, \mathbf{v} \in \mathbb{S}^{d-1},$$

where $\mathbb{S} := \{\mathbf{v} \in \mathbb{R}^d : \|\mathbf{v}\| = 1\}$ is the $(d-1)$ unit sphere.

Before proceeding with the proofs of main theorems, we present our main lemmas.

Lemma D.1. At each iteration $k \in [K]$, under fixed parameters θ and θ^* , for $\mathbf{x}_k \sim p_\theta(\cdot | y_k^*, \mathcal{D})$, $\mathbf{x}^* \sim p_{\theta^*}(\cdot | y_k^*, \mathcal{D})$, we have

$$\mathbb{E}_{\mathbf{x}_k \sim p_\theta(\cdot | y_k^*, \mathcal{D}), \mathbf{x}^* \sim p_{\theta^*}(\cdot | y_k^*, \mathcal{D})} [\|\mathbf{x}^* - \mathbf{x}_k\|] \leq 8\sqrt{d}\sigma + \|\mathbb{E}_{\mathbf{x}^*}[\mathbf{x}^*] - \mathbb{E}_{\mathbf{x}_k}[\mathbf{x}_k]\|, \quad (21)$$

$$\mathbb{E}_{\mathbf{x}_k \sim p_\theta(\cdot | y_k^*, \mathcal{D}), \mathbf{x}^* \sim p_{\theta^*}(\cdot | y_k^*, \mathcal{D})} [\|\mathbf{x}^* - \mathbf{x}_k\|] \geq \|\mathbb{E}_{\mathbf{x}^*}[\mathbf{x}^*] - \mathbb{E}_{\mathbf{x}_k}[\mathbf{x}_k]\|. \quad (22)$$

Proof of Lemma D.1. To bound $\mathbb{E}[\|\mathbf{x}^* - \mathbf{x}_k\|]$, by triangle inequality,

$$\begin{aligned} \mathbb{E}_{\mathbf{x}_k, \mathbf{x}^*} [\|\mathbf{x}^* - \mathbf{x}_k\|] &= \mathbb{E} [\|\mathbf{x}^* - \mathbb{E}[\mathbf{x}^*] + \mathbb{E}[\mathbf{x}_k] - \mathbf{x}_k + \mathbb{E}[\mathbf{x}^*] - \mathbb{E}[\mathbf{x}_k]\|] \\ &\leq \mathbb{E} [\|\mathbf{x}^* - \mathbb{E}[\mathbf{x}^*]\|] + \mathbb{E} [\|\mathbf{x}_k - \mathbb{E}[\mathbf{x}_k]\|] + \mathbb{E} [\|\mathbb{E}[\mathbf{x}^*] - \mathbb{E}[\mathbf{x}_k]\|]. \end{aligned}$$

Under [assumption 3](#), by [Lemma D.3](#), we have,

$$\mathbb{E}_{\mathbf{x}_k, \mathbf{x}^*} [\|\mathbf{x}^* - \mathbf{x}_k\|] \leq 8\sqrt{d}\sigma + \|\mathbb{E}[\mathbf{x}^*] - \mathbb{E}[\mathbf{x}_k]\|.$$

Applying triangle inequality completes the step. In addition, it can be easily seen that

$$\mathbb{E}_{\mathbf{x}_k, \mathbf{x}^*} [\|\mathbf{x}^* - \mathbf{x}_k\|] \geq \|\mathbb{E}[\mathbf{x}^*] - \mathbb{E}[\mathbf{x}_k]\|.$$

□

Lemma D.2. *At each iteration $k \in [K]$, under fixed parameters θ and θ^* , for $\mathbf{x}_k \sim p_\theta(\cdot | y_k^*, \mathcal{D})$, $\mathbf{x}^* \sim p_{\theta^*}(\cdot | y_k^*, \mathcal{D})$, we have*

$$\text{Var}_{\mathbf{x}_k \sim p_\theta(\cdot | y_k^*, \mathcal{D}), \mathbf{x}^* \sim p_{\theta^*}(\cdot | y_k^*, \mathcal{D})}(\|\mathbf{x}^* - \mathbf{x}_k\|) \leq c_3 d \sigma^2. \quad (23)$$

