

---

# Bayesian Optimization of High-dimensional Outputs with Human Feedback

---

**Qing Feng**  
Meta  
San Francisco, CA  
qingfeng@meta.com

**Zhiyuan Jerry Lin**  
Meta  
USA  
zylin@meta.com

**Yujia Zhang**  
Cubist Systematic Strategies  
USA  
yz685@cornell.edu

**Benjamin Letham**  
Meta  
USA  
bletham@meta.com

**Jelena Markovic-Voronov**  
Meta  
USA  
jmarkovic@meta.com

**Ryan-Rhys Griffiths**  
FutureHouse  
USA  
ryanrhys@meta.com

**Peter I. Frazier**  
Cornell University  
Ithaca, NY  
pf98@cornell.edu

**Eytan Bakshy**  
Meta  
USA  
ebakshy@meta.com

## Abstract

We consider optimizing the inputs to a black-box function that produces high-dimensional outputs such as natural language, images, or robot trajectories. A human decision maker (DM) has a utility function over these outputs. We may learn about the DM’s utility by presenting a small set of outputs and asking which one they prefer. We may learn about the black-box function by evaluating it at adaptively chosen inputs. Given a limited number of such learning opportunities, our goal is to find the input to the black box that maximizes the DM’s utility for the output generated. Previously proposed methods for this and related tasks either do not scale to high dimensional outputs or are statistically inefficient because they ignore information in the outputs. Our proposed approach overcomes these challenges using Bayesian optimization and a novel embedding of high-dimensional outputs into a low-dimensional latent space customized for this task. This embedding is designed to both minimize error when reconstructing high-dimensional outputs and support accurate prediction of human judgments. We demonstrate that this approach significantly improves over baseline methods.

## 1 Introduction

Bayesian optimization (BO) is an efficient method for black-box optimization with applications across machine learning, materials science, engineering, chemistry, and A/B testing. In new grey-box Bayesian optimization approaches, “opening the black box” by modeling intermediate outputs arising during objective function evaluation improves objective function prediction and optimization performance when solving challenging black-box optimization problems [2, 3]. For example, when designing a new material, its properties may depend on the precursors used to synthesize the material and parameters describing the synthesis [25, 57, 19]. In physical simulations, including telecommunications, optical design, and fluid dynamics, directly modeling a simulator’s spatiotemporal outputs can dramatically improve performance [38]. In personalized policy optimization for internet

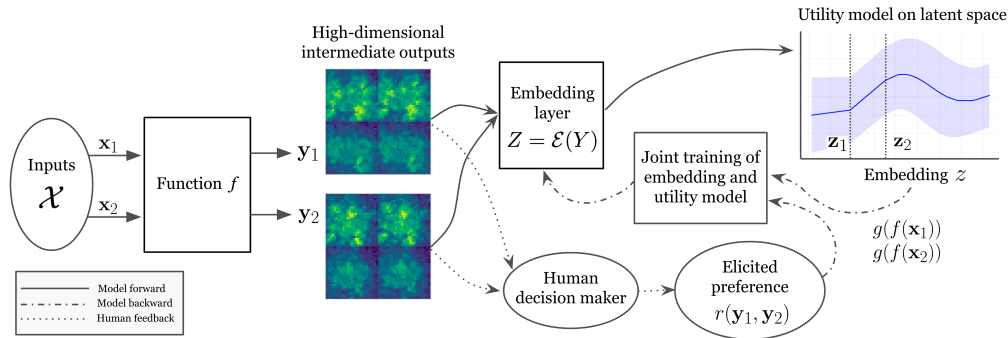


Figure 1: The framework of HDP-GP used for BOHF. Inputs  $\mathbf{x}$  are evaluated by the black-box function  $f$ , producing high-dimensional outputs  $\mathbf{y}$  (e.g., images). The outputs  $\mathbf{y}$  are embedded into a lower-dimensional embedding  $Z$ , producing a representation  $E(\mathbf{y})$ . The human decision maker evaluates pairs of the intermediate outputs and reveals a preference. These elicited preferences are used to jointly learn the embedding and a utility function over the latent space  $Z$  (posterior for utility function shown at right). Utility optimization is then performed in the latent space.

applications, overall experience quality can be decomposed into parameters that primarily affect certain subgroups, and their corresponding outputs [18].

This work aims to open the black box for *Bayesian optimization with human feedback (BOHF)*—a new class of problems where a human decision-maker (DM) judges an object’s quality based on its attributes  $\mathbf{y} \in Y \subseteq \mathbb{R}^k$  according to an unknown utility function  $g(\mathbf{y})$ , and the object’s attributes  $\mathbf{y} = f(\mathbf{x})$  depend on the inputs  $\mathbf{x} \in X \subseteq \mathbb{R}^d$  used to construct the object through an unknown and time-consuming-to-evaluate black-box function  $f : X \rightarrow Y$ . In BOHF, we can interrogate  $f$  by evaluating it at  $\mathbf{x}$  of our choice and we can interrogate  $g$  by asking the DM which of two attribute vectors  $\mathbf{y}, \mathbf{y}^\theta$  they prefer, giving a signal on whether  $g(f(\mathbf{x})) > g(f(\mathbf{x}^\theta))$  or vice versa. The  $\mathbf{y}, \mathbf{y}^\theta$  presented can either be equal to  $f(\mathbf{x})$  at previously-evaluated  $\mathbf{x}$  or, if the application allows it, constructed *de novo*. We refer to these ways of learning about  $f$  and  $g$  as “experiments” and “queries” respectively. Understanding that DM time is valuable and  $f$  is slow to evaluate, we wish to efficiently allocate a limited number of experiments and queries to solve:

$$\mathbf{x} \in \arg \max_{\mathbf{x} \in X} g(f(\mathbf{x})). \quad (1)$$

For example, an engineer configuring cell phone towers may choose a tower configuration  $\mathbf{x}$  and use a time-consuming simulation to compute the resulting pattern of signal power over space  $\mathbf{y} = f(\mathbf{x})$ . The engineer judges the quality of this pattern using their latent utility function  $g(\mathbf{y})$ . To help the engineer find the cell phone tower configuration with the highest utility, a BOHF algorithm can run the simulation to model the mapping from tower configurations  $\mathbf{x}$  to signal power patterns  $\mathbf{y} = f(\mathbf{x})$ , and can query the engineer’s preferences  $g(\mathbf{y})$  by asking which they prefer among pairs of signal power patterns. Later, we will illustrate our method using this example in Fig 1.

BOHF includes two previously-considered problems as special cases: (1) BO with preference exploration (BOPE) [36] assumes that  $f(\mathbf{x})$  is evaluated only when it will be used within a query to the DM while BOHF allows evaluating  $f(\mathbf{x})$  solely to learn about  $f$ . The additional flexibility offered by BOHF is important when DM time is more limited than evaluation of  $f$ . Additionally, as we explain below, past BOPE methods are limited to low-dimensional  $\mathbf{y}$ . (2) In preferential Bayesian optimization (PBO) [26], an algorithm iteratively chooses sets of  $\mathbf{x}$  to evaluate. For each set, attribute vectors  $f(\mathbf{x})$  are evaluated for each  $\mathbf{x}$  in the set and the DM is asked which attribute vector they prefer. While it is computed, the algorithm does not use the information in  $f(\mathbf{x})$ .

Past success in grey-box Bayesian optimization suggests it *should* be possible to use observations of  $f(\mathbf{x})$  (e.g., the signal power pattern) to improve our model of the DM’s utility beyond what is possible using the inputs alone (the tower configurations that generated the signal pattern). This however, poses a substantial challenge: the intermediate output  $f(\mathbf{x})$  is often high-dimensional, especially for complex objects like images, text, or movies. Such complex objects are common in PBO applications. As a result, PBO has ignored the intermediate output, effectively treating  $g \circ f$  in (1) as one black-box function. While BOPE models intermediate outputs, computational limits have

driven past work to focus on applications where  $f(\mathbf{x})$  is low-dimensional: typically 10 dimensions or fewer. While applying a standard dimensionality-reduction method to  $f(\mathbf{x})$  might seem appealing, the resulting low-dimensional representation will ignore preference feedback and thus risks losing key information needed to accurately predict  $g(f(\mathbf{x}))$ . The low-dimensional representation of  $f$  needs to capture the aspects of  $f$  that influence  $g$  and are most germane to the DM’s choice.

