Enhancing scientific Bayesian optimization via physics-informed operator priors

Sean Hooten, Wolfger Peelaers, Thomas Van Vaerenbergh, Marco Fiorentino
Hewlett Packard Labs
{sean.hooten, wolfger.peelaers, thomas.van-vaerenbergh, marco.fiorentino}@hpe.com

Abstract

Optimization problems in science and engineering often entail conducting expensive experiments and/or large-scale parametric partial differential equation simulations. Hence, methods that optimally trade off between the costs of ground-truth sampling and computation are invaluable. To this end, we propose a novel framework fusing pre-trained physics-informed operator priors with Bayesian optimization (BO). We experimentally demonstrate that our methods are complementary to and improve the sample efficiency and optimization performance of BO in any scientific optimization scenario, including single- and composite-objective problems with in-distribution and out-of-distribution optima. We observe over an order-of-magnitude improvement in mean-squared-error over baselines on large-scale linear and nonlinear test problems. Furthermore, we reveal that increasing effort to scale up physics-informed pre-training results in continuous improvements in sample efficiency, allowing significant freedom in trading off cost sources.

1 Introduction

The goal of many optimization problems in science is to find an optimal parameter setting or input function choice u that maps to a scalar, vector, or tensor field profile v with desirable physical behavior or favorable properties characterized by a known scalar figure-of-merit functional $f(v)$. Such problems are often constrained by known physical laws or equations-of-motion in the form of parametric partial differential equations (PDEs) and by initial and/or boundary conditions, with PDE solutions characterized by an unknown operator $\mathcal{A}(u) = v$. In these types of scientific composite optimization problems, the objective is to find $u^* = \operatorname{argmin}_u f(\mathcal{A}(u))$.

Scientific optimization problems may consist of a combination of real-world experiments and numerical PDE evaluations. Both experiments and PDE simulations can be expensive in terms of time, monetary cost, and compute, thereby necessitating sample efficiency. One option to improve sample efficiency is to use Bayesian optimization (BO), but traditional Gaussian process (GP) models do not scale well to large multi-output scientific problems. Neural network based probabilistic models such as deep ensembles and Bayesian neural networks have been suggested as alternatives to GPs in high-dimensional uncertainty quantification settings [1, 2] and BO [3–5]. However, to this point, such models have not been utilized to their full capacity. Indeed, while the solution operator $\mathcal{A}$ itself may not be known directly, we often have access to the governing initial/boundary value problem. Neural network models that exploit knowledge of PDE-based residual loss functions in training are referred to as physics-informed neural networks (PINNs) when solving single-instance PDEs and physics-informed neural operators (PINOs) when solving parametric PDEs. This paper proposes employing PINOs, both deterministic and probabilistic, as prior models for BO in a novel methodology that we will refer to as physics-informed Bayesian optimization (PIBO). We demonstrate that PIBO facilitates an optimal balance between performance, computational cost, and sample efficiency

1st Workshop on Differentiable Systems and Scientific Machine Learning @ EurIPS 2025

Table 1: Summary of PDE test problems.

Governing Law	Domain	Equation Form	Initial Condition	Boundary Condition
Heat	$x \in [0,1], t \in [0,1]$	$\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial x^2}$	$v(x,0) = u(x)$	$v(0,t) = 0, v(1,t) = 0$
Diffusion-reaction	$x \in [0,1], t \in [0,1]$	$\frac{\partial v}{\partial t} = D\frac{\partial^2 v}{\partial x^2} + kv^2 + u(x)$	$v(x,0) = 0$	$v(0,t) = 0, v(1,t) = 0$
Advection	$x \in [0,1], t \in [0,1]$	$\frac{\partial v}{\partial t} + u(x)\frac{\partial v}{\partial x} = 0$	$v(x,0) = \sin(\pi x)$	$v(0,t) = \sin\left(\frac{\pi}{2}t\right)$
Burgers'	$x \in [0,1], t \in [0,1]$	$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} - \nu\frac{\partial^2 v}{\partial x^2} = 0$	$v(x,0) = u(x)$	$v(0,t) = v(1,t), \frac{\partial v}{\partial x}(0,t) = \frac{\partial v}{\partial x}(1,t)$

in grey-box composite BO on a number of high-dimensional scientific optimization problems, and easily overcomes the limitations and subpar performance of all baselines.

Related work. PIBO combines physics-informed neural operators (PINOs) with Bayesian optimization (BO). Neural operator models have been developed as universal approximators for mappings between Banach spaces of continuously differentiable functions. Notable examples include deep operator network (DeepONet) [6], separable operator network (SepONet) [7], Fourier neural operator (FNO) [8], among many others [9–21]. Detailed reviews may be found in [22–24]. While the vast majority of operator model research in science presumes supervised learning with existing large scientific datasets, a smaller subset of research has extended physics-informed training methods to neural operators [7, 25–28]. While there are many physics-informed and -constrained methods proposed in the literature (see reviews such as [29–31]), this paper will be restricted to the popular varietal utilizing PDE-residual loss functions. Physics-informed training is known to be computationally intensive [32–34], especially if one desires high accuracy guarantees [35]. Fortunately, recent advancements in speeding up the calculation of derivatives of PINNs [34, 36, 37] and PINOs [7] have made physics-informed learning a more attractive option – which we take full advantage of in this paper. Furthermore, one of the great benefits of PIBO, as we will show, is that one can choose optimally how much computation is spent on training a generalized physics-informed prior versus exploiting a posterior model with ground-truth samples. In other words, a physics-informed prior with perfect accuracy is not needed nor warranted for the purposes of BO.

While BO is a relatively old subject, it remains an active domain of research with recent works extending GPs to many data inputs, structured data, or inductive biases [38–48]. Despite these advances, GPs still have trouble scaling tractably to problems with very large input and output dimension, as commonly encountered in scientific optimization. Probabilistic or uncertainty-quantifying neural network methods, such as deep ensembles, have been shown to be an attractive option in high-dimensional settings. For example, [3] and [4] empirically assessed the efficacy of a menagerie of surrogate models, including deep ensembles, in black-box optimization problems. [5] proposed DeepONet randomized prior network ensembles for BO. Finally, [49] used a deterministic Thompson sampling scheme with PINNs to optimize over single-instance PDE solutions. To our knowledge, no prior works have used PINOs in a BO context to solve optimization problems over parametric PDE solutions, which is by far the most valuable and practical setting for scientific optimization. Finally, some other works [50, 51] use the moniker “physics-informed Bayesian optimization” but in far different contexts where neither physics-informed learning nor PDE field models are utilized.

2 Physics-informed Bayesian optimization: methods and results

Description. There are five steps involved in the PIBO pipeline: (1) pre-train or acquire a PINO model, (2) initialize and train a posterior model using ground-truth data, (3) perform BO acquisition to find candidates, (4) perform ground-truth sampling of the most promising candidates, and (5) repeat steps (2)-(4) until a stopping criterion is met. In our case the stopping criterion is 30 iterations (30 ground-truth observations). We describe steps (1)-(4) below.

1. There are many types of PINO models. While we make use of the SepONet [7] in this work, PIBO is agnostic to this choice. Note that we refer to the PINO model as a pre-trained mean operator (PTMO) to emphasize that the model serves as a prior for BO. A general description of physics-informed learning is provided in Appendix B.1 and specific PI loss functions for our test problems are provided in Appendix D. Our experimental hyperparameters for pre-training

are provided in Appendices E and F. Please note that the PTMO is not trained using any ground-truth data; it is only trained using the physics equations-of-motion and prior assumptions on the distributions of the PDE parameters. In our experiments, we consider both cases where the optimal PDE parameters land in- or out-of-distribution (ID or OOD) of these prior assumptions.