Proof of Lemma D.2. By definition of variance,

$$\text{Var}_{\mathbf{x}_k, \mathbf{x}^*}(\|\mathbf{x}^* - \mathbf{x}_k\|) = \mathbb{E}[\|\mathbf{x}^* - \mathbf{x}_k\|^2] - (\mathbb{E}[\|\mathbf{x}^* - \mathbf{x}_k\|])^2. \quad (24)$$

Expanding the first term leads to

$$\begin{aligned} \mathbb{E}[\|\mathbf{x}^* - \mathbf{x}_k\|^2] &= \mathbb{E}[(\mathbf{x}^* - \mathbf{x}_k)^\top (\mathbf{x}^* - \mathbf{x}_k)] \\ &= \mathbb{E}[\|\mathbf{x}^*\|^2] + \mathbb{E}[\|\mathbf{x}_k\|^2] - 2\mathbb{E}[(\mathbf{x}_k)^\top \mathbf{x}^*] \\ &= \mathbb{E}[\|\mathbf{x}^*\|^2] + \mathbb{E}[\|\mathbf{x}_k\|^2] - 2\mathbb{E}[(\mathbf{x}_k)^\top] \mathbb{E}[\mathbf{x}^*], \end{aligned} \quad (25)$$

where the last equality is due to the independence between \mathbf{x}^* and \mathbf{x}_k .

Under [Assumption 3](#) and by [Lemma D.4](#), we have

$$\begin{aligned} \mathbb{E}[\|\mathbf{x}^*\|^2] &= \mathbb{E}[\|\mathbf{x}^* - \mathbb{E}[\mathbf{x}^*] + \mathbb{E}[\mathbf{x}^*]\|^2] \\ &= \mathbb{E}[(\mathbf{x}^* - \mathbb{E}[\mathbf{x}^*])^\top (\mathbf{x}^* - \mathbb{E}[\mathbf{x}^*])] + \|\mathbb{E}[\mathbf{x}^*]\|^2 \\ &= \text{tr}(\mathbb{E}[(\mathbf{x}^* - \mathbb{E}[\mathbf{x}^*])(\mathbf{x}^* - \mathbb{E}[\mathbf{x}^*])^\top]) + \|\mathbb{E}[\mathbf{x}^*]\|^2 \\ &\leq C d \sigma^2 + \|\mathbb{E}[\mathbf{x}^*]\|^2. \end{aligned}$$

Here, the second equality holds as the cross terms vanish due to the fact that $\mathbb{E}[\mathbf{x}^* - \mathbb{E}[\mathbf{x}^*]] = 0$. Similarly,

$$\mathbb{E}[\|\mathbf{x}_k\|^2] \leq C d \sigma^2 + \|\mathbb{E}[\mathbf{x}_k]\|^2.$$

Substituting the above two results back to [Equation \(25\)](#),

$$\begin{aligned} \mathbb{E}[\|\mathbf{x}^* - \mathbf{x}_k\|^2] &\leq 2C d \sigma^2 + \|\mathbb{E}[\mathbf{x}_k]\|^2 + \|\mathbb{E}[\mathbf{x}^*]\|^2 - 2\mathbb{E}[(\mathbf{x}_k)^\top \mathbf{x}^*] \\ &\leq 2C d \sigma^2 + \|\mathbb{E}[\mathbf{x}_k] - \mathbb{E}[\mathbf{x}^*]\|^2. \end{aligned} \quad (26)$$

Substituting [Equation \(26\)](#) back to [Equation \(24\)](#) and applying [Lemma D.1](#) leads to

$$\text{Var}_{\mathbf{x}_k, \mathbf{x}^*}(\|\mathbf{x}^* - \mathbf{x}_k\|) \leq 2C d \sigma^2 + \|\mathbb{E}[\mathbf{x}_k] - \mathbb{E}[\mathbf{x}^*]\|^2 - (8\sqrt{d}\sigma + \|\mathbb{E}[\mathbf{x}^*] - \mathbb{E}[\mathbf{x}_k]\|)^2 \leq c_3 d \sigma^2. \quad \square$$

With the above results, we are ready to prove [Theorem 3](#) and [Theorem 4](#).