Here, we develop the high-dimensional preferential Gaussian process (HDP-GP) model, which uses human feedback to jointly learn a low-dimensional representation of  $f$  and a utility model  $g$ . This joint learning and a tailored architecture for embedding  $f$  ensure that the representation stays aligned with the DM’s utility, in contrast to simply minimizing reconstruction error of  $f$ . This avoids losing information needed to represent the DM’s utility at the expense of structure that is only needed for reconstruction. For example, when recording a movie of a robot performing a task, the background may be needed for accurate reconstruction while fine details of how the robot grasps an object may be orders of magnitude more important for the DM’s utility.

Fig 1 illustrates HDP-GP for BOHF using data and a real embedding from the Cell Tower problem (Sec 3). The DM is given a pair of images and identifies the better of the two, according to the coverage uniformity and interference patterns. In HDP-GP, these image outputs and DM feedback jointly train an embedding into a latent low-dimensional space. We summarize our contributions as:

1. Motivated by real-world decision making processes, we formulate Bayesian optimization with *human* feedback as a grey-box optimization problem with respect to high-dimensional outputs.
2. We propose HDP-GP, a novel model architecture for efficiently modeling latent preferences over high-dimensional outputs whose loss function is tailored toward representing features of the output space relevant to the DM’s utility function.
3. We demonstrate empirically that HDP-GP achieves substantially better optimization performance for PBO and BOPE tasks in high-dimensional settings.

## 2 Framework for BO with Human Feedback over High-dimensional Outputs

The HDP-GP is a framework for BO with human feedback about high-dimensional objects. It leverages and aligns high-dimensional outputs with the DM’s utilities and can be applied to both PBO and BOPE setups. See App A for details on PBO and BOPE and App B for related work.

HDP-GP consists of: (1) an embedding layer that maps high-dimensional outputs  $Y$  to a low-dimensional latent space  $Z$ ; (2) a latent output model that maps from input space  $X$  to  $Z$ ; (3) a utility model that maps from the latent space  $Z$  to the DM’s utility; (4) human feedback that is used to jointly train both of those components, end-to-end, which facilitates output dimensionality reduction through the embedding layer, with supervision from the utility function. Specifically, binary comparisons  $f(\mathbf{y}_j^0, \mathbf{y}_j^{00}, r(\mathbf{y}_j^0, \mathbf{y}_j^{00})) \mathcal{G}_{j=1}^m$  are considered here, where  $(\mathbf{y}_j^0, \mathbf{y}_j^{00})$  is the pair presented to the DM and  $r(\mathbf{y}_j^0, \mathbf{y}_j^{00}) \in \{0, 1\}$  indicates the output that the DM prefers.

**Embedding layer** At the core of the HDP-GP is a transformation, denoted  $E : \mathbb{R}^k \rightarrow \mathbb{R}^p$ , from the output space to a low-dimensional latent output space  $Z = E(Y)$ . This transformation serves the crucial role of distilling effective information for modeling utility. This  $E$  is a versatile transformation encompassing a spectrum of dimensionality reduction methods. The decoder  $D : \mathbb{R}^p \rightarrow \mathbb{R}^k$  maps points back to the original high-dimensional space. The reconstruction error is the  $L_2$  loss,  $\|Y - D(E(Y))\|_2^2$ . We leverage autoencoders (AEs) [50] due to their flexibility and extensibility, making them adept at handling complex structured outputs, such as images, graphs, and sequences.

**Latent output and Utility model** We utilize a multi-output GP  $GP(\mu_0^z, K_0^z)$  on the latent output space  $Z$ . We fit the latent output model on the encoded dataset  $f(\mathbf{x}_i, \mathbf{z}_i) \mathcal{G}_{i=1}^n$ , yielding a posterior mean function  $\mu_n^z$  and posterior covariance function  $K_n^z$ . For computational tractability, we model all  $p$  latent outputs independently so that the fitting cost is linear in  $p$ . The utility model on the embedding  $Z$  is then a GP with a probit likelihood for the DM’s response distribution. We compute an approximate GP posterior with mean  $\mu_n^g$  and covariance function  $K_n^g$  using the Laplace approximation [12]. The primary advantage of HDP-GP lies in its focused learning of the output subspace relevant to the utility function. For instance, when a DM selects between two robot trajectories, only a specific segment of the route may influence the decision; thus, modeling the entire output space would not yield significant utility-related information. Moreover, the scalability of latent

outputs makes HDP-GP more adept at handling high-dimensional outputs compared to the potential scalability issues encountered by the MOGP as the output dimension increases.

**Joint End-to-End Training** We perform joint end-to-end training for the AE, latent output model, and utility model. This ensures that the representation of the outputs remains aligned with the target task: the learning of the decision maker’s utilities with respect to the high-dimensional outputs. The training procedure alternates between refining the latent output model, and updating the AE and utility model. Detailed training workflow is outlined in Algorithm 1.

### 3 Experiments

We evaluate HDP-GP’s performance in both PBO and BOPE settings compared to the state-of-the-art BOHF methods [4, 36] that do not take advantage of the compositional structure of high-dimensional attributes  $y$  (“Non-compositional BO”).

In PBO settings, we evaluate 80 design points using 16 quasi-random initial points and 32 pairwise comparisons. Non-compositional BO in PBO settings is performed as described by Section A.2 using analytic EUBO as the acquisition function [4, 36]. In BOPE settings, we interleave 8 stages of PE and experimentation, starting with 16 quasi-random points, followed by 16 pairwise comparisons and one batch of 8 candidates per experimentation stage. Non-compositional BO in BOPE settings uses qNEIUU for experimentation and analytic EUBO for preference exploration [36, 35]. Moreover, we ablate against HDP-GP without joint training, where unsupervised dimensionality reduction is performed and the AE and utility model are trained separately.

**Image Generation.** Our objective is to generate a rectangle within a 20 × 20 blank square panel and achieve a square of specific size positioned at the center of the panel. The input space is defined to be  $[0, 1]^4$ . We let  $(x_1, x_2)$  and  $(x_3, x_4)$  each specify a vertex, constructing a rectangle with these vertices serving as diagonally opposite corners. Each output is a 400-d binary array representing the resulting image. The utility function is designed to promote the resemblance of the generated image to the desired square at the center of the panel.

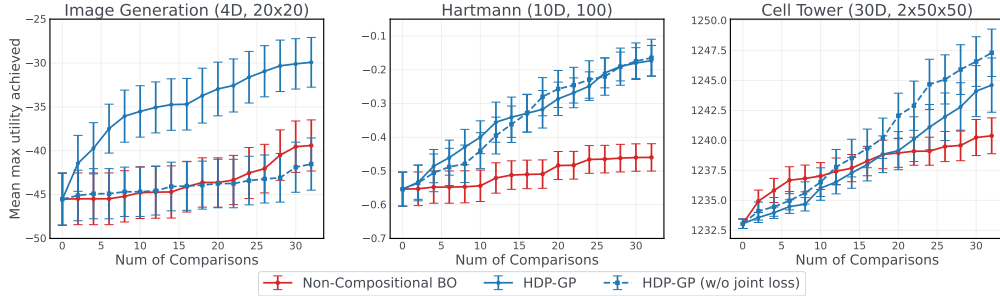


Figure 2: Results of three test problems in the PBO setting.