2. Physics-informed models may not be accurate and do not generalize well outside of their training distributions. Therefore, we use the PTMO as a prior for a posterior model, which provides uncertainty quantification for predictions given the likelihood of observing ground-truth data. In practice, given ground-truth data samples $\{(u_i, v_i)_{i=1}^n\}$ such that $v_i = \mathcal{A}(u_i)$, and given predictions of the PTMO at those points $\hat{v}_i = \hat{\mathcal{A}}_\theta(u_i)$ such that $\hat{\mathcal{A}}_\theta$ denotes the PTMO predictive operator, we train the posterior model on the residuals of the data denoted by $\mathcal{D} = \{(u_i, v_i - \hat{v}_i)_{i=1}^n\}$. Adding the predictions of the PTMO to the posterior over residuals then gives us uncertainty quantified predictions during BO acquisition. We mainly invoke two types of posterior models: GPs and deep ensembles (DE). A summary of all models is provided in Table 3, DEs are described in Appendix B.2, and specific posterior predictive formulas for all physics-informed posterior models are provided in Appendix C. Note that we also consider baseline models in our experiments that do not include the PTMO prior. Hyperparameters and architectural choices for our experiments are provided in Appendices E and F.
3. Given the (physics-informed) posterior model, we may now perform Bayesian optimization acquisition in the standard way. Recall that our optimization problem is written as $u^* = \operatorname{argmin}_u f(\mathcal{A}(u))$ where f is some known scalar function(al). This type of optimization problem is generally referred to as composite-objective (CO), since we may consider and model the intermediate (field) values $v = \mathcal{A}(u)$. This is in contrast to traditional single-objective (SO) where we only model the final scalar observations y where $y = f(v)$. We consider both CO and SO models and objectives in our experiments. Further description of CO can be found in Appendix B.3. Both SO and CO posterior models are defined in Table 3. In our experiments we use the log expected improvement (LogEI) acquisition function [52], but ablations with other functions may be found in Appendix H.
4. BO acquisition returns promising candidates by balancing exploration and exploitation of the objective function. In our case, we select one candidate per acquisition denoted u_+, and simulate it on the ground-truth PDE operator $v_+ = \mathcal{A}(u_+)$. $\mathcal{A}$ may generally represent some physical experimental outcome or the result of a PDE simulation (as in our case). The PDE test problems are provided in Table 1, descriptions of ground-truth simulators are in Appendix D, and other hyperparameters (like discretization) are provided in Appendix E. After observation, (u_+, v_+) is added to the dataset for posterior refinement.

Results. For reference, the PDE test problems are summarized in Table 1, the figure-of-merit is a mean-squared error with respect to a randomly chosen target optimum. The out-of-distribution (OOD) and in-distribution (ID) target optima are given in Appendix D.5. Final OOD and ID test results are shown in Figure 1(a)-(d) and Figure 1(e)-(h), respectively. Each curve represents the median figure-of-merit for each optimization iteration (ground-truth evaluations) and the error bars represent the 20%-80% percentiles over 10 trials. Descriptions of all models are provided in Table 3.

Overall, the most reliably top-performing model is PTMO-CO-DE, i.e., the composite-objective (CO) model that relies on pre-trained mean operator (PTMO) prior and deep ensemble (DE) posterior models. Indeed, for any given problem, it offers $\gtrsim 10\times$ improvement in figure-of-merit over all baselines after 30 ground-truth evaluations. In some cases, such as the OOD diffusion reaction, ID heat, ID diffusion reaction, and ID advection, the PTMO-CO-DE offered two to three orders of magnitude improvement over baselines after 30 evaluations. Of the baselines, directly exploiting the PTMO model (E-PTMO) performed the best on all problems except for Burgers' equation. Of the BO-based baselines, CO-DE was the most competitive (and better than E-PTMO on Burgers' equation). We conclude that models that combine the exploitation of the PTMO and the exploration, refinement, and generalization offered by the CO-DE enjoy the best performance and sample efficiency.

While the single-objective models such as PTMO-SO-GP and SO-GP were not competitive with their composite deep ensemble (CO-DE) counterparts in terms of sample efficiency, we note that PTMO-SO-GP was always >10× (often >50×) more performant than SO-GP. Generally SO posterior models are much quicker to train and exploit than CO versions, thus allowing one to trade off computational wall-time versus performance. Therefore, our results indicate that physics-informed priors are complementary to and improve the sample efficiency of BO in any scientific optimization scenario, including for single-objective and composite-objective models and for ID and OOD optima.

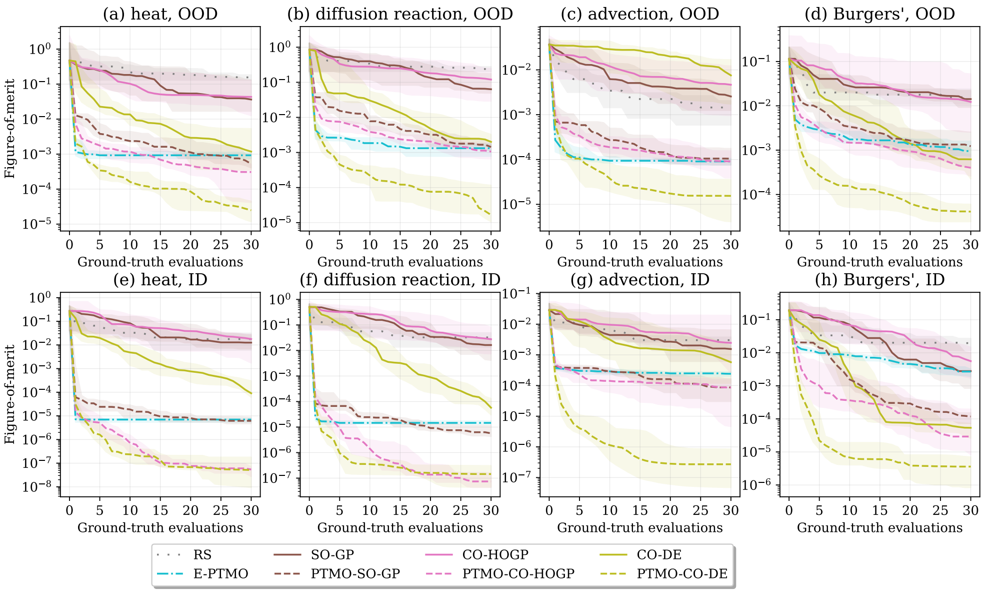


Figure 1: Out-of-distribution (OOD), (a)-(d), and in-distribution (ID), (e)-(h), test results for single-objective (SO) and composite (CO) Bayesian optimization using Gaussian process (GP), high-order GP (HOGP), and deep ensemble (DE) uncertainty models. We supply a pre-trained physics-informed mean operator (PTMO) as an inductive bias to define a prior. We also explore random sampling (RS) and exploiting (E) the PTMO.

3 Discussion

In Figure 1 we demontrated that pre-trained priors can extraordinarily improve scientific BO performance. We expand on this result in Figure 2, where we break down PIBO performance versus computational effort towards prior pre-training. Specifically, we show the figure-of-merit and sample efficiency (number of ground-truth evaluations to obtain 5×10^{-4} mean-squared-error) of the PTMO-CO-DE method on OOD test problems versus the number of PTMO pre-training iterations. We see that both metrics improve drastically as the prior is trained for longer, indicating that the PTMO can improve OOD generalization. However, there are eventually diminishing returns, where the utility of the prior model wanes. Nevertheless, this brings us to the exciting conclusions that (i) a perfect physics-informed prior is not needed to obtain strong results for BO even when target optima are OOD, and (ii) for the same reason, one can optimally trade off how much effort is spent on prior pre-training versus BO exploitation and ground-truth sampling. Finally, it should be emphasized that pre-training the prior is a one-time expense; it can be further recycled for optimization of other target optima. Nevertheless, we find that the pre-training computational expense does not take up the majority of the PIBO runtime – please see our breakdowns of the optimization runtimes in Appendix G.1, Figure 6.

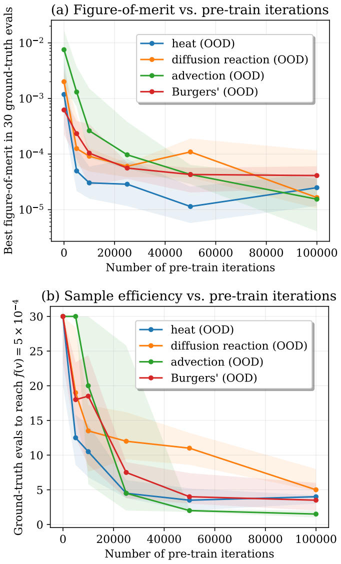


Figure 2: Increasing effort towards pre-training a physics-informed mean operator (PTMO) prior for Bayesian optimization results in continuous improvements in figure-of-merit and sample efficiency.

To summarize, we have demonstrated that physics-informed Bayesian optimization (PIBO) methods, particularly pre-trained physics-informed mean operator priors for composite deep ensembles, offer extremely high performance, reliability, and sample efficiency on PDE-constrained scientific optimization problems, outperforming all baselines by over an order of magnitude even for target optima that lie out-of-distribution.

References

[1] M. Abdar *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021.

[2] J. Gawlikowski *et al.*, "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, pp. 1–77, 2023.