Theorem 3. *At each iteration $k \in [K]$, define the sub-optimality performance gap as*

$$\Delta(p_\theta, y_k^*) = \left| y_k^* - \max_{j \in [N]} f(\mathbf{x}_j) \right|, \quad \text{where } \mathbf{x}_j \sim p_\theta(\cdot | y_k^*, \mathcal{D}), \quad \forall j \in [N]. \quad (18)$$

Assume that there exists some $\theta^ \sim p(\theta | \mathcal{D})$ that produces a predictive distribution $p_{\theta^*}(\cdot | \mathcal{D})$ such that it is able to generate a sample \mathbf{x}^* that perfectly reconstructs y_k^* . Suppose function f is L -Lipschitz and each sample is σ -subGaussian, it can be shown that*

$$\mathbb{E}[\Delta(p_\theta, y_k^*)] \leq c_1 L \sqrt{d} \sigma,$$

where c_1 is some universal constant and the empirical estimator $\widehat{\mathbb{E}}[\Delta(p_\theta, y_k^*)]$ is unbiased.

Proof of Theorem 3. Recall that we consider the case where $N = 1$, and denote \mathbf{x}_k the generated sample in the k -th iteration, i.e. $\mathbf{x}_k \sim p_\theta(\cdot | y_k^*, \mathcal{D})$, where $\theta \sim p(\theta | \mathcal{D})$. In each iteration k , with the existence of $\theta^* \sim p(\theta^* | \mathcal{D})$, we have $y_k^* = f(\mathbf{x}^*)$, where $\mathbf{x}^* \sim p_{\theta^*}(\cdot | y_k^*, \mathcal{D})$. Hence, under [Assumption 2](#),

$$\mathbb{E}[\Delta(p_\theta, y_k^*)] = \mathbb{E}[|f(\mathbf{x}^*) - f(\mathbf{x}_k)|] \leq L \mathbb{E}[\|\mathbf{x}^* - \mathbf{x}_k\|].$$

By [Lemma D.1](#), we have

$$\mathbb{E}[\Delta(p_\theta, y_k^*)] \leq 8L\sqrt{d}\sigma + \mathbb{E}_{\theta, \theta^*}[\|\mathbb{E}[\mathbf{x}^*] - \mathbb{E}[\mathbf{x}_k]\|]. \quad \square$$

Theorem 4. (*Sub-optimality bound*) At each iteration $k \in [K]$, suppose M model parameters $\{\theta_i\}_{i=1}^M$ are generated from the ensemble model for some fixed dataset \mathcal{D} . Suppose function f is L -Lipschitz, it can be shown that the variance of the sub-optimality performance gap of each model is bounded by the epidemic uncertainty:

$$\text{Var}(\Delta(p_{\theta_i}, y_k^*)) \leq c_2 L^2 d \sigma^2 + c_2 L^2 \Delta_{\text{epistemic}}(y_k^*, \mathcal{D}), \quad \forall i \in M, \quad (19)$$

where c_2 is some universal positive constant.

Proof of Theorem 4. At every iteration $k \in [K]$, let the target function value on which the conditional diffusion model conditions be y_k^* . The statement needs to hold for each conditional diffusion model in the ensemble, and thus for simplicity of notation, the subscript i of θ_i is dropped in the remaining proof. With the existence of $\theta^* \sim p(\theta | \mathcal{D})$, we have $y_k^* = f(\mathbf{x}^*)$, where $\mathbf{x}^* \sim p_{\theta^*}(\cdot | y_k^*, \mathcal{D})$. Recall that $f(\mathbf{x}_k)$ is achieved by $\mathbf{x}_k \sim p_{\theta}(\cdot | y_k^*, \mathcal{D})$, where $\theta \sim p(\theta | \mathcal{D})$, and $N = 1$.