**Contextual 10-D Hartmann Functions** We consider an input space  $[0, 1]^{10}$  and 100-d output space building on the 5D Hartmann test function in [18], which computes the average of the Hartmann6 function assuming the sixth dimension follows  $U[0,1]$ . We apply the 5D Hartmann function on inputs 1-5 and 6-10 respectively to construct two basis outputs, then project the basis outputs to 100 dimensions by repeating the first and second basis output 50 times each. The utility function is a classification model trained on synthetic datasets. In this setup, we craft a scenario where each output is sparse with respect to the input space, depending solely on 5 parameters.

**Cell-Tower Coverage** We optimize the transmission power and down-tilt angle for each of 15 cell towers (30-d input space) based on [38]. The simulator generates two 50 × 50 intensity maps ( $k = 5000$ ) representing signal power and interference respectively. The utility function is an equally weighted quality metric that prefers high total coverage and low total interference.

**Results** Fig 2 shows the PBO setting. HDP-GP with joint training outperforms all baselines for image generation. Without joint training, the output subspace captures the variation across all pixels, while the utility function focuses on a specific region in the center, causing misalignment. In the contextual 10-D Hartmann functions, the utility function leverages all outputs and is not sparse.

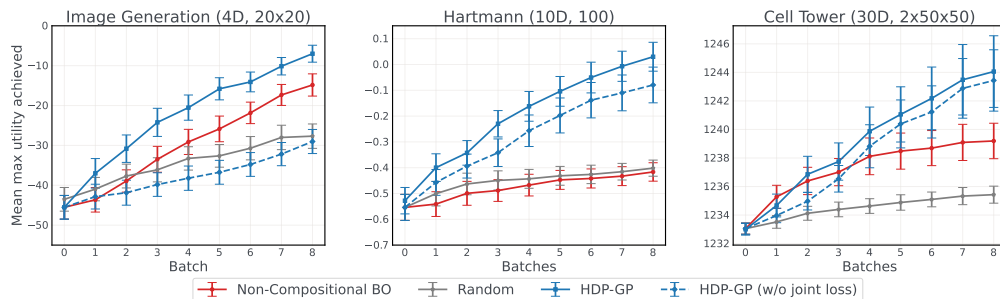


Figure 3: Results of three test problems in the BOPE setting.

Standard Bayesian preferential optimization, which does not utilize the composite structure, performs poorly. In the cell-tower problem, the high-dimensional output space poses computational challenges, highlighting HDP-GP’s superior performance. Fig 3 shows the BOPE setting, where HDP-GP outperforms both non-compositional and random search across all three cases.

The results demonstrate that HDP-GP with joint training consistently outperforms non-compositional BO methods. While HDP-GP without joint training can sometimes perform similarly to HDP-GP with joint training, in tasks such as image generation, joint training of the embedding layer can specifically guide the representation in the latent space, leading to better optimization results. This underscores the efficacy of HDP-GP in distilling pertinent information from high-dimensional outputs for the optimization of utility. Additional test problems and details can be found in the Appendix D.

#### 4 Limitations and conclusion

A great number of real-world problems involve optimization of many competing objectives and complex structured data, yet it is often difficult for a human decision maker to articulate their true utility over such outputs. We introduce the HDP-GP model which captures high-dimensional data underlying decision-maker’s choices. We show that this method improves the performance of Bayesian optimization with Human Feedback, including preferential Bayesian optimization and Bayesian optimization with preference exploration.

Our approach unlocks the ability to effectively perform BOHF over high-dimensional outcomes—a problem with numerous applications in science and engineering. In this work, we show substantial improvements over BOPE and PBO using a simple one-layer auto-encoder architecture, but there is a large design space of improvements to the autoencoder (e.g., multiple layers, the use of regularization and/or denoising autoencoders), and similar more SoTA methods such as diffusion models. Alternative dimensionality reduction techniques could also be considered, such as encoder-only architectures, diffusion models, and pre-trained modules. This could enable HDP-GP to be applied to output spaces of structured objects such as graphs and sequence data in applications such as human-in-the-loop drug discovery [54] and human feedback on generative AI models such as LLMs and image generation [46].

#### References

- [1] Rika Antonova, Akshara Rai, Tianyu Li, and Danica Kragic. Bayesian optimization in variational latent spaces with dynamic compression. In *Conference on Robot Learning*, pages 456–465. PMLR, 2020.
- [2] Raul Astudillo and Peter Frazier. Bayesian optimization of composite functions. In *International Conference on Machine Learning*, pages 354–363. PMLR, 2019.
- [3] Raul Astudillo and Peter Frazier. Multi-attribute Bayesian optimization with interactive preference learning. In *International Conference on Artificial Intelligence and Statistics*, pages 4496–4507. PMLR, 2020.
- [4] Raul Astudillo, Zhiyuan Jerry Lin, Eytan Bakshy, and Peter Frazier. qEUBO: A decision-theoretic acquisition function for preferential Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 1093–1114. PMLR, 2023.

- [5] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. BoTorch: a framework for efficient Monte-Carlo Bayesian optimization. *Advances in Neural Information Processing Systems*, 33:21524–21538, 2020.
- [6] Alessio Benavoli, Dario Azzimonti, and Dario Piga. Preferential Bayesian optimisation with skew Gaussian processes. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1842–1850, 2021.
- [7] Mickaël Binois, David Ginsbourger, and Olivier Roustant. On the choice of the low-dimensional domain for global optimization via random embeddings. *Journal of Global Optimization*, 76: 69–90, 2020.
- [8] Craig Boutilier. A POMDP formulation of preference elicitation problems. In *Eighteenth National Conference on Artificial Intelligence*, page 239–246. American Association for Artificial Intelligence, 2002. ISBN 0262511290.
- [9] Eric Brochu. *Interactive Bayesian optimization: Learning user preferences for graphics and animation*. PhD thesis, University of British Columbia, 2010.
- [10] Eric Brochu, Nando Freitas, and Abhijeet Ghosh. Active preference learning with discrete choice data. *Advances in Neural Information Processing Systems*, 20, 2007.
- [11] Urszula Chajewska, Daphne Koller, and Ronald Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 363–369, 2000.
- [12] Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 137–144, 2005.
- [13] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems*, 33:9851–9864, 2020.
- [14] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in Neural Information Processing Systems*, 34, 2021.
- [15] Stephan Eissman, Daniel Levy, Rui Shu, Stefan Bartzsch, and Stefano Ermon. Bayesian optimization and attribute adjustment. In *Proc. 34th Conference on Uncertainty in Artificial Intelligence*, 2018.
- [16] David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pages 493–503. PMLR, 2021.
- [17] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. *Advances in Neural Information Processing Systems*, 32, 2019.
- [18] Qing Feng, Benjamin Letham, Hongzi Mao, and Eytan Bakshy. High-dimensional contextual policy search with unknown context rewards using bayesian optimization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 22032–22044, 2020.
- [19] Peter I Frazier and Jialei Wang. Bayesian optimization for materials design. *Information science for materials discovery and design*, pages 45–75, 2016.
- [20] Johannes Fürnkranz and Eyke Hüllermeier. Pairwise preference learning and ranking. In *Machine Learning: ECML 2003: 14th European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 14*, pages 145–156. Springer, 2003.
- [21] Johannes Fürnkranz and Eyke Hüllermeier. *Preference Learning*. Springer Science & Business Media, 2010.