[3] S. Kim, P. Y. Lu, C. Loh, J. Smith, J. Snoek, and M. Soljačić, "Deep learning for bayesian optimization of scientific problems with high-dimensional structure," *Transactions on Machine Learning Research*, 2022. [Online]. Available: `https://openreview.net/forum?id=tPMQ6Je2rB`.

[4] Y. L. Li, T. G. J. Rudner, and A. G. Wilson, "A study of bayesian neural network surrogates for bayesian optimization," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: `https://openreview.net/forum?id=SA19ijj44B`.

[5] M. A. Bhouri, M. Joly, R. Yu, S. Sarkar, and P. Perdikaris, "Scalable Bayesian optimization with high-dimensional outputs using randomized prior networks," 2023. DOI: `10.48550/arXiv.2302.07260`. arXiv: `2302.07260 [cs]`.

[6] L. Lu, P. Jin, and G. E. Karniadakis, "DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators," *Nature Machine Intelligence*, vol. 3, no. 3, pp. 218–229, Mar. 2021, ISSN: 2522-5839. DOI: `10.1038/s42256-021-00302-5`. arXiv: `1910.03193 [cs]`.

[7] X. Yu *et al.*, "Separable Operator Networks," *Transactions on Machine Learning Research*, Sep. 2024, ISSN: 2835-8856.

[8] Z. Li *et al.*, *Fourier Neural Operator for Parametric Partial Differential Equations*, May 2021. DOI: `10.48550/arXiv.2010.08895`. arXiv: `2010.08895 [cs]`.

[9] Z. Li *et al.*, *Neural operator: Graph kernel network for partial differential equations*, 2020. arXiv: `2003.03485 [cs.LG]`. [Online]. Available: `https://arxiv.org/abs/2003.03485`.

[10] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, *Model Reduction and Neural Networks for Parametric PDEs*, Jun. 2021. DOI: `10.48550/arXiv.2005.03180`. arXiv: `2005.03180 [math]`.

[11] S. Lanthaler, *Operator learning with PCA-Net: Upper and lower complexity bounds*, Oct. 2023. DOI: `10.48550/arXiv.2303.16317`. arXiv: `2303.16317 [cs]`.

[12] J. Gu, L. Wen, Y. Chen, and S. Chen, *An explainable operator approximation framework under the guideline of Green's function*, Dec. 2024. DOI: `10.48550/arXiv.2412.16644`. arXiv: `2412.16644 [physics]`.

[13] M. Herde *et al.*, *Poseidon: Efficient Foundation Models for PDEs*, Nov. 2024. DOI: `10.48550/arXiv.2405.19101`. arXiv: `2405.19101 [cs]`.

[14] J. Seidman, G. Kissas, P. Perdikaris, and G. J. Pappas, "NOMAD: Nonlinear manifold decoders for operator learning," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35, Curran Associates, Inc., 2022, pp. 5601–5613.

[15] T. Tripura and S. Chakraborty, "Wavelet Neural Operator for solving parametric partial differential equations in computational mechanics problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 404, p. 115 783, Feb. 2023, ISSN: 0045-7825. DOI: `10.1016/j.cma.2022.115783`.

[16] W. Xiong, X. Huang, Z. Zhang, R. Deng, P. Sun, and Y. Tian, *Koopman neural operator as a mesh-free solver of non-linear partial differential equations*, May 2024. DOI: `10.48550/arXiv.2301.10022`. arXiv: `2301.10022 [cs]`.

[17] B. Alkin, A. Fürst, S. Schmid, L. Gruber, M. Holzleitner, and J. Brandstetter, *Universal Physics Transformers: A Framework For Efficiently Scaling Neural Operators*, Oct. 2024. DOI: `10.48550/arXiv.2402.12365`. arXiv: `2402.12365 [cs]`.

[18] B. Alkin, T. Kronlachner, S. Papa, S. Pirker, T. Lichtenegger, and J. Brandstetter, *NeuralDEM – Real-time Simulation of Industrial Particulate Flows*, Nov. 2024. DOI: `10.48550/arXiv.2411.09678`. arXiv: `2411.09678 [cs]`.

[19] P. Hu *et al.*, *Wavelet Diffusion Neural Operator*, Apr. 2025. DOI: `10.48550/arXiv.2412.04833`. arXiv: `2412.04833 [cs]`.

[20] A. Jiao, Q. Yan, J. Harlim, and L. Lu, *Solving forward and inverse PDE problems on unknown manifolds via physics-informed neural operators*, Jul. 2024. DOI: 10.48550/arXiv.2407.05477. arXiv: 2407.05477 [math].

[21] M. A. Rahman *et al.*, *Pretraining Codomain Attention Neural Operators for Solving Multiphysics PDEs*, Nov. 2024. DOI: 10.48550/arXiv.2403.12553. arXiv: 2403.12553 [cs].

[22] N. Boullé and A. Townsend, "A Mathematical Guide to Operator Learning," in vol. 25, 2024, pp. 83–125. DOI: 10.1016/bs.hna.2024.05.003. arXiv: 2312.14688 [math].

[23] N. B. Kovachki, S. Lanthaler, and A. M. Stuart, *Operator Learning: Algorithms and Analysis*, Feb. 2024. DOI: 10.48550/arXiv.2402.15715. arXiv: 2402.15715 [cs].

[24] N. Kovachki *et al.*, *Neural Operator: Learning Maps Between Function Spaces*, May 2, 2024. arXiv: 2108.08481 [cs.LG]. [Online]. Available: http://arxiv.org/abs/2108.08481.

[25] Z. Li *et al.*, *Physics-Informed Neural Operator for Learning Partial Differential Equations*, Jul. 2023. DOI: 10.48550/arXiv.2111.03794. arXiv: 2111.03794 [cs].

[26] Y. Teng, X. Zhang, Z. Wang, and L. Ju, "Learning Green's Functions of Linear Reaction-Diffusion Equations with Application to Fast Numerical Solver," in *Proceedings of Mathematical and Scientific Machine Learning*, PMLR, Sep. 2022, pp. 1–16.

[27] S. Goswami, A. Bora, Y. Yu, and G. E. Karniadakis, *Physics-Informed Deep Neural Operator Networks*, Jul. 2022. DOI: 10.48550/arXiv.2207.05748. arXiv: 2207.05748 [cs].

[28] S. Wang, H. Wang, and P. Perdikaris, *Learning the solution operator of parametric partial differential equations with physics-informed DeepOnets*, Mar. 2021. DOI: 10.48550/arXiv.2103.10974. arXiv: 2103.10974 [cs].

[29] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next," *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, Jul. 2022, ISSN: 1573-7691. DOI: 10.1007/s10915-022-01939-z.

[30] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, *An Expert's Guide to Training Physics-informed Neural Networks*, Aug. 2023. DOI: 10.48550/arXiv.2308.08468. arXiv: 2308.08468 [cs].

[31] J. D. Toscano *et al.*, *From PINNs to PIKANs: Recent Advances in Physics-Informed Machine Learning*, Oct. 2024. DOI: 10.48550/arXiv.2410.13228. arXiv: 2410.13228 [cs].

[32] D. He *et al.*, *Learning Physics-Informed Neural Networks without Stacked Back-propagation*, Feb. 2023. DOI: 10.48550/arXiv.2202.09340. arXiv: 2202.09340 [cs].

[33] S. Wang, Y. Teng, and P. Perdikaris, *Understanding and mitigating gradient pathologies in physics-informed neural networks*, 2020. arXiv: 2001.04536 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2001.04536.

[34] J. Cho, S. Nam, H. Yang, S.-B. Yun, Y. Hong, and E. Park, *Separable Physics-Informed Neural Networks*, Oct. 2023. DOI: 10.48550/arXiv.2306.15969. arXiv: 2306.15969 [cs].

[35] T. G. Grossmann, U. J. Komorowska, J. Latz, and C.-B. Schönlieb, "Can physics-informed neural networks beat the finite element method?" *IMA Journal of Applied Mathematics*, vol. 89, no. 1, pp. 143–174, Jan. 2024, ISSN: 0272-4960. DOI: 10.1093/imamat/hxae011.

[36] S. K. Vemuri, T. Büchner, J. Niebling, and J. Denzler, "Functional Tensor Decompositions for Physics-Informed Neural Networks," in *Pattern Recognition*, A. Antonacopoulos, S. Chaudhuri, R. Chellappa, C.-L. Liu, S. Bhattacharya, and U. Pal, Eds., Cham: Springer Nature Switzerland, 2025, pp. 32–46, ISBN: 978-3-031-78389-0. DOI: 10.1007/978-3-031-78389-0_3.