Thus, by Eve's law, the overall variance of $\Delta(p_{\theta}, y_k^*)$ can be decomposed as:

$$\begin{aligned} \text{Var}(\Delta(p_{\theta}, y_k^*)) &= \text{Var}(|y_k^* - f(\mathbf{x}_k)|) \\ &= \text{Var}(|f(\mathbf{x}^*) - f(\mathbf{x}_k)|) \\ &= \underbrace{\mathbb{E}_{\theta, \theta^*} [\text{Var}_{\mathbf{x}_k, \mathbf{x}^*} (|f(\mathbf{x}^*) - f(\mathbf{x}_k)| | \theta, \theta^*)]}_{T_1} + \underbrace{\text{Var}_{\theta, \theta^*} (\mathbb{E}_{\mathbf{x}_k, \mathbf{x}^*} [|f(\mathbf{x}^*) - f(\mathbf{x}_k)| | \theta, \theta^*])}_{T_2}. \end{aligned}$$

In particular, the first term T_1 corresponds to the aleatoric component and the second term T_2 corresponds to the epidemic component. We then proceed to bound the above two terms separately.

Step 1: bound T_1 . Under [Assumption 2](#),

$$\text{Var}_{\mathbf{x}_k, \mathbf{x}^*} (|f(\mathbf{x}^*) - f(\mathbf{x}_k)| | \theta, \theta^*) \leq L^2 \text{Var}_{\mathbf{x}_k, \mathbf{x}^*} (\|\mathbf{x}^* - \mathbf{x}_k\| | \theta, \theta^*).$$

Under [Assumption 3](#) and by [Lemma D.2](#),

$$T_1 \leq L^2 \mathbb{E}_{\theta, \theta^*} [\text{Var}_{\mathbf{x}_k, \mathbf{x}^*} (\|\mathbf{x}^* - \mathbf{x}_k\| | \theta, \theta^*)] \leq c_3 L^2 d \sigma^2. \quad (27)$$

Step 2: bound T_2 . Under [Assumption 2](#),

$$T_2 \leq L^2 \text{Var}_{\theta, \theta^*} (\mathbb{E}_{\mathbf{x}_k, \mathbf{x}^*} [\|\mathbf{x}^* - \mathbf{x}_k\| | \theta, \theta^*])$$

By [Lemma D.1](#),

$$\begin{aligned} \text{Var}_{\theta, \theta^*} (\mathbb{E}_{\mathbf{x}_k, \mathbf{x}^*} [|f(\mathbf{x}^*) - f(\mathbf{x}_k)| | \theta, \theta^*]) &\leq \text{Var}_{\theta, \theta^*} \left(\mathbb{E}_{\theta, \theta^*} \left[8\sqrt{d}\sigma + \|\mathbb{E}_{\mathbf{x}^*}[\mathbf{x}^*] - \mathbb{E}_{\mathbf{x}_k}[\mathbf{x}_k]\| \right] \right) \\ &\leq \text{Var}_{\theta, \theta^*} (\|\mathbb{E}_{\mathbf{x}^*}[\mathbf{x}^*] - \mathbb{E}_{\mathbf{x}_k}[\mathbf{x}_k]\|). \end{aligned}$$

Then by property of variance, we have

$$\text{Var}_{\theta, \theta^*} (\|\mathbb{E}_{\mathbf{x}^*}[\mathbf{x}^*] - \mathbb{E}_{\mathbf{x}_k}[\mathbf{x}_k]\|) = \mathbb{E}_{\theta, \theta^*} \left[\|\mathbb{E}_{\mathbf{x}^*}[\mathbf{x}^*] - \mathbb{E}_{\mathbf{x}_k}[\mathbf{x}_k]\|^2 \right] - \left(\mathbb{E}_{\theta, \theta^*} \left[\|\mathbb{E}_{\mathbf{x}^*}[\mathbf{x}^*] - \mathbb{E}_{\mathbf{x}_k}[\mathbf{x}_k]\| \right] \right)^2.$$