- [22] Jacob Gardner, Chuan Guo, Kilian Weinberger, Roman Garnett, and Roger Grosse. Discovering and exploiting additive structure for Bayesian optimization. In *Artificial Intelligence and Statistics*, pages 1311–1319. PMLR, 2017.
- [23] Roman Garnett, Michael A Osborne, and Philipp Hennig. Active learning of linear embeddings for Gaussian processes. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 230–239, 2014.
- [24] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.
- [25] Aldair E Gongora, Bowen Xu, Wyatt Perry, Chika Okoye, Patrick Riley, Kristofer G Reyes, Elise F Morgan, and Keith A Brown. A bayesian experimental autonomous researcher for mechanical design. *Science advances*, 6(15):eaaz1708, 2020.
- [26] Javier González, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. Preferential Bayesian optimization. In *International Conference on Machine Learning*, pages 1282–1291. PMLR, 2017.
- [27] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 11(2):577–586, 2020.
- [28] Ryan-Rhys Griffiths, Leo Klarner, Henry B Moss, Aditya Ravuri, Sang Truong, Bojana Rankovic, Yuanqi Du, Arian Jamasb, Julius Schwartz, Austin Tripp, et al. GAUCHE: A library for Gaussian processes in chemistry. *arXiv preprint arXiv:2212.04450*, 2022.
- [29] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the Fundamentals of Financial Decision Making: Part I*, pages 99–127. World Scientific, 2013.
- [30] Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *International Conference on Machine Learning*, pages 295–304. PMLR, 2015.
- [31] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.
- [32] Ksenia Korovina, Sailun Xu, Kirthevasan Kandasamy, Willie Neiswanger, Barnabas Poczos, Jeff Schneider, and Eric Xing. ChemBO: Bayesian optimization of small organic molecules with synthesizable recommendations. In *International Conference on Artificial Intelligence and Statistics*, pages 3393–3403. PMLR, 2020.
- [33] Benjamin Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems 33*, NeurIPS, 2020.
- [34] Cheng Li, Sunil Gupta, Santu Rana, Vu Nguyen, Svetha Venkatesh, and Alistair Shilton. High dimensional Bayesian optimization using dropout. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2096–2102, 2017.
- [35] Zhiyuan Jerry Lin, Adam Obeng, and Eytan Bakshy. Preference learning for real-world multi-objective decision making. In *ICML 2020 Workshop on Real World Experiment Design and Active Learning*, 2020.
- [36] Zhiyuan Jerry Lin, Raul Astudillo, Peter Frazier, and Eytan Bakshy. Preference exploration for efficient Bayesian optimization with multiple outcomes. In *International Conference on Artificial Intelligence and Statistics*, pages 4235–4258. PMLR, 2022.
- [37] Xiaoyu Lu, Javier Gonzalez, Zhenwen Dai, and Neil D Lawrence. Structured variationally auto-encoded optimization. In *International Conference on Machine Learning*, pages 3267–3275. PMLR, 2018.

- [38] Wesley J Maddox, Maximilian Balandat, Andrew G Wilson, and Eytan Bakshy. Bayesian optimization with high-dimensional outputs. *Advances in Neural Information Processing Systems*, 34:19274–19287, 2021.
- [39] Natalie Maus, Haydn Jones, Juston Moore, Matt J Kusner, John Bradshaw, and Jacob Gardner. Local latent space Bayesian optimization over structured inputs. *Advances in Neural Information Processing Systems*, 35:34505–34518, 2022.
- [40] Riccardo Moriconi, Marc Peter Deisenroth, and KS Sesh Kumar. High-dimensional Bayesian optimization using low-dimensional feature spaces. *Machine Learning*, 109(9):1925–1943, 2020.
- [41] Henry B Moss and Ryan-Rhys Griffiths. Gaussian process molecule property prediction with FlowMO. *arXiv preprint arXiv:2010.01118*, 2020.
- [42] Amin Nayebi, Alexander Munteanu, and Matthias Poloczek. A framework for Bayesian optimization in embedded subspaces. In *Proceedings of the 36th International Conference on Machine Learning*, ICML, pages 4752–4761, 2019.
- [43] Quoc Phong Nguyen, Sebastian Tay, Bryan Kian Hsiang Low, and Patrick Jaillet. Top-k ranking Bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9135–9143, 2021.
- [44] Jens Brehm Bagger Nielsen, Jakob Nielsen, and Jan Larsen. Perception-based personalization of hearing aids using Gaussian processes and active learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(1):162–173, 2014.
- [45] Pascal Notin, José Miguel Hernández-Lobato, and Yarin Gal. Improving black-box optimization in VAE latent space using decoder uncertainty. *Advances in Neural Information Processing Systems*, 34:802–814, 2021.
- [46] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.
- [47] Leonard Papenmeier, Matthias Poloczek, and Luigi Nardi. Increasing the scope as you learn: Adaptive Bayesian optimization in nested subspaces. In *Advances in Neural Information Processing Systems 35, NeurIPS 2022*, volume 35, 2022.
- [48] Maurice Rahme, Ian Abraham, Matthew Elwin, and Todd Murphey. Spotminimini: Pybullet gym environment for gait modulation with bezier curves, 2020. URL [https://github.com/moribots/spot\\_mini\\_mini](https://github.com/moribots/spot_mini_mini).
- [49] Santu Rana, Cheng Li, Sunil Gupta, Vu Nguyen, and Svetha Venkatesh. High dimensional Bayesian optimization with elastic Gaussian process. In *International Conference on Machine Learning*, pages 2883–2891. PMLR, 2017.
- [50] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- [51] Eero Siivola, Akash Kumar Dhaka, Michael Riis Andersen, Javier González, Pablo García Moreno, and Aki Vehtari. Preferential batch Bayesian optimization. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2021.
- [52] Lei Song, Ke Xue, Xiaobin Huang, and Chao Qian. Monte Carlo tree search based variable selection for high dimensional Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2022.
- [53] Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. In *International Conference on Machine Learning*, pages 20459–20478. PMLR, 2022.



- [54] Iris Sundin, Alexey Voronov, Haoping Xiao, Kostas Papadopoulos, Esben Jannik Bjerrum, Markus Heinonen, Atanas Patronov, Samuel Kaski, and Ola Engkvist. Human-in-the-loop assisted de novo molecular design. *Journal of Cheminformatics*, 14(1):1–16, 2022.
- [55] Sonja Surjanovic and Derek Bingham. Virtual library of simulation experiments. 2013. URL <https://www.sfu.ca/~essurjano/langer.html>.
- [56] Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33:11259–11272, 2020.
- [57] Tsuyoshi Ueno, Trevor David Rhone, Zhufeng Hou, Teruyasu Mizoguchi, and Koji Tsuda. Combo: An efficient bayesian optimization library for materials science. *Materials discovery*, 4:18–21, 2016.
- [58] Paolo Viappiani and Craig Boutilier. Optimal Bayesian recommendation sets and myopically optimal choice query sets. *Advances in Neural Information Processing Systems*, 23, 2010.
- [59] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando de Freitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.
- [60] Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Multi-objective Bayesian global optimization using expected hypervolume improvement gradient. *Swarm and Evolutionary Computation*, 44:945–956, 2019.
- [61] M Zhang, H Li, and S Su. High dimensional Bayesian optimization via supervised dimension reduction. In *International Joint Conference on Artificial Intelligence*. International Joint Conference on Artificial Intelligence, 2019.
- [62] Yangwenhui Zhang, Hong Qian, Xiang Shu, and Aimin Zhou. High-dimensional dueling optimization with preference embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11280–11288, 2023.
- [63] Juliusz Krzysztof Ziomek and Haitham Bou Ammar. Are random decompositions all we need in high dimensional Bayesian optimisation? In *International Conference on Machine Learning*, pages 43347–43368. PMLR, 2023.

## A Background: Bayesian Optimization with Preferential Human Feedback

We describe review our data and assumptions about the data-generating process. We then describe the settings considered in prior work on PBO and BOPE.

### A.1 Data

Denote the observed intermediate outputs at the input points  $\tilde{f}\mathbf{x}_i \mathcal{G}_{i=1}^n$  as  $\mathbf{y}_i = f(\mathbf{x}_i)$  and the dataset comprising  $m$  DM-provided binary comparisons as  $P_m = \tilde{f}(\mathbf{y}_j^0, \mathbf{y}_j^{00}), r(\mathbf{y}_j^0, \mathbf{y}_j^{00}) \mathcal{G}_{j=1}^m$ , where  $(\mathbf{y}_j^0, \mathbf{y}_j^{00})$  is the pair presented to the DM and  $r(\mathbf{y}_j^0, \mathbf{y}_j^{00}) \in \{0, 1\}$  indicates the outcome that the DM prefers. The preference feedback is assumed to be consistent with a latent utility function  $g$ . The outcomes over which the DM has provided comparisons have been generated by  $f$ , i.e., that  $\mathbf{y}_j^0 = f(\mathbf{x}_i)$  for some  $i$  for each  $j$ , and similarly for  $\mathbf{y}_j^{00}$ .