[37] Z. Shi, Z. Hu, M. Lin, and K. Kawaguchi, *Stochastic taylor derivative estimator: Efficient amortization for arbitrary differential operators*, 2025. arXiv: 2412.00088 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2412.00088.

[38] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When gaussian process meets big data: A review of scalable gps," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 11, pp. 4405–4423, 2020.

[39] J. Wu, M. Poloczek, A. G. Wilson, and P. I. Frazier, "Bayesian optimization with gradients," *Advances in neural information processing systems*, vol. 30, 2017.

[40] X. Yang, D. Barajas-Solano, G. Tartakovsky, and A. Tartakovsky, “Physics-Informed CoKriging: A Gaussian-Process-Regression-Based Multifidelity Method for Data-Model Convergence,” *Journal of Computational Physics*, vol. 395, pp. 410–431, Oct. 2019, ISSN: 00219991. DOI: `10.1016/j.jcp.2019.06.041`. arXiv: `1811.09757 [stat]`.

[41] G. Pang and G. E. Karniadakis, “Physics-informed learning machines for partial differential equations: Gaussian processes versus neural networks,” *Emerging frontiers in nonlinear science*, pp. 323–343, 2020.

[42] Y. Chen, B. Hosseini, H. Owhadi, and A. M. Stuart, *Solving and Learning Nonlinear PDEs with Gaussian Processes*, Aug. 2021. DOI: `10.48550/arXiv.2103.12959`. arXiv: `2103.12959 [math]`.

[43] M. Pförtner, I. Steinwart, P. Hennig, and J. Wenger, *Physics-Informed Gaussian Process Regression Generalizes Linear PDE Solvers*, Apr. 2024. DOI: `10.48550/arXiv.2212.12474`. arXiv: `2212.12474 [cs]`.

[44] A. K. Uhrenholt and B. S. Jensen, “Efficient bayesian optimization for target vector estimation,” in *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 2661–2670.

[45] R. Astudillo and P. I. Frazier, “Bayesian optimization of composite functions,” in *International Conference on Machine Learning*, PMLR, 2019, pp. 354–363.

[46] M. A. Alvarez, L. Rosasco, and N. D. Lawrence, “Kernels for vector-valued functions: A review,” *Foundations and Trends® in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[47] S. Zhe, W. Xing, and R. M. Kirby, “Scalable high-order gaussian process regression,” in *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 2611–2620.

[48] W. J. Maddox, M. Balandat, A. G. Wilson, and E. Bakshy, “Bayesian optimization with high-dimensional outputs,” *Advances in neural information processing systems*, vol. 34, pp. 19 274–19 287, 2021.

[49] D. Phan-Trong, H. T. Tran, A. Shilton, and S. Gupta, *PINN-BO: A Black-box Optimization Algorithm using Physics-Informed Neural Networks*, Feb. 2024. DOI: `10.48550/arXiv.2402.03243`. arXiv: `2402.03243 [cs]`.

[50] D. Khatamsaz, R. Neuberger, A. M. Roy, S. H. Zadeh, R. Otis, and R. Arróyave, “A physics informed bayesian optimization approach for material design: Application to NiTi shape memory alloys,” *npj Computational Materials*, vol. 9, no. 1, p. 221, Dec. 2023.

[51] W. Kobayashi, T. Otsuka, Y. K. Wakabayashi, and G. Tei, “Physics-informed Bayesian optimization suitable for extrapolation of materials growth,” *npj Computational Materials*, vol. 11, no. 1, p. 36, Feb. 2025.

[52] S. Ament, S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy, *Unexpected Improvements to Expected Improvement for Bayesian Optimization*, Jan. 2025. DOI: `10.48550/arXiv.2310.20708`. arXiv: `2310.20708 [cs]`.

[53] S. Wang, X. Yu, and P. Perdikaris, *When and why PINNs fail to train: A neural tangent kernel perspective*, Jul. 2020. DOI: `10.48550/arXiv.2007.14527`. arXiv: `2007.14527 [cs]`.

[54] S. Wang, A. K. Bhartari, B. Li, and P. Perdikaris, *Gradient Alignment in Physics-informed Neural Networks: A Second-Order Optimization Perspective*, Feb. 2025. DOI: `10.48550/arXiv.2502.00604`. arXiv: `2502.00604 [cs]`.

[55] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” *Advances in neural information processing systems*, vol. 30, 2017.

[56] S. Fort, H. Hu, and B. Lakshminarayanan, “Deep ensembles: A loss landscape perspective,” *arXiv preprint arXiv:1912.02757*, 2019.

[57] F. K. Gustafsson, M. Danelljan, and T. B. Schon, “Evaluating scalable bayesian deep learning methods for robust computer vision,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 318–319.

[58] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, “Snapshot ensembles: Train 1, get m for free,” *arXiv preprint arXiv:1704.00109*, 2017.

[59] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, “Loss surfaces, mode connectivity, and fast ensembling of dnns,” *Advances in neural information processing systems*, vol. 31, 2018.

[60] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," *arXiv preprint arXiv:1803.05407*, 2018.

[61] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for bayesian uncertainty in deep learning," *Advances in neural information processing systems*, vol. 32, 2019.

[62] A. G. Wilson and P. Izmailov, "Bayesian deep learning and a probabilistic perspective of generalization," *Advances in neural information processing systems*, vol. 33, pp. 4697–4708, 2020.

[63] A. G. Wilson, "The case for bayesian deep learning," *arXiv preprint arXiv:2001.10995*, 2020.

[64] P. Kidger, "On Neural Differential Equations," Ph.D. dissertation, University of Oxford, 2021.

[65] A. F. Psaros, X. Meng, Z. Zou, L. Guo, and G. E. Karniadakis, "Uncertainty Quantification in Scientific Machine Learning: Methods, Metrics, and Comparisons," *Journal of Computational Physics*, vol. 477, p. 111 902, Mar. 2023, ISSN: 00219991. DOI: 10.1016/j.jcp.2022.111902. arXiv: 2201.07766 [cs].

[66] Z. Zou, X. Meng, A. F. Psaros, and G. E. Karniadakis, "NeuralUQ: A Comprehensive Library for Uncertainty Quantification in Neural Differential Equations and Operators," *SIAM Review*, vol. 66, no. 1, pp. 161–190, Feb. 2024, ISSN: 0036-1445. DOI: 10.1137/22M1518189.

[67] Z. Zou, X. Meng, and G. E. Karniadakis, "Uncertainty quantification for noisy inputs–outputs in physics-informed neural networks and neural operators," *Computer Methods in Applied Mechanics and Engineering*, vol. 433, p. 117 479, Jan. 2025, ISSN: 0045-7825. DOI: 10.1016/j.cma.2024.117479.

[68] G. Lin, C. Moya, and Z. Zhang, *Accelerated replica exchange stochastic gradient Langevin diffusion enhanced Bayesian DeepONet for solving noisy parametric PDEs*, Nov. 2021. DOI: 10.48550/arXiv.2111.02484. arXiv: 2111.02484 [math].

[69] S. Garg and S. Chakraborty, *Variational Bayes Deep Operator Network: A data-driven Bayesian solver for parametric differential equations*, Jun. 2022. DOI: 10.48550/arXiv.2206.05655. arXiv: 2206.05655 [stat].

[70] A. Pensoneault and X. Zhu, *Uncertainty quantification for deeponets with ensemble kalman inversion*, Mar. 2024. DOI: 10.48550/arXiv.2403.03444. arXiv: 2403.03444 [cs].

[71] Y. Yang, G. Kissas, and P. Perdikaris, "Scalable Uncertainty Quantification for Deep Operator Networks using Randomized Priors," *Computer Methods in Applied Mechanics and Engineering*, vol. 399, p. 115 399, Sep. 2022, ISSN: 00457825. DOI: 10.1016/j.cma.2022.115399. arXiv: 2203.03048 [cs].

[72] S. Kumar, R. Nayek, and S. Chakraborty, *Neural Operator induced Gaussian Process framework for probabilistic solution of parametric partial differential equations*, Apr. 2024. DOI: 10.48550/arXiv.2404.15618. arXiv: 2404.15618 [stat].

A Glossary and summary of methods

A glossary of commonly used notation is provided in Table 2. All acronyms, methods, and baselines used in the paper are summarized in Table 3. Finally, pseudocode for the PIBO fine-tuning algorithm, PTMO-CO-DE-FT, is provided in Algorithm 1.