From the proof of [Lemma D.2](#), we have

$$\begin{aligned} &\mathbb{E}_{\theta, \theta^*} \left[\|\mathbb{E}_{\mathbf{x}^*}[\mathbf{x}^* | \theta^*] - \mathbb{E}_{\mathbf{x}_k}[\mathbf{x}_k | \theta]\|^2 \right] \\ &= \mathbb{E}_{\theta^*} [\mathbb{E}_{\mathbf{x}^*} [\|\mathbf{x}^*\|^2 | \theta^*]] + \mathbb{E}_{\theta} [\mathbb{E}_{\mathbf{x}_k} [\|\mathbf{x}_k\|^2 | \theta]] - 2\mathbb{E}_{\theta, \theta^*} [\mathbb{E}_{\mathbf{x}_k} [(\mathbf{x}_k | \theta)]^T \mathbb{E}_{\mathbf{x}^*} [\mathbf{x}^* | \theta^*]] \\ &= 2(\mathbb{E}_{\theta} [\mathbb{E}_{\mathbf{x}_k} [\|\mathbf{x}_k\|^2 | \theta]] - \mathbb{E}_{\theta, \theta^*} [\mathbb{E}_{\mathbf{x}_k} [(\mathbf{x}_k | \theta)]^T \mathbb{E}_{\mathbf{x}^*} [\mathbf{x}^* | \theta^*]]) \end{aligned}$$

Combining the above results, we have

$$T_2 \leq L^2 \text{Var}_{\theta, \theta^*} (\|\mathbb{E}_{\mathbf{x}^*}[\mathbf{x}^*] - \mathbb{E}_{\mathbf{x}_k}[\mathbf{x}_k]\|) \leq 2L^2 \text{Var}_{\theta} (\mathbb{E}_{\mathbf{x}_k}[\mathbf{x}_k]). \quad (28)$$

Combining [Equation \(27\)](#) and [Equation \(28\)](#) completes the proof:

$$\text{Var}(\Delta(p_{\theta}, y_k^*)) \leq c_3 L^2 d \sigma^2 + 2L^2 \text{Var}_{\theta} (\mathbb{E}_{\mathbf{x}_k}[\mathbf{x}_k]).$$

□

D.2 Supporting Lemmas

Lemma D.3. Let $\mathbf{x} \in \mathbb{R}^d$ be a σ -subGaussian random vector, then

$$\mathbb{E}[\|\mathbf{x} - \mathbb{E}[\mathbf{x}]\|] \leq 4\sigma\sqrt{d}. \quad (29)$$

Lemma D.4. Let $\mathbf{x} \in \mathbb{R}^d$ be a σ -subGaussian random vector, then its variance satisfies:

$$\text{Var}[\mathbf{x}] \leq C d \sigma^2, \quad (30)$$

where C is some positive constant.

Proof of lemma D.4. By definition of sub-Gaussian vector, for any direction $\mathbf{u} \in \mathbb{R}^d$ with $\|\mathbf{u}\| = 1$,

$$\mathbb{E} \left[\exp(\lambda \mathbf{u}^\top (\mathbf{x} - \mathbb{E}[\mathbf{x}])) \right] \leq \exp \left(\frac{\lambda^2 \sigma^2}{2} \right), \quad \forall \lambda \in \mathbb{R}.$$

This implies that the second moment in any direction \mathbf{u} satisfies:

$$\mathbb{E} \left[\mathbf{u}^\top ((\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\top) \right] \leq \sigma^2.$$

Therefore, the maximum eigenvalue of the covariance matrix is upper-bounded by $C\sigma^2$, where C is some positive constant.

$$\text{Var}[\mathbf{x}] = \text{tr} \left(\mathbb{E} \left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(\mathbf{x} - \mathbb{E}[\mathbf{x}])^\top \right] \right) \leq C d \sigma^2. \quad \square$$

Lemma D.5. In each iteration $k \in [K]$, let \mathcal{D} be the collected dataset, θ and θ^* are parameters independently drawn from posterior $p(\theta|\mathcal{D})$, $\mathbf{x}_k \sim p_\theta(\cdot|y_k^*, \mathcal{D})$ and $\mathbf{x}^* \sim p_{\theta^*}(\cdot|y_k^*, \mathcal{D})$. For any measurable function f , and $\sigma(\mathcal{D})$ -measurable random variable \mathbf{x}_k ,