### A.2 Preferential Bayesian Optimization (PBO)

In PBO, we seek to solve problem (1) using a workflow in which an algorithm first chooses a pair of points  $(\mathbf{x}_i^0, \mathbf{x}_i^{00})$  and then the function  $f$  is evaluated to obtain  $\mathbf{y}_i^0 = f(\mathbf{x}_i^0)$  and  $\mathbf{y}_i^{00} = f(\mathbf{x}_i^{00})$ . Finally the pair  $(\mathbf{y}_i^0, \mathbf{y}_i^{00})$  is presented to the DM, who selects their preferred option  $r(\mathbf{y}_i^0, \mathbf{y}_i^{00})$ . The standard PBO formulation does not consider the values  $\mathbf{y}_i^0$  and  $\mathbf{y}_i^{00}$  when selecting pairs of points, the PBO model is a mapping directly from  $X$  to utility. In the work here, we consider a compositional extension to PBO in which we instead learn a mapping from the range of  $f$  to utility. When such an intermediate model can be constructed, the acquisition function can be optimized by applying the utility model to the intermediate outcomes like  $g(f(\cdot))$ .

### A.3 Bayesian Optimization with Preference Exploration (BOPE)

In BOPE, we break the PBO workflow into two constituent components: experimentation and preference exploration (PE). In experimentation, we choose points  $\mathbf{x}_i$  at which to evaluate  $f(\mathbf{x}_i)$ , obtaining  $\mathbf{y}_i$ . These need not be presented to the DM. In preference exploration, we choose pairs  $(\mathbf{y}_j^0, \mathbf{y}_j^{00})$  to present to the DM for feedback. These pairs may either be drawn from the pool of previously evaluated  $\mathbf{y}_i$  or, if the application supports it, generated *de novo*. Experiments or preference exploration can be performed in batches or sequentially. Compared to PBO, BOPE offers the flexibility to be able to learn about  $f$  when the DM is unavailable or their time is much more expensive than the cost of performing an experiment.

In our demonstration of the usefulness of HDP-GP, we leverage two tools from the BOPE literature. First is the Expected Utility of the Best Option (EUBO) acquisition function for preference exploration [4]. This acquisition function values a pair  $(\mathbf{y}^0, \mathbf{y}^{00})$  according to the expected utility of the better of the two options, as measured by a random utility function drawn from the posterior over  $g$ . The second is the Noisy Expected Improvement under Utility Uncertainty (qNEIUU) acquisition function [36] for experimentation, which extends the EIUU acquisition function in Astudillo and Frazier [3]. Mathematically,  $\text{qNEIUU}(\mathbf{x}_{1:q}) = \mathbb{E}_{m,b}[\tilde{f}\max g(f(\mathbf{x}_{1:q})) \max g(f(X_b))g^+]$ . Here,  $\mathbf{x}_{1:q}$  denotes a size- $q$  batch of input vectors,  $X_b$  denotes the size- $b$  experimentation data observed so far,  $\mathbb{E}_{m,b}[\cdot] = \mathbb{E}[jP_m, D_b]$  denotes the conditional expectation given the data from queries in  $P_m$  and experimentation data  $D_b$ , and  $f^+g^+$  denotes the positive component of the quantity.

## B Related Work

**High-Dimensional Bayesian Optimization** Black-box optimization in high dimensions may be formulated as the global optimization problem  $\mathbf{x} \in \arg \max_{\mathbf{x} \in X} f(\mathbf{x})$ , where  $f : X \rightarrow \mathbb{R}$  is over a high-dimensional input domain  $X$ . Optimization problems in more than 15-20 continuous dimensions are challenging for classical BO algorithms [59] and as such, many approaches have been taken to extend BO to high dimensions including trust region optimization [17], variable selection [16, 52], and the assumption of low-dimensional additive structure [30, 22]. Our focus in this paper is on methods that produce a low-dimensional embedding. The foundational work on embedding-based BO [59] made use of random embeddings. This work was subsequently extended in [23, 49, 34, 42, 33, 7, 47, 63].

Recently, there has been much interest in autoencoder (AE) embeddings stemming from [24]. This work has been extended to consider constraints on the AE latent space [27, 40, 45] as well as label guidance in learning the latent space [61, 15, 56, 53, 39]. Applications have included chemistry [32], robotics [1], and optimal kernel construction [37]. These studies focus on optimizing high-dimensional inputs with known objectives. Optimization takes place within the latent space of design inputs, with new candidate designs generated by decoding the latent optimizer. In contrast, our approach involves utilizing autoencoders to discover latent representation of the output space to facilitate learning an unknown utility function.

**Preferential Bayesian Optimization (PBO)** Early work in the field of preference learning (PL) [21] arose from the realization that in many problems where it is difficult to specify a ground truth utility function over outcomes, the qualitative judgement of human DMs can be used to produce relative rankings of outcomes [29]. As such, this work focused on learning utility functions from datasets of pairwise preferences elicited from human DMs (e.g., [11, 8, 20, 58]). Ideas from the PL literature were later carried over to GPs [12, 6] and applied to Bayesian optimization [10, 9, 26] where the black-box objective takes the form of a utility function. Subsequent works have focused on new acquisition functions [44, 6, 4], batch variants of PBO [51, 43], and structured outcome spaces using specialized GP kernels [54, 41, 28]. Dueling BO has been explored for high-dimensional inputs using random linear embeddings [62]. Most PBO algorithms operate directly on the input space  $X$ . Work on PBO seeks to reduce the number of pairwise comparisons made between black-box evaluations, but does not motivate whether the function evaluations themselves are costly, or whether making the comparisons is costly. On the other hand, Bayesian optimization with Preference Exploration (BOPE) [35] assumes that evaluating the function itself is costly, but queries to the decision maker are inexpensive. The method utilizes compositional PBO [3] to perform Bayesian optimization with respect to an unknown utility over outcomes generated by the inputs.

**Multi-objective Bayesian Optimization** In multi-objective BO, there are several (often competing) objectives  $f_1, \dots, f_m$  and we wish to recover the Pareto frontier of non-dominated objective configurations. Recent work on multi-objective BO has focused on developing acquisition functions that explicitly target increasing the hypervolume of the known Pareto frontier with respect to a pre-specified reference point [60, 13, 14], however, these methods become impractical for high-dimensional outcomes due to the exponentially increasing surface area of the Pareto frontier.

## C Details of the HDP-GP Framework

### C.1 HDP-GP Algorithm

Algorithm 1 provides the full details of the HDP-GP framework. The training procedure alternates between refining the latent outcome model, and updating the autoencoder (AE) and utility model. Given the updated encoded latent space  $Z$ , we refit the latent outcome model every 200 epochs. Given a fitted latent outcome model, predictions for the latent embeddings  $z$  are obtained, denoted as  $\tilde{z}$ . Subsequently, the loss function of the utility model is computed, involving the marginal log-likelihood with  $\tilde{z}$  as input. Additionally, the reconstruction error  $kY - D(E(Y))k_2^2$  of the autoencoder is computed. The joint loss function, encompassing both the autoencoder and utility model components, is optimized to update the hyper-parameters of the utility model and autoencoder. The joint loss ensures that both the data reconstruction and learning of the utility function are effectively balanced and refined during training.

The choice of hyperparameters  $B$ ,  $M$ , and  $H$  largely depends on the problem dimension and the availability of computational and human resources. As a case in point for human resources, supplement section C1 of Lin et al. [36] contains a real-world pilot study with findings that human DMs take an average of around 10 seconds to compare two outcomes of up to 10 dimensions. For higher-dimensional outcomes, such as those considered in the HDP-GP framework, the comparison would take longer. While monotonically increasing the values of  $B$ ,  $M$ ,  $H$  would be beneficial from the point of view of good model fit and optimization performance, it may place infeasible demands on the time of human decision-makers.