Table 2: Glossary

Name	Description
$\mathcal{U}, \mathcal{V}$	Banach spaces of differentiable functions, defined on compact domains Ω_u, Ω_v
u	$u \in \mathcal{U}$ is the input function
v	$v \in \mathcal{V}$ is an output function with $v = \mathcal{A}(u)$.
f	$f : \mathcal{V} \to \mathbb{R}$ is the figure-of-merit functional, in our case the standard L^2 norm.
$\mathcal{A}$	$\mathcal{A} : \mathcal{U} \to \mathcal{V}$, unknown ground-truth operator, which satisfies some PDE.
$\hat{\mathcal{A}}_\theta$	$\hat{\mathcal{A}}_\theta : \mathcal{U} \to \mathcal{V}$, neural operator approximation of $\mathcal{A}$ with parameters θ.
$\hat{\mathcal{R}}_\phi$	$\hat{\mathcal{R}}_\phi : \mathcal{U} \to \mathcal{V}$, neural operator approximation of $u \mapsto v - \hat{\mathcal{A}}_\theta(u)$ with parameters ϕ.
$\mathcal{D}^{pi}$	$\mathcal{D}^{pi} = \{(u_i)_{i=1}^{N_u} : u_i \in \mathcal{U}, N_u \geq 1\}$, the physics-informed dataset.
$\mathcal{D}^{sv}$	$\mathcal{D}^{sv} = \{(u_i, v_i)_{i=1}^{N_v} : v_i = \mathcal{A}(u_i), u_i \in \mathcal{U}, v_i \in \mathcal{V}, N_v \geq 1\}$, the supervised dataset.
$\tilde{\mathcal{D}}^{sv}$	$\tilde{\mathcal{D}}^{sv} = \{(u_i, \tilde{v_i}) : \tilde{v_i} = v_i - \hat{\mathcal{A}}_\theta(u_i), \forall (u_i, v_i) \in \mathcal{D}^{sv}\}$, a supervised dataset of residuals.

Table 3: Summary of terms, proposed methods, and baselines

Name	Description
SO	Single-objective optimization
CO	Composite-objective optimization
GP	Single-output Gaussian process
HOGP	High-order Gaussian process [48]
DE	Deep ensemble, eq. (5)
PTMO	Pre-trained physics-informed mean operator, eq. (7)
PTOE	Pre-trained physics-informed operator ensemble prior, eq. (11)
FT	Physics-informed fine-tuning during Bayesian optimization
PTMO-SO-GP	See eq. (8)
PTMO-CO-HOGP	See eq. (9)
PTMO-CO-DE	See eq. (10)
PTOE-CO-DE	See eq. (12)
PTMO-CO-DE-FT	See algorithm 1
RS	Random search
SO-GP	Single-objective Gaussian process (no physics-informed prior)
CO-HOGP	Composite multi-task Gaussian process (no physics-informed prior)
CO-DE	Composite-objective deep ensemble (no physics-informed prior)
E-PTMO	Strictly exploit the physics-informed operator model using gradient descent

B Background and definitions

B.1 Physics-informed operator learning

Let $\mathcal{X}_u, \mathcal{X}_v$ be Banach spaces, $\Omega_u \subseteq \mathcal{X}_u, \Omega_v \subseteq \mathcal{X}_v$ be compact sets, and $\mathcal{U} \subseteq C^k(\Omega_u), \mathcal{V} \subseteq C^k(\Omega_v)$ be Banach spaces of k-fold continuously differentiable functions defined on Ω_u, Ω_v, with $k \geq 1$. The objective of operator learning is to learn an approximation to an unknown mapping $\mathcal{A} : \mathcal{U} \to \mathcal{V}$. While most operator learning implementations assume a supervised training setting, many applications have access to equations-of-motion, often in the form of partial differential equations (PDEs), initial conditions, and boundary conditions underlying $\mathcal{A}$. In particular, for $u \in \mathcal{U}$, $\mathcal{A}$ satisfies

$$\begin{aligned} \mathcal{F}(u, \mathcal{A}(u))(x) &= 0, \quad x \in \Omega_v, \\ \mathcal{B}(u, \mathcal{A}(u))(x) &= 0, \quad x \in \partial\Omega_v, \end{aligned} \tag{1}$$

Algorithm 1: PTMO-CO-DE-FT (fine-tuned) BO

```
Data: $\hat{\mathcal{A}}_\theta, \hat{\mathcal{R}}_{\phi_j}, \mathcal{D}^{sv}, \mathcal{D}^{pi}, C \geq 1, N_{iter} \geq 1, m \geq 1$
Result: $u \approx \text{argmin}_u\, f(\mathcal{A}(u))$
for $i \leftarrow 1$ to $N_{iter}$ do
    $\theta \approx \text{argmin}_\theta\, L_{pi}(\theta; \mathcal{D}^{pi})$ ;    // physics-informed mean operator training
    $\tilde{D}^{sv} \leftarrow \{(u, \tilde{v}) : \tilde{v} = v - \hat{\mathcal{A}}_\theta(u), \forall v \in \mathcal{D}^{sv}\}$ ;    // construct residual dataset
    for $j \leftarrow 1$ to $m$ do
        $\phi_j \approx \text{argmin}_\phi\, L(\phi; \tilde{D}^{sv})$ ;    // optimize each ensemble member independently
    end
    for $c \leftarrow 1$ to $C$ do
        $u_c \approx \text{argmin}_u\, \alpha(u)$;    // obtain candidates using the acquisition function
    end
    $u_+ \leftarrow \text{argmin}(\alpha(u_1), ..., \alpha(u_C))$;    // select the best candidate
    $v_+ = \mathcal{A}(u_+)$;    // ground-truth evaluation of best candidate
    $\mathcal{D}^{sv} \leftarrow \mathcal{D}^{sv} \cup \{(u_+, v_+)\}$;    // append ground-truth data to the supervised dataset
    $\mathcal{D}^{pi} \leftarrow \mathcal{D}^{pi} \cup \{(u_c)_{c=1}^{C}\}$;    // append all of the candidates to the PI dataset
end
```

where $\mathcal{F}$ is a nonlinear differential operator and $\mathcal{B}$ represents boundary and initial conditions (potentially also taking the form of a nonlinear differential operator). In this common situation, $u \in \mathcal{U}$ often represents parameterized boundary/initial conditions, parameter values or maps, or vector fields while $v = \mathcal{A}(u) \in \mathcal{V}$ often represents the resulting field solution to the boundary/initial value problem. In physics-informed operator learning, we assume that eq. (1) is known or partially known, and consequently we may consider it during training.

Let $\mathcal{D}^{sv} = \{(u_i, v_i)_{i=1}^{n}\}$ be a supervised training set of function pairs where $v_i = \mathcal{A}(u_i)$. Let $\mathcal{D}^{pi} = \{(u_i)_{i=1}^{n'}\}$ be a set of training input functions for the unsupervised physics-informed learning. We desire to approximate $\mathcal{A}$ via a learned operator $\hat{\mathcal{A}}_\theta : \mathcal{U} \to \mathcal{V}$ with parameters θ. We consider the supervised and unsupervised (physics-informed) loss functions, respectively:

$$L_{sv}(\theta; \mathcal{D}^{sv}) = \frac{1}{n} \sum_{i=1}^{n} \|v_i - \hat{\mathcal{A}}_\theta(u_i)\|^2_{L^2(\Omega_v)}, \tag{2}$$

$$L_{pi}(\theta; \mathcal{D}^{pi}, \lambda_f, \lambda_b) = \frac{1}{n'} \sum_{i=1}^{n'} \left(\lambda_f \|\mathcal{F}(u_i, \hat{\mathcal{A}}_\theta(u_i))\|^2_{L^2(\Omega_v)} + \lambda_b \|\mathcal{B}(u_i, \hat{\mathcal{A}}_\theta(u_i))\|^2_{L^2(\partial\Omega_v)} \right), \tag{3}$$

where L_{sv} and L_{pi} are the supervised and unsupervised (physics-informed) losses, respectively, $\|\cdot\|_{L^2(\cdot)}$ is the standard 2-norm integrated over the indicated domain, and $\lambda_f, \lambda_b \geq 0$ control the relative weight of the physics-informed loss terms. Note that we take $\lambda_f = \lambda_b = 1$ throughout this work for simplicity and to remove reliance of our results on specification of hyperparameters, which may not be tractable in cost-aware BO situations. However, note that further improvements to PIBO may potentially be obtained by adaptively tuning these hyperparameters [33], leveraging theoretical insight [53], or using second-order approaches [54]. Given some $\hat{\mathcal{A}}_\theta$, the typical goal is to minimize one or both loss functions $\theta^* = \text{argmin}_\theta\, L_{\{sv,pi\}}(\theta; \mathcal{D}^{\{sv,pi\}})$. However, for Bayesian optimization we will be more interested in sampling the posterior of θ, explained below.