$$\mathbb{E}[f(\mathbf{x}_k)] = \mathbb{E}[f(\mathbf{x}^*)].$$

Proof of Lemma D.5. Since the black-box function f is measurable, and by the nature of [Algorithm 1](#), in each iteration k , the generated sample \mathbf{x}_k , the target function value y_k^* , the predictive distribution $p_\theta(\cdot|y_k^*, \mathcal{D})$, the posterior distribution $p(\theta|\mathcal{D})$ are $\sigma(\mathcal{D})$ -measurable at iteration k , the only randomness in $f(\mathbf{x})$ comes from the random sampling in the algorithm. Thus, condition on the training data \mathcal{D} and target value y_k^* , by tower rule,

$$\begin{aligned} \mathbb{E}[f(\mathbf{x}_k)] &= \mathbb{E}[\mathbb{E}[f(\mathbf{x}_k)|\theta]] = \int_{\theta} \int_{\mathbf{x}_k} f(\mathbf{x}_k) p_\theta(\mathbf{x}_k|y_k^*, \mathcal{D}) p(\theta|\mathcal{D}) d\mathbf{x}_k d\theta \\ &= \int_{\theta} \int_{\mathbf{x}_k} f(\mathbf{x}_k) p_\theta(\mathbf{x}_k|y_k^*, \mathcal{D}) d\mathbf{x}_k p(\theta|\mathcal{D}) d\theta. \end{aligned}$$

Note that both the true parameter θ^* and the chosen parameter θ are drawn from the same posterior distribution $p(\theta|\mathcal{D})$, we have

$$\int_{\theta} \int_{\mathbf{x}} f(\mathbf{x}) p_\theta(\mathbf{x}|y_k^*, \mathcal{D}) d\mathbf{x} p(\theta|\mathcal{D}) d\theta = \int_{\theta^*} \int_{\mathbf{x}} f(\mathbf{x}) p_{\theta^*}(\mathbf{x}|y_k^*, \mathcal{D}) d\mathbf{x} p(\theta^*|\mathcal{D}) d\theta^*.$$

As a result, we have

$$\mathbb{E}[f(\mathbf{x}_k)] = \int_{\theta^*} \int_{\mathbf{x}^*} f(\mathbf{x}^*) p_{\theta^*}(\mathbf{x}^*|y_k^*, \mathcal{D}) d\mathbf{x}^* p(\theta^*|\mathcal{D}) d\theta^* = \mathbb{E}[\mathbb{E}[f(\mathbf{x}^*)|\theta^*]] = \mathbb{E}[f(\mathbf{x}^*)]. \quad \square$$

Corollary 1. In each iteration $k \in [K]$, let \mathcal{D} be the collected dataset, θ and θ^* are parameters independently drawn from posterior $p(\theta|\mathcal{D})$, $\mathbf{x}_k \sim p_\theta(\cdot|y_k^*, \mathcal{D})$ and $\mathbf{x}^* \sim p_{\theta^*}(\cdot|y_k^*, \mathcal{D})$. For any measurable function f , and $\sigma(\mathcal{D})$ -measurable random variable \mathbf{x}_k ,

$$\mathbb{E}[\|\mathbf{x}_k\|] = \mathbb{E}[\|\mathbf{x}^*\|].$$

Proof of Corollary 1. Since the norm function is deterministic and $\sigma(\mathcal{D})$ -measurable, the proof directly follows that of [Lemma D.5](#). \square

E Optimality of Proposed Acquisition Function

Theorem 1. Let \mathcal{Y} be the constructed candidate set at each iteration $k \in [K]$ in [Algorithm 1](#). By adopting UaE as the acquisition function to guide the sample generation process in conditional diffusion model, Diff-BBO ([Algorithm 1](#)) achieves a near-optimal solution for the online BBO problem defined in [Equation \(1\)](#):

$$\max_{y_k \in \mathbb{R}} \sum_{k=1}^K f(\mathbf{x}_k), \mathbf{x}_k \sim p_\theta(\cdot | y_k, \mathcal{D}), \theta \in \Theta \Rightarrow \max_{y_k \in \mathcal{Y}} \sum_{k=1}^K \alpha(y_k, \mathcal{D}).$$