---

**Algorithm 1** Utility-Aligned High-dimensional Greybox Bayesian Optimization (HDP-GP)

---

- 1: Evaluate initial experiment designs  $X_0 \subseteq \mathbb{R}^n$ . Observe outcomes  $Y_0 \subseteq \mathbb{R}^k$  and pairwise comparisons  $P_{m_0} = f(\mathbf{y}_j^0, \mathbf{y}_j^{00}, r(\mathbf{y}_j^0, \mathbf{y}_j^{00}))g_{j=0}^{m_0}$ .
  - 2: Set  $u_0 = \max_{\mathbf{y} \in Y_0} g(\mathbf{y})$ . Set  $m = m_0$ ,  $\tilde{m} = 0$ ;  $b = 0$ ,  $\tilde{b} = 0$ ;  $\eta = 0$ .
  - 3: **while**  $\eta < H$  **do**
  - 4:     Fit autoencoder  $(E, D)$ , outcome model  $\tilde{f}$ , and utility model  $\tilde{g}$  using Algorithm ?? on  $X_b, Y_b, P_m, (E, D)$ .
  - 5:     **while**  $\tilde{m} < M$  **do**
  - 6:         Select a new pair of observed outcomes  $(\mathbf{y}_{m+1}^0, \mathbf{y}_{m+1}^{00})$  by maximizing in-sample EUBO.
  - 7:         Query the user with  $(\mathbf{y}_{m+1}^0, \mathbf{y}_{m+1}^{00})$  and collect preference  $r(\mathbf{y}_{m+1}^0, \mathbf{y}_{m+1}^{00})$ .
  - 8:         Let  $P_{m+1} = f(\mathbf{y}_{m+1}^0, \mathbf{y}_{m+1}^{00}, r(\mathbf{y}_{m+1}^0, \mathbf{y}_{m+1}^{00}))g_{j=1}^{m+1}$ .
  - 9:         Refit utility model  $\tilde{g}$  on encoded preference data  $\tilde{P}_{m+1} = f(E(\mathbf{y}_j^0), E(\mathbf{y}_j^{00}), r(\mathbf{y}_j^0, \mathbf{y}_j^{00}))g_{j=1}^{m+1}$ .
  - 10:         Set  $m = m + 1$ ,  $\tilde{m} = \tilde{m} + 1$ .
  - 11:     **end while**
  - 12:     **while**  $\tilde{b} < B$  **do**
  - 13:         Generate a batch of  $q$  candidates  $X_{\text{cand}}$  by optimizing qNEIUU. Observe outcomes  $Y_{\text{cand}}$ .
  - 14:         Let  $X_{b+1} = X_b \cup X_{\text{cand}}$ ,  $Y_{b+1} = Y_b \cup Y_{\text{cand}}$ .
  - 15:         Refit outcome model  $\tilde{f}$  on  $(X_{b+1}, E(Y_{b+1}))$ .
  - 16:         Compute best so far  $u_{b+1} = \max_{\mathbf{y} \in Y_{b+1}} g(\mathbf{y})$ .
  - 17:         Set  $b = b + 1$ ,  $\tilde{b} = \tilde{b} + 1$ .
  - 18:     **end while**
  - 19:     Set  $\eta = \eta + 1$ . Reset  $\tilde{m} = 0$ ;  $\tilde{b} = 0$ .
  - 20: **end while**
- 

## C.2 Dimensionality Reduction

The autoencoder (AE) component of HDP-GP used to reduce the dimensionality of high-dimensional outcomes comprises a single linear layer for the encoder and a single linear layer plus sigmoid activation function for the decoder. The Adam optimizer [31] is used for training with the default learning rate of 0.01 and 600 epochs for end-to-end training of the AE. Every 200 epochs the latent outcome model is re-fitted where the latent outcomes are the encoded outcomes in the AE case.

## C.3 Preferential Bayesian Optimization

The preferential Bayesian Optimization (PBO) method directly learns the mapping from the input space  $X$  to the utility space by fitting a pairwise GP model [12]. The method also uses in-sample EUBO to select pairs of observed outcomes  $(\mathbf{y}^0 = f(\mathbf{x}^0), \mathbf{y}^{00} = f(\mathbf{x}^{00}))$  for the DM to compare but keeps only the pairs  $(\mathbf{x}^0, \mathbf{x}^{00})$  as the input to the pairwise GP model.

## D Details of the Experiments

### D.1 Experiment Setup

We examine how HDP-GP performs in two different settings: PBO and BOPE. Recall that in PBO, we are eliciting human feedback fully interactively between 2 new design points immediately after evaluating them; while in BOPE, we evaluate a batch of  $q$  points, and perform  $p$  pairwise comparisons selected from  $D_b$ , the data of all observed design points we have collected so far.

Specifically, in PBO experiments, we start with 16 quasi-random initial points, and perform 32 pairwise comparisons between a subsequently selected 64 points, resulting in a total of 80 design points being evaluated. We keep the design point evaluation budget the same in our BOPE experiments by doing 8 stages of interleaving PE and experimentation following 16 quasi-random initial points. Within each PE stage, we conduct 16 pairwise comparisons, while each experimentation stage entails the generation of one batch comprising 8 candidates.

The figures show the mean results across replications (25 replications for the contextual adaptive bitrate simulation (ABR) problem and 50 for all other experiments), and the error bars correspond to 2 standard errors. All experiments are implemented in BoTorch [5]. Code for replicating the methods and benchmark experiments in this work is at <https://anonymous.4open.science/r/LSPE-FEDD> and will be made public upon acceptance.

For the outcome model for all methods, we use a standard ARD Matérn-5/2 kernel when the input dimension is less than 10; in cases where the input dimension exceeds 10, we use the SAAS kernel [16], which identifies important dimensions in the input space through a sparsity-inducing prior on GP lengthscales. Similarly, for the PBO utility model that directly maps from designs to utility, we use a standard ARD Matérn-5/2 kernel for low-dimensional inputs and the SAAS kernel for high-dimensional inputs. We show HDP-GP results with latent space dimension being 16 in all figures unless otherwise indicated. Finally, we perform sensitivity analyses on varying noise levels and the choice of embedding dimension. Appendix E.1 provides further analyses of the PE strategy, acquisition function and the effect of DM error on BOPE performance. The results show that HDP-GP is robust to changes in response noises and latent embedding dimensions in the tested ranges.

## D.2 Rectangle Image Generation

Inspired by [56], we generate rectangles in a blank square image and seek to encourage resemblance to a square of desired size located at the center of the image. We consider images with 20 pixels per side, such that the outcome is represented by a  $k = 400$ -dimensional binary vector. We set the input space to be  $[0, 1]^4$ . For any  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ , we let  $(x_1, x_2)$  and  $(x_3, x_4)$  each specify a vertex (rounded down to the nearest pixel) and construct a rectangle with these coordinates as its diagonal vertices. The outcome function maps from  $[0, 1]^4$  to  $[0, 1]^k$ , producing a  $k$ -dimensional flattened image vector. The outcome model pixel values are allowed to be continuous in the  $[0, 1]$  interval, although the training data consists of binary pixel values 1 (black) and 0 (white). The utility function encourages resemblance to a square of size 64 located at the center of the image. It is the negative difference between an outcome vector and the flattened binary vector representing the desired square.