B.2 Parallelized posterior sampling for deep ensembles

Bayesian optimization relies upon sampling a posterior predictive approximation that contains enough information for the purpose of exploring the uncertainty of and exploiting knowledge of the optimization domain, in an effort to choose where to evaluate new ground-truth samples. For a model $\hat{\mathcal{A}}_\theta$, $w \in \{sv, pi\}$, and known dataset $\mathcal{D}^w$, the posterior predictive distribution over output functions may be written

$$p(v|u, \mathcal{D}^w) = \int p(v|u, \theta) p(\theta|\mathcal{D}^w) d\theta, \tag{4}$$

where $u \in \mathcal{U}$ is an input function and $p(\theta|\mathcal{D}^w) = \exp\left(-L_w(\theta;\mathcal{D}^w)\right)/Z$ is the model posterior for $L_w(\theta;\mathcal{D}^w)$, defined as either the supervised or physics-informed loss functions in eq. (2) and eq. (3), and Z a normalization constant that is independent of θ. $p(v|u,\theta)$ is the distribution of a single operator's predictions, often chosen to be deterministic or Gaussian.

For composite optimization, we will consider deep ensembles as our primary predictive posterior sampling method (although we will also consider single-output and multi-task Gaussian processes in Section 2). Ensembling approximates eq. (4) by attempting to sample its modes m times independently, where each mode is given equal weight:

$$p(v|u,\mathcal{D}^w) \approx \frac{1}{m}\sum_{i=1}^{m} p(v|u,\theta_i), \quad \theta_i \sim \operatorname{argmin}_\theta L_w(\theta;\mathcal{D}^w)\ . \tag{5}$$

Neural network ensembles (or deep ensembles) have been investigated in several previous works. Deep ensembles have emerged as a powerful method to quantify predictive uncertainty [55]. Their efficacy has been noted theoretically in [56] and experimentally in, for example, [57]. Following [55], several other ensemble techniques have been put forward, such as snapshot ensembles [58], fast geometric ensembles [59] and stochastic weight averaging [60–62]. Comprehensive reviews of uncertainty quantification can be found in [1, 2]. Moreover, while originally introduced as a non-Bayesian alternative to uncertainty quantification, it has been argued in [62, 63] that, in fact, deep ensembles ought to be thought of as a practical means to accurately estimate the Bayesian predictive posterior – a property we exploit to the fullest extent in our work. We have built a scalable library based on JAX to launch independent, vectorized (parallel) posterior samplers for supervised and physics-informed operator learning. This allows for m parallel training sessions on GPU at a training time cost that empirically is only a factor of 2 or 3 longer than training a single model.

B.3 Composite Bayesian optimization over functions

Under the assumption that evaluating a ground-truth functional $u \mapsto f(\mathcal{A}(u))$ is in some sense expensive, composite Bayesian optimization is an algorithm that seeks to find optimal $u^* = \operatorname{argmin}_u f(\mathcal{A}(u))$ by exploiting a surrogate approximation $\hat{\mathcal{A}} \approx \mathcal{A}$ in an effort to improve sample efficiency. In particular, we optimize an acquisition functional defined over our surrogate operator model to determine candidate functions u_c to evaluate on the ground-truth functional. Given posterior predictive model $p(v|u,\mathcal{D})$, the acquisition step of BO optimizes

$$u_c = \operatorname{argmax}_u \alpha(u) := \operatorname{argmax}_u \mathbb{E}[a(f(v))|v \sim p(v|u,\mathcal{D})], \tag{6}$$

where f is the figure-of-merit functional, $a : \mathbb{R} \to \mathbb{R}$ is the utility function which is designed to guide the selection of the next point, and α is the acquisition functional. Popular choices for the acquisition functional are upper confidence bound (UCB) and expected improvement (EI). Recently, [52] showed that the logarithm of the expected improvement (LOGEI) offers improved numerics and performance over EI. We will primarily use LOGEI in this paper, but ablation experiments with other acquisition functions may be found in Appendix H.

C Posterior predictive formulas

We introduce several models to include physics prior information in Bayesian optimization. The first set of models trains a pre-trained physics-informed neural operator (PINO) to define a deterministic prior model. As the operator defines the prior mean, we refer to this method as PTMO, for pre-trained physics-informed mean operator. The next set of models consider a pre-trained physics-informed operator ensemble (PTOE) as a prior. For probabilistic posterior modeling, we will consider single-objective (SO) and composite objective (CO) varietals, where SO will assume a standard GP and CO will assume either a multi-task GP or a deep ensemble (DE). Finally, we will consider models that are fine-tuned with physics information during BO. Below, we will provide posterior predictives for each model that are subsequently used in BO via Equation (6). All methods are summarized in Table 3.

C.1 Pre-trained physics-informed mean operator (PTMO) prior models

Pre-trained physics-informed mean operator (PTMO). Given the unsupervised dataset of inputs $\mathcal{D}^{pi}$, we may train a neural operator to minimize the physics-informed loss in eq. (3). The PTMO model is denoted $\hat{\mathcal{A}}_\theta : \mathcal{U} \to \mathcal{V}$, with posterior predictive

$$p(v|u, \mathcal{D}^{pi}) = \delta(v - \hat{\mathcal{A}}_\theta(u)), \quad \theta \approx \text{argmin}_\theta\, L_{pi}(\theta; \mathcal{D}^{pi}) \tag{7}$$

where δ is the functional Dirac delta.

PTMO single-objective Gaussian process (PTMO-SO-GP). Recall that $f : \mathcal{V} \to \mathbb{R}$ is the (known) figure-of-merit defined over the output functions $v \in \mathcal{V}$. Let $k : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ be a kernel functional defined over inputs u. Our first model uses eq. (7) as a prior and builds a single-objective Gaussian process (SO-GP) to model the residual errors of the figure-of-merit in the BO loop. Given supervised data $\mathcal{D}^{sv}$, the PTMO-SO-GP posterior predictive can be written

$$\begin{aligned} p(f(v) \,|\, u, \mathcal{D}^{pi}, \mathcal{D}^{sv}) = \mathcal{N}(f(\hat{\mathcal{A}}_\theta(u)) + K_{uU}(K_{UU} + \sigma^2 I)^{-1}(f(V) - f(\hat{\mathcal{A}}_\theta(U))), \\ K_{uu} - K_{uU}(K_{UU} + \sigma^2 I)^{-1} K_{Uu}), \end{aligned} \tag{8}$$

where $f(\hat{\mathcal{A}}_\theta(\cdot))$ is the prior mean functional, U is the vector of functions formed by collecting all $u \in \mathcal{D}^{sv}$, $f(\hat{\mathcal{A}}_\theta(U))$ is the pointwise evaluation of the mean functional on all $u \in U$, V is the vector of functions formed from collecting all $v \in \mathcal{D}$, $f(V)$ is the pointwise evaluation of the figure-of-merit on all $v \in V$, σ is the potential noise in the observations, and $K_{(\cdot,\cdot)}$ is the kernel matrix obtained from applying the kernel to the indicated inputs.

PTMO composite-objective high-order GP (PTMO-CO-HOGP). Moving on to the composite optimization (CO) with a PTMO, we will first consider a multi-task GP. We will make use of the high-order GP (HOGP) model that exploits Kronecker structure in the covariance to enable scalable high-dimensional correlated outputs [48]. The posterior predictive may be written

$$\begin{aligned} p(v|u, \mathcal{D}^{pi}, \mathcal{D}^{sv}) = \mathcal{N}(\hat{\mathcal{A}}_\theta(u) + (K_{uU} \otimes K_{VV})(K_{UU} \otimes K_{VV} + \sigma^2 I)^{-1} \text{vec}(V - \hat{\mathcal{A}}_\theta(U)), \\ (K_{uu} \otimes K_{VV}) - (K_{uU} \otimes K_{VV})(K_{UU} \otimes K_{VV} + \sigma^2 I)^{-1}(K_{Uu} \otimes K_{VV})), \end{aligned} \tag{9}$$

where K_{VV}, which models the covariance of the outputs, can be further decomposed for structured outputs, such as if $v \in \mathcal{V}$ is discretized on a regular Cartesian grid. See [48] for further information.