Proof of Theorem 1. Following [Theorem 4](#), we can express the function evaluation as follows,

$$f(\mathbf{x}_k) = y_k - (y_k - f(\mathbf{x}_k)), \forall k \in [K].$$

The overall objective of the optimization problem defined in [Equation \(1\)](#) can then be further decomposed as

$$\begin{aligned} & \max_{y_k \in \mathbb{R}} \sum_{k=1}^K f(\mathbf{x}_k), \mathbf{x}_k \sim p_\theta(\cdot | y_k), \theta \in \Theta \\ \Leftrightarrow & \max_{y_k \in \mathbb{R}} \sum_{k=1}^K y_k - (y_k - f(\mathbf{x}_k)), \mathbf{x}_k \sim p_\theta(\cdot | y_k), \theta \in \Theta \\ \Rightarrow & \max_{y_k \in \mathbb{R}} \sum_{k=1}^K y_k - \Delta(p_\theta, y_k). \end{aligned}$$

By [Theorem 4](#), which shows $\Delta(p_\theta, y_k^*)$ can be effectively upper bounded the epidemic uncertainty, we therefore have

$$\max_{y_k \in \mathbb{R}} \sum_{k=1}^K f(\mathbf{x}_k), \mathbf{x}_k \sim p_\theta(\cdot | y_k), \theta \in \Theta \Rightarrow \max_{y_k \in \mathcal{Y}} \sum_{k=1}^K y_k - \Delta_{\text{epidemic}}(y_k, \mathcal{D})$$

Essentially, our chosen acquisition function allows Diff-BBO to maximize the lower bound of the original optimization problem. \square

F Experiment Details

F.1 Dataset Details.

DesignBench ([Trabucco et al., 2022](#)) is a benchmark for real-world black-box optimization tasks. For continuous tasks, we use Superconductor, D’Kitty Morphology and Ant Morphology benchmarks. For discrete tasks, we utilize TFBind8 and TFBind10 benchmarks. We exclude Hopper due to the domain is known to be buggy, as explained in Appendix C in ([Krishnamoorthy et al., 2023](#)). We also exclude NAS due to the significant computational resource requirement. Additionally, we exclude the ChEMBL task because the oracle model exhibits non-trivial discrepancies when queried with the same design.

- **Superconductor (materials optimization).** This task involves searching for materials with high critical temperatures. The dataset comprises 17,014 vectors, each with 86 components that represent the number of atoms of each chemical element in the formula. The provided oracle function is a pre-trained random forest regression model.
- **D’Kitty Morphology (robot morphology optimization).** This task focuses on optimizing the parameters of a D’Kitty robot, including the size, orientation, and location of the limbs, to make it suitable for a specific navigation task. The dataset consists of 10,004 entries with a parameter dimension of 56. It utilizes MuJoCO ([Todorov et al., 2012](#)), a robot simulator, as the oracle function.
- **Ant Morphology (robot morphology optimization).** Similar to D’Kitty, this task aims to optimize the parameters of a quadruped robot to maximize its speed. It includes 10,004 data points with a parameter dimension of 60. It also uses MuJoCO as the oracle function.

- **TFBind8 (DNA sequence optimization).** This task seeks to identify the DNA sequence of length eight with the highest binding affinity to the transcription factor SIX6 REF R1. The design space comprises sequences of nucleotides represented as categorical variables. The dataset size is 32,898, with a dimension of 8. The ground truth is used as a direct oracle since the affinity for the entire design space is available.
- **TFBind10 (DNA sequence optimization).** Similar to TFBind8, this task aims to find the DNA sequence of length ten that exhibits the highest binding affinity with transcription factor SIX6 REF R1. The design space consists of all possible nucleotide sequences. The dataset size is 10,000, with a dimension of 10. The ground truth is used as a direct oracle since the affinity for the entire design space is available.

Molecular Discovery. A key problem in drug discovery is the optimization of a compound’s activity against a biological target with therapeutic value. Similar to other papers (Eckmann et al., 2022; Jeon and Kim, 2020; Lee et al., 2023; Noh et al., 2022), we attempt to optimize the score from AutoDock4 (Morris et al., 2009), which is a physics-based estimator of binding affinity. The oracle is a feed-forward model as a surrogate to AutoDock4. The surrogate model is trained until convergence on 10,000 compounds randomly sampled from the latent space (using $\mathcal{N}(0, 1)$) and their computed objective values with AutoDock4. We construct our continuous design space by fixing a random protein embedding and randomly sampling 10,000 molecular embedding of dimension 32.