## D.3 Contextual Adaptive Bitrate Control

Given bitrate vector  $b$ , stall time vector  $t$  and bitrate change vector  $c$ , the full outcome vector for this problem is  $(b, t, c)$  and the utility for this problem is defined as

$$\begin{aligned}
 g(b, t, c) = & \sum_{i=1}^{I_b} b_i + (\text{stall time coeff}) \sum_{i=1}^{I_t} t_i + (\text{bitrate change coeff}) \sum_{i=1}^{I_c} c_i \\
 & + (\text{bitrate smoothness coeff}) \ kr \ bk_2^2 + (\text{stall time smoothness coeff}) \ krtk_2^2 \\
 & + (\text{worst bitrate reward coeff}) \ \min b_i + (\text{worse stall time coeff}) \ \max t_i \\
 & + (\text{worst bitrate change coeff}) \ \max c_i.
 \end{aligned}$$

In our simulations, stall time coeff=-2.8, bitrate change coeff=-0.5, bitrate smoothness coeff=-0.1, stall time smoothness coeff=-0.1, worst bitrate reward coeff=2.0, worse stall time coeff=-5.0, worst bitrate change coeff = -0.1.

## D.4 Contextual 10-D Hartmann

We set the input space to be  $[0, 1]^{10}$  for this problem. Following [18], we apply the contextual 5D Hartmann function, which computes the average of the Hartmann6 function assuming the sixth dimension follows  $U[0,1]$ , to input dimensions 1-5 and 6-10 respectively to generate two basis outputs. To expand the two basis functions into a full 100-dimensional output, a  $2 \times 100$  loading matrix i.e.  $[[\bar{1}_{50}, \bar{0}_{50}]; [\bar{0}_{50}, \bar{1}_{50}]]$  is created. In this case, each outcome would only depend on five input parameters, either dimensions 1-5 or 6-10. The utility function is modeled as a random forest classification, trained on synthetic datasets generated using the scikit-learn package. This design mimics the decision tree-type processes inherent in DM decision-making.

## D.5 Cell-Tower Coverage

The cell tower problem follows the setup described in [38]. In this problem, we tune the transmission power and down-tilt angle of each of 15 cell towers (30-dimensional input space) based on simulated outcomes so as to maximize total coverage while minimizing total interference. For each parameter configuration, the simulator generates 50 × 50 intensity maps of signal power and interference respectively, resulting in a 5000-dimensional outcome space. The utility function is designed to be 0.5 × total power − 0.5 × total interference, placing equal weights on both metrics.

## D.6 50D Synthetic Problem

In this problem, we generate 25 basis functions by sampling from a GP prior with lengthscales ranging from 0.2 to 0.5. A block-diagonal matrix is then constructed to project the low-dimensional subspace to the full outcome space. The utility function is formulated as a weighted sum of these 50 outcomes, with high weights assigned to the first 5 outcomes, each corresponding to a distinct basis function.

## D.7 Langermann

We investigate the synthetic composite function task introduced by [2]. We apply additional transformations to obtain a 75-D outcome space and we use the original 2D Langermann function as the utility function.

The Langermann function [55] (2D) is modified to have high-dimensional outcomes (75D). For this problem the outcome function is

$$f(\mathbf{x}) = (h_1(\mathbf{x}), \cos(\pi h_1(\mathbf{x})), \exp(-h_1(\mathbf{x})/\pi), h_2(\mathbf{x}), \cos(\pi h_2(\mathbf{x})), \exp(-h_2(\mathbf{x})/\pi)),$$

where  $h_1(\mathbf{x}) = (h_{1,1}(\mathbf{x}), \dots, h_{1,J_1}(\mathbf{x}))$ ,  $h_2(\mathbf{x}) = (h_{2,1}(\mathbf{x}), \dots, h_{2,J_2}(\mathbf{x}))$  with

$$h_{1,j}(\mathbf{x}) = \sum_{i=1}^d (x_i - A_{1,ij})^2, \quad j = 1, \dots, J_1,$$

$$h_{2,j}(\mathbf{x}) = \sum_{i=1}^d (x_i - A_{2,ij})^2, \quad j = 1, \dots, J_2,$$

with the  $\cos$  and  $\exp$  operations in the outcome function applied component-wise. The outcome will then have  $3J_1 + 3J_2$  dimensions. The utility function for this problem is defined so as to depend only on the first  $J_1$  coordinates of the outcome vector:

$$g(\mathbf{y}) = \sum_{j=1}^{J_1} c_j \exp(-y_j/\pi) \cos(\pi y_j).$$

In our simulations,  $d = 2$ ,  $J_1 = 5$ ,  $J_2 = 20$ ,  $c = (1, 2, 5, 2, 3)$ ,  $A_1 = \begin{pmatrix} 3 & 5 & 2 & 1 & 7 \\ 5 & 2 & 1 & 4 & 8 \end{pmatrix}$ ,  $A_2 = \begin{pmatrix} 0 & 0.5 & 1 & 1.5 & \dots & 9.5 \\ 10 & 9.5 & 9 & 8.5 & \dots & 0.5 \end{pmatrix}$  and  $X = [0, 10]^2$ .

## D.8 Robot

In this task, we focus on training a four-legged robot to walk by optimizing a 3-dimensional policy. The policy parameters include the swing period, step velocity, and clearance height, and the optimization is conducted using the Spot Mini Mini simulator [48]. The outcome vector is represented by the trajectory of the robot’s centroid in three dimensions over 50 timestamps (150-dimensional outcome space). To expedite evaluations, we employ a surrogate trained on a set of simulation data. The utility function is constructed to reward the robot’s adherence to a predefined trajectory direction. Simultaneously, weights are imposed for deviations and the smoothness of the trajectory in all three dimensions.

Since the robot walking simulation in Spot Mini Mini environment [48] takes a long time to run, we train a decision tree regressor as a surrogate model. We construct a training set of 40,000 data

points, where the 150-dimensional outcome is generated by recording the position at every time step. The utility in this example is chosen such that it rewards the robot movements while also taking into account drift, fluctuations and smoothness. A single robot trajectory is given by  $x$ ,  $y$  and  $z$  vectors, denoting the robot movements along the  $x$ ,  $y$  and  $z$  directions, and so the  $(x, y, z)$  vector here denotes the full outcome vector (acknowledging some abuse of notation since  $\mathbf{x}$  and  $\mathbf{y}$  denote the input and outcome vectors in the general problem formulation). Then, the utility is defined as

$$\begin{aligned}
 u(x, y, z) = & \Delta x + (\text{y drift coeff}) \int \Delta y + (\text{final z award coeff}) \Delta z \\
 & + (\text{y fluctuation coeff}) \text{std}(y) + (\text{z fluctuation coeff}) \text{std}(z) \\
 & + (\text{x smoothness coeff}) \int \Gamma x k_2^2 + (\text{y smoothness coeff}) \int \Gamma y k_2^2 \\
 & + (\text{z smoothness coeff}) \int \Gamma z k_2^2.
 \end{aligned}$$

Here,  $\Delta x = x_s - x_1$ ,  $\Delta y = y_s - y_1$ ,  $\Delta z = z_s - z_1$  are the differences between the ending and the starting  $x$ ,  $y$  and  $z$  values, respectively, where  $s$  is the total number of steps.  $\Gamma x$ ,  $\Gamma y$  and  $\Gamma z$  are the second order gradients of vectors  $x$ ,  $y$  and  $z$  and  $\text{std}(y)$  and  $\text{std}(z)$  are the standard deviations of the  $y$  and  $z$ , respectively. In our simulations, the parameters are taken to be: y drift coeff=2, final z reward coeff=20, y fluctuation coeff=1, z fluctuation coeff=25, x smoothness coeff=500, y smoothness coeff=500, z smoothness coeff=100.

## E Additional Experiment Results

### E.1 Sensitivity Analyses and Ablation Studies

To assess the performance of HDP-GP for different embedding dimensions and the impact of joint training, we designed a test problem with a 5D input space and 50D output space. The outcome space is derived from sample paths of 25 independent GP draws projected into a 50D space. The utility function is constructed as a weighted combination of these 50 outcomes. Further details regarding the data generation process are provided in the Appendix D. Figure 4 illustrates the results of HDP-GP with latent embeddings set at 10 and 20, both with and without joint training. Notably, employing joint training yields significantly better performance, even when the selected latent dimension is smaller than the intrinsic dimensionality of the outcomes. In contrast, without joint training, achieving superior results requires a higher embedding dimension to outperform PBO. This observation underscores the effectiveness of joint training in aligning dimensionality reduction with utility.