PTMO composite-objective deep ensemble (PTMO-CO-DE). A more attractive CO posterior from the perspective of efficient computation is the deep ensemble. In this case we perform probabilistic modeling using a deep ensemble composed of neural operator models $\hat{\mathcal{R}}_\phi : \mathcal{U} \to \mathcal{V}$ for some posterior-sampled parameters ϕ. Let $\tilde{\mathcal{D}}^{sv}$ be a dataset identical to the regular supervised dataset $\mathcal{D}^{sv}$ except all entries of v are replaced with the residuals $\tilde{v} = v - \hat{\mathcal{A}}_\theta(u)$. Let any given model output a normal distribution with $p(\tilde{v}|u, \phi_i) = \delta(\tilde{v} - \hat{\mathcal{R}}_{\phi_i}(u))$. The predictive posterior for PTMO-CO-DE may then be written,

$$p(v|u, \mathcal{D}^{pi}, \mathcal{D}^{sv}) = \frac{1}{m} \sum_{i=1}^{m} \delta(v - (\hat{\mathcal{A}}_\theta(u) + \hat{\mathcal{R}}_{\phi_i}(u))), \quad \phi_i \sim \text{argmin}_\phi\, L_{sv}(\phi; \tilde{\mathcal{D}}^{sv}), \tag{10}$$

where L_{sv} is the supervised training loss from eq. (2).

C.2 Pre-trained operator ensemble (PTOE) physics-informed prior models

In this case we will only consider CO. PTOE allows us to quantify uncertainty of the physics-informed learning process before beginning BO, potentially relieving bias during acquisition when the physics-informed prior is not well-trained, or when the target optimum is out-of-distribution of the physics-informed training set $\mathcal{D}^{pi}$. The PTOE posterior predictive is obtained by optimizing the physics-informed loss, eq. (3), m times independently:

$$p(v|u, \mathcal{D}^{pi}) = \frac{1}{m} \sum_{i=1}^{m} \delta(v - \hat{\mathcal{A}}_{\theta_i}(u)), \quad \theta_i \sim \text{argmin}_\theta\, L_{pi}(\theta; \mathcal{D}^{pi}), \tag{11}$$

where δ is the functional Dirac delta, and each of $\hat{\mathcal{A}}_\theta$ are trained independently.

PTOE composite-objective deep ensemble (PTOE-CO-DE). Given the PTOE posterior predictive, we fix the parameters θ_i and initialize a new ensemble with an equal number of models $\hat{\mathcal{R}}_{\phi_i}$ for $i = 1, ..., m$. Each model $\hat{\mathcal{R}}_{\phi_i}$ is trained to minimize the error of the residuals between ground-truth data and the corresponding i-th model from the PTOE: $\tilde{v} = v - \mathcal{A}_{\theta_i}(u)$, with corresponding augmented datasets denoted by $\tilde{\mathcal{D}}_i^{sv}$. The final posterior predictive for PTOE-CO-DE is written,

$$p(v|u, \mathcal{D}^{pi}, \mathcal{D}^{sv}) = \frac{1}{m}\sum_{i=1}^{m} \delta(v - (\hat{\mathcal{A}}_{\theta_i}(u) + \hat{\mathcal{R}}_{\phi_i}(u))), \quad \phi_i \sim \text{argmin}_\phi \, L_{sv}(\phi; \tilde{\mathcal{D}}_i^{sv}). \tag{12}$$

C.3 Fine-tuning with physics information during BO (PTMO-CO-DE-FT)

The last strategy we propose is to use physics-informed fine-tuning during the BO loop itself, in an effort to avoid upfront pre-training costs and potentially improve generalization. We will consider a simple variation of the PTMO-CO-DE method from above. The first step is to obtain the PTMO posterior predictive above, i.e., eq. (10). The next step is to identify a set of candidate inputs $\{(u_c)_{c=1}^C\}$ for some $C \geq 1$ using a multi-start acquisition of Equation (6). Usually we keep only the best candidate, say, u_1, find the ground-truth $v_1 = \mathcal{A}(u_1)$, and add it to our supervised dataset $\mathcal{D}^{sv} \leftarrow \mathcal{D}^{sv} \cup \{(u_1, v_1)\}$, while the rest of the candidates $\{u_c\}_{c=2}^C$ are discarded. We propose to instead append all of the candidates to the unsupervised training set $\mathcal{D}^{pi} \leftarrow \mathcal{D}^{pi} \cup \{(u_c)_{c=1}^C\}$. Then, we may fine-tune the physics-informed operator model $\hat{\mathcal{A}}_\theta$, re-train the posterior model given updated $\mathcal{D}^{pi}$ and $\hat{\mathcal{A}}_\theta$, and continue to iterate. The PIBO fine-tuning algorithm for PTMO-CO-DE-FT is summarized in Algorithm 1 in Appendix A.

D PDE test problems and ground-truth simulations

D.1 Heat equation

The parametric time-dependent one-dimensional heat equation initial value problem is given in the first row of Table 1.

The input function is given by $u(x) = \text{window}(x) \cdot \tilde{u}(x)$, where $\tilde{u}(x) \sim \mathcal{GP}(0, k(x, x'))$ is sampled from Gaussian process with RBF kernel (k), length scale 0.2, and amplitude 1.0. $\text{window}(x) = 1 - 16(x - \frac{1}{2})^4$ is a window function that enforces boundary conditions. We discretize $u(x)$ along $d = 128$ uniform points in the unit inteval $x \in [0, 1]$. For physics-informed prior function and ensemble training, we sample $\mathcal{D}^{pi} = \{(u_i)_{i=1}^{5{,}000}\}$ samples. The in-distribution optimum is sampled independently from this same distribution, provided in Figure 4(a). The out-of-distribution optimum is sampled independently from a similar distribution, except the kernel function is exchanged with an RBF kernel with length scale 0.1. It is provided in Figure 3(a). The physics-informed prior training does not see any samples from the OOD distribution.

Ground-truth samples $v(x, t)$ are computed using an explicit solver Tsit5 available in diffrax [64] with initial time step 0.0001, maximum time step of 0.001, time interval $t \in [0, 1]$, absolute tolerance 10^{-10}, and relative tolerance 10^{-10} for ODE time-stepping. We downsample the output in time to obtain a final shape $[128, 128]$ for training the posterior model.

The physics-informed loss function can be written as

$$\begin{aligned} L(\theta; \mathcal{D}^{pi}) = &\frac{1}{N_u N_c}\sum_{i=1}^{N_u}\sum_{j=1}^{N_c}\left(\frac{\partial \hat{\mathcal{A}}_\theta(u_i)(x_j, t_j)}{\partial t} - \frac{\partial^2 \hat{\mathcal{A}}_\theta(u_i)(x_j, t_j)}{\partial x^2}\right)^2 \\ &+ \frac{1}{N_u N_b}\sum_{i=1}^{N_u}\sum_{j=1}^{N_b}(\hat{\mathcal{A}}_\theta(u_i)(0, t_j))^2 + (\hat{\mathcal{A}}_\theta(u_i)(1, t_j))^2 \\ &+ \frac{1}{N_u N_I}\sum_{i=1}^{N_u}\sum_{j=1}^{N_I}(\hat{\mathcal{A}}_\theta(u_i)(x_j, 0) - u_i(x_j))^2 \end{aligned} \tag{13}$$

where $\hat{\mathcal{A}}_\theta$ is the neural operator model, $N_c = 128 \times 128$, $N_b = 128$, and $N_I = 128$ correspond to the interior, boundary, and initial condition points, sampled uniformly each iteration. $N_u = 128$ is the batch size for input functions.

D.2 Diffusion-reaction equation

The parametric nonlinear time-dependent one-dimensional diffusion-reaction equation initial value problem is given in the second row of Table 1. The parameters D and k are $\mathcal{D} = 0.01$ and $\| = 0.01$.

The input function u is sampled from a Gaussian process with RBF kernel (k), length scale 0.2, and amplitude 1.0. We discretize $u(x)$ along $d = 128$ uniform points in the unit inteval $x \in [0, 1]$. For physics-informed prior function and ensemble training, we sample $\mathcal{D}^{pi} = \{(u_i)_{i=1}^{5{,}000}\}$ samples. The in-distribution optimum is sampled independently from this same distribution, provided in Figure 4(b). The out-of-distribution optimum is sampled independently from a similar distribution, except the kernel function is exchanged with an RBF kernel with length scale 0.1. It is provided in Figure 3(b). The physics-informed prior training does not see any samples from the OOD distribution.