For each task, we arrange the offline dataset from Krishnamoorthy et al. (2023) in ascending order based on objective values and select data from the 25th to the 50th percentile as the initial training dataset. We prioritize data with lower objective scores to better observe performance differences across each baseline. The overview of all the task statistics is provided in Table 1.

Task	Size	Dimensions	Task Max
TFBind8	32,898	8	1.0
TFBind10	10,000	10	2.128
D’Kitty	10,004	56	340.0
Ant	10,004	60	590.0
Superconductor	17,014	86	185.0
Molecular Discovery	10,000	32	1.0

Table 1: Data Statistics

F.2 Implementation Details.

We train our model on NVIDIA A100 GPU and report the average performance over 3 random runs along with standard deviation for each task. For discrete tasks, we follow the procedure in Krishnamoorthy et al. (2023) where we convert the d -dimensional vector to a $d \times c$ one hot vector regarding c classes. We then approximate logits by interpolating between a uniform distribution and the one hot distribution using a mixing factor of 0.6. We jointly train a conditional and unconditional model with the same model by randomly set the conditioning value to 0 with dropout probability of 0.15.

For each task, we fix the learning rate at 0.001 with batch size of 256. We use 5 ensemble models to estimate the uncertainty for our acquisition function. We set hidden dimensions to 1024 and gamma to 2. We use 10% of the available data at each iteration as validation set during training.

F.3 Ablation study

In this section, we conduct ablation studies to investigate the impact of our designed acquisition function, UaE. We compare Diff-BBO with the fixed condition approach. Instead of using UaE to dynamically determine which y to condition on, the fixed condition approach always generates new samples conditioned on $w \cdot \phi_k$ (Line 9 of Algorithm 1) with a fixed weight w . As shown in Figure 4, Diff-BBO consistently outperforms the fixed condition approach. This demonstrates that our acquisition function is effective in identifying the optimal y for conditioning.

Furthermore, we evaluate the effect of batch size, aka the number of queries per iteration on Diff-BBO on the Superconductor task. As shown in Figure 5, we compare the objective function score over number of function evaluations. We can see the performance of our approach remains similar when the batch size becomes small, suggesting remarkable robustness across different batch sizes. Hence,

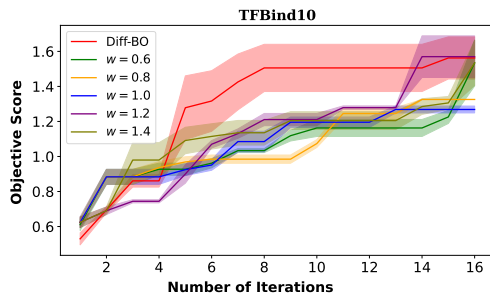


Figure 4: Impact of acquisition function design for black-box optimization on the TFBIND10 task. Comparison of Diff-BBO with five fixed-condition approaches, each with different conditioning weights. Results averaged across three random runs.

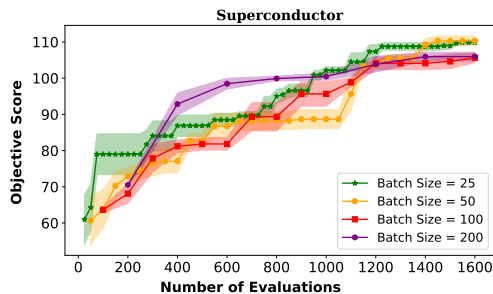


Figure 5: Ablation study to evaluate the effect of batch size on the superconductor task. The mean and standard deviation across three random seeds are plotted. Diff-BBO shows robust performances across different batch size given the same total number of evaluations.

Diff-BBO is a highly-scalable inverse modeling approach that can efficiently leverage parallelism to handle larger computational loads without compromising performance.

G Impact Statement

Optimization techniques can address various real-world problems, including drug and material design. Our method enhances sample-efficient online black-box optimization, potentially accelerating solutions in these areas. However, caution is needed to prevent misuse, such as optimizing drugs to enhance harmful side effects.