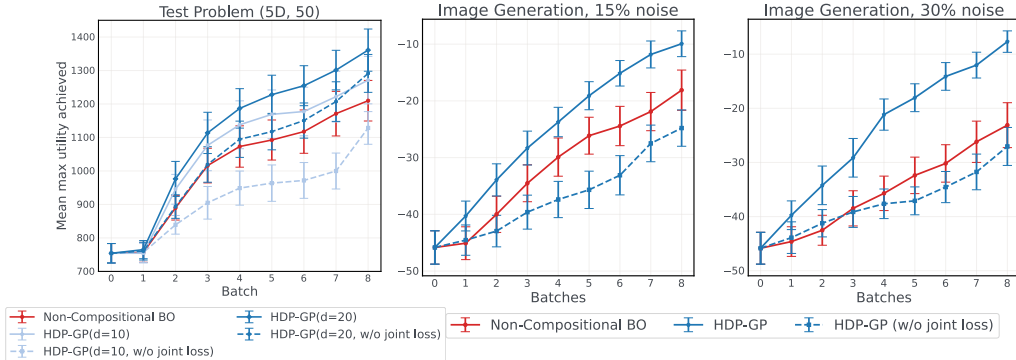


Figure 4: (Left) Optimization performance for a 5D test problem with 50 outputs and a 25-dimensional latent structure. Joint training of the HDP-GP becomes more important when the embedding size is significantly smaller than the true latent dimensionality of the output space. (Middle, Right) Effect of decision maker noise on BOPE under the HDP-GP model. Optimization is robust to DM noise since errors in decision making have a relatively small effect on the output model.

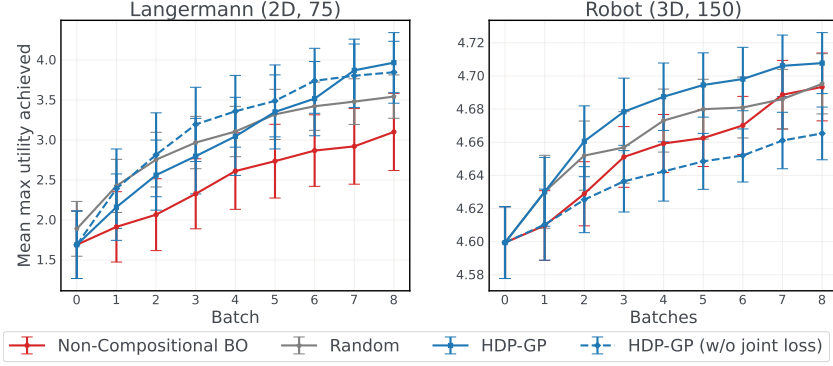


Figure 5: Results of Langermann and Robot problems.

### E.1.1 The Choice of Autoencoder Embedding Dimension

In this section we report a sensitivity analysis on the embedding dimension of the autoencoder component of the HDP-GP framework as well as an ablation study on the incorporation of a kernel with a sparse prior for the outcome model.

In Figure 6 and Figure 7 we show the results of a sensitivity analysis on the embedding dimension of the autoencoder across six problems with the general trend that increasing the embedding dimension improves optimization performance.

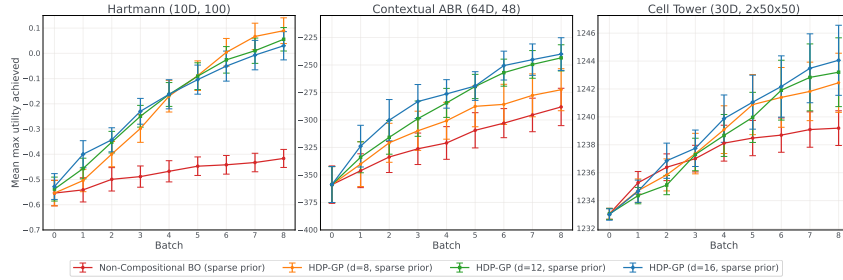


Figure 6: Sensitivity analysis on the choice of embedding dimension for the Hartmann, Contextual ABR and Cell Tower problems.

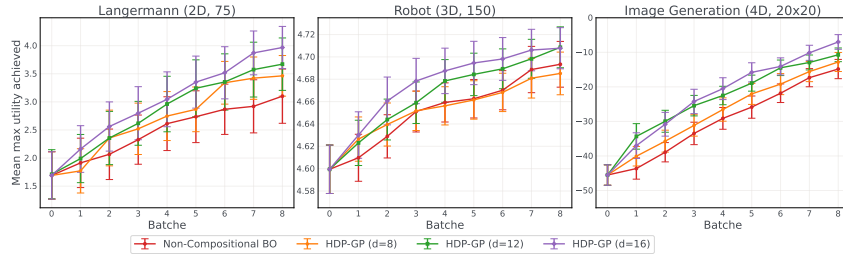


Figure 7: Sensitivity analysis on the choice of embedding dimension for the Langermann, Robot and Image Generation problems.

### E.1.2 The Choice of Outcome Model Kernel

In Figure 8 we show the results of an ablation study to assess the performance of using the SAAS kernel [16] (i.e., imposing a sparsity-inducing prior on the input lengthscales) compared to using a standard ARD Matérn-5/2 kernel for the outcome model. The results indicate that a sparse prior improves performance across the Hartmann, Contextual ABR and Cell Tower tasks.



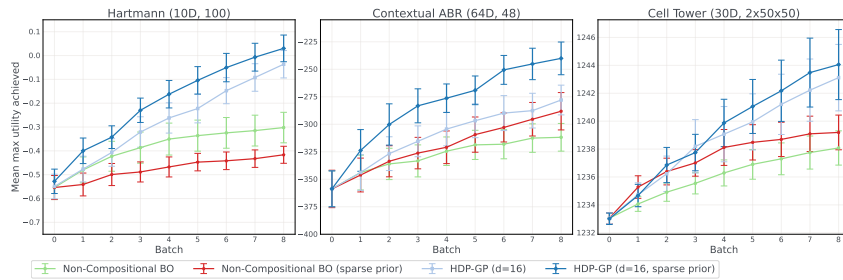


Figure 8: Ablation study on the choice of kernel for the Hartmann, Contextual ABR and Cell Tower problems.

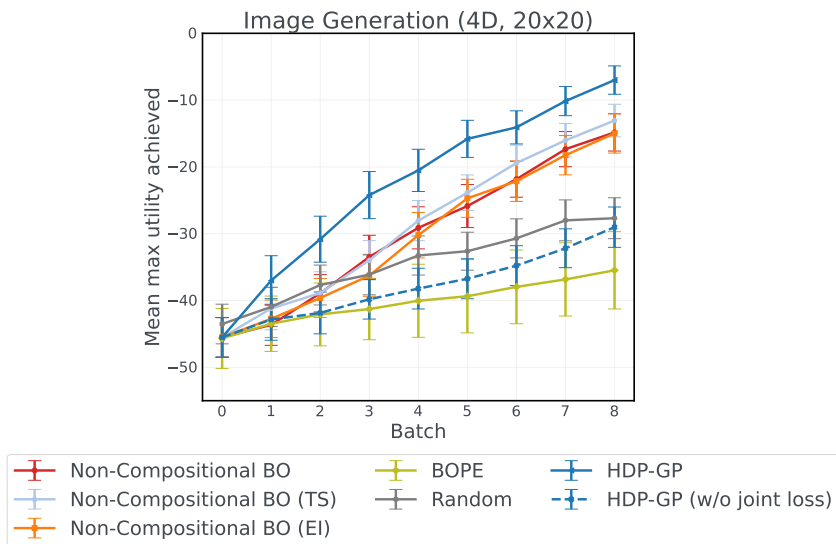


Figure 9: Sensitivity analysis on the choice of preference exploration acquisition function for non-compositional BO.