Ground-truth samples $v(x, t)$ are computed using an explicit solver Tsit5 available in diffrax [64] with initial time step 0.0001, maximum time step of 0.001, time interval $t \in [0, 1]$, absolute tolerance 10^{-10}, and relative tolerance 10^{-10} for ODE time-stepping. We downsample the output in time to obtain a final shape $[128, 128]$ for training the posterior model.

The physics-informed loss function can be written as

$$\begin{aligned} L(\theta; \mathcal{D}^{pi}) = & \frac{1}{N_u N_c} \sum_{i=1}^{N_u} \sum_{j=1}^{N_c} \left(\frac{\partial \hat{\mathcal{A}}_\theta(u_i)(x_j, t_j)}{\partial t} - D \frac{\partial^2 \hat{\mathcal{A}}_\theta(u_i)(x_j, t_j)}{\partial x^2} - k(\hat{\mathcal{A}}_\theta(u_i)(x_j, t_j))^2 - u_i(x_j) \right)^2 \\ & + \frac{1}{N_u N_b} \sum_{i=1}^{N_u} \sum_{j=1}^{N_b} (\hat{\mathcal{A}}_\theta(u_i)(0, t_j))^2 + (\hat{\mathcal{A}}_\theta(u_i)(1, t_j))^2 \\ & + \frac{1}{N_u N_I} \sum_{i=1}^{N_u} \sum_{j=1}^{N_I} (\hat{\mathcal{A}}_\theta(u_i)(x_j, 0))^2 \end{aligned} \tag{14}$$

where $D = k = 0.01$, $\hat{\mathcal{A}}_\theta$ is the neural operator model, $N_c = 128 \times 128$, $N_b = 128$, and $N_I = 128$ correspond to the interior, boundary, and initial condition points, sampled independently and uniformly each iteration. $N_u = 128$ is the batch size for input functions.

D.3 Advection equation

The parametric time-dependent one-dimensional advection equation initial value problem is given in the third row of Table 1.

The input function is given by $u(x) = \tilde{u}(x) - \min_x \tilde{u}(x) + 1$, where $\tilde{u}(x) \sim \mathcal{GP}(0, k(x, x'))$ is sampled from Gaussian process with RBF kernel (k), length scale 0.2, and amplitude 1.0. We discretize $u(x)$ along $d = 128$ uniform points in the unit interval $x \in [0, 1]$. For physics-informed prior function and ensemble training, we sample $\mathcal{D}^{pi} = \{(u_i)_{i=1}^{5{,}000}\}$ samples. The in-distribution optimum is sampled independently from this same distribution, provided in Figure 4(c). The out-of-distribution optimum is sampled independently from a similar distribution, except the kernel function is exchanged with an RBF kernel with length scale 0.1. It is provided in Figure 3(c). The physics-informed prior training does not see any samples from the OOD distribution.

Ground-truth samples $v(x, t)$ are computed using an explicit solver Tsit5 available in diffrax [64] with initial time step 0.0001, maximum time step of 0.001, time interval $t \in [0, 1]$, absolute tolerance 10^{-10}, and relative tolerance 10^{-10} for ODE time-stepping. We downsample the output in time to obtain a final shape $[128, 128]$ for training the posterior model.

The physics-informed loss function can be written as

$$
\begin{aligned}
L(\theta; \mathcal{D}^{pi}) = & \frac{1}{N_u N_c} \sum_{i=1}^{N_u} \sum_{j=1}^{N_c} \left(\frac{\partial \hat{\mathcal{A}}_\theta(u_i)(x_j, t_j)}{\partial t} - u_i(x_j) \frac{\partial \hat{\mathcal{A}}_\theta(u_i)(x_j, t_j)}{\partial x} \right)^2 \\
& + \frac{1}{N_u N_b} \sum_{i=1}^{N_u} \sum_{j=1}^{N_b} \left(\hat{\mathcal{A}}_\theta(u_i)(0, t_j) - \sin\left(\frac{\pi}{2} t_j\right) \right)^2 \\
& + \frac{1}{N_u N_I} \sum_{i=1}^{N_u} \sum_{j=1}^{N_I} (\hat{\mathcal{A}}_\theta(u_i)(x_j, 0) - \sin(\pi x_j))^2
\end{aligned} \tag{15}
$$

where $\hat{\mathcal{A}}_\theta$ is the neural operator model, $N_c = 128 \times 128$, $N_b = 128$, and $N_I = 128$ correspond to the interior, boundary, and initial condition points, sampled uniformly each iteration. $N_u = 128$ is the batch size for input functions.

D.4 Burgers' Equation

The nonlinear parametric time-dependent one-dimensional Burgers' equation initial value problem is given in the fourth row of Table 1. The parameter $\nu = 0.01$. Note that Burgers' equation assumes periodic boundary conditions.

The input function is given by,

$$
u(x) = \frac{1}{4} \left(\varepsilon_1 + \sum_{n=1}^{50} \left(\frac{\varepsilon_{2n}}{n^3} \sin(2\pi n x) + \frac{\varepsilon_{2n+1}}{n^3} \cos(2\pi n x) \right) \right), \quad \varepsilon_i \sim \mathcal{N}(0,1), \quad i = 1, ..., 101 \tag{16}
$$

which was chosen to maintain periodicity. We discretize $u(x)$ along $d = 101$ uniform points in the unit interval $x \in [0,1]$. For physics-informed prior function and ensemble training, we sample $\mathcal{D}^{pi} = \{(u_i)_{i=1}^{5{,}000}\}$ samples. The in-distribution optimum is sampled independently from this same distribution, provided in Figure 4(d). The out-of-distribution optimum is sampled independently from a similar distribution given by,

$$
u(x) = \frac{1}{4} \left(\varepsilon_1 + \sum_{n=1}^{50} \left(\frac{\varepsilon_{2n}}{n^2} \sin(2\pi n x) + \frac{\varepsilon_{2n+1}}{n^2} \cos(2\pi n x) \right) \right), \quad \varepsilon_i \sim \mathcal{N}(0,1), \quad i = 1, ..., 101 \tag{17}
$$

It is provided in Figure 3(d). The physics-informed prior training does not see any samples from the OOD distribution.

Ground-truth samples $v(x,t)$ are computed using a Fourier spectral method with an implicit solver Kvaeron5 available in diffrax [64] with initial time step 0.00001, maximum time step of 0.0001, time interval $t \in [0,1]$, absolute tolerance 10^{-12}, and relative tolerance 10^{-12} for ODE time-stepping. We downsample the output in time to obtain a final shape $[101, 101]$ for training the posterior model.

The physics-informed loss function can be written as

$$
\begin{aligned}
L(\theta; \mathcal{D}^{pi}) = & \frac{1}{N_u N_c} \sum_{i=1}^{N_u} \sum_{j=1}^{N_c} \left(\frac{\partial \hat{\mathcal{A}}_\theta(u_i)(x_j, t_j)}{\partial t} + \hat{\mathcal{A}}_\theta(u_i)(x_j, t_j) \frac{\partial \hat{\mathcal{A}}_\theta(u_i)(x_j, t_j)}{\partial x} - \nu \frac{\partial^2 \hat{\mathcal{A}}_\theta(u_i)(x_j, t_j)}{\partial x^2} \right)^2 \\
& + \frac{1}{N_u N_b} \sum_{i=1}^{N_u} \sum_{j=1}^{N_b} \left(\hat{\mathcal{A}}_\theta(u_i)(0, t_j) - \hat{\mathcal{A}}_\theta(u_i)(1, t_j) \right)^2 + \left(\frac{\hat{\mathcal{A}}_\theta(u_i)(x, t_j)}{\partial x} \bigg|_{x=0} - \frac{\hat{\mathcal{A}}_\theta(u_i)(x, t_j)}{\partial x} \bigg|_{x=1} \right)^2 \\
& + \frac{1}{N_u N_I} \sum_{i=1}^{N_u} \sum_{j=1}^{N_I} (\hat{\mathcal{A}}_\theta(u_i)(x_j, 0) - u_i(x_j))^2
\end{aligned} \tag{18}
$$

where $\nu = 0.01$, $\hat{\mathcal{A}}_\theta$ is the neural operator model, $N_c = 128 \times 128$, $N_b = 128$, and $N_I = 128$ correspond to the interior, boundary, and initial condition points, sampled uniformly each iteration. $N_u = 128$ is the batch size for input functions.

D.5 Desired optima

The out-of-distribution (OOD) and in-distribution (ID) desired optima for each PDE test problem are depicted in Figure 3 and Figure 4, respectively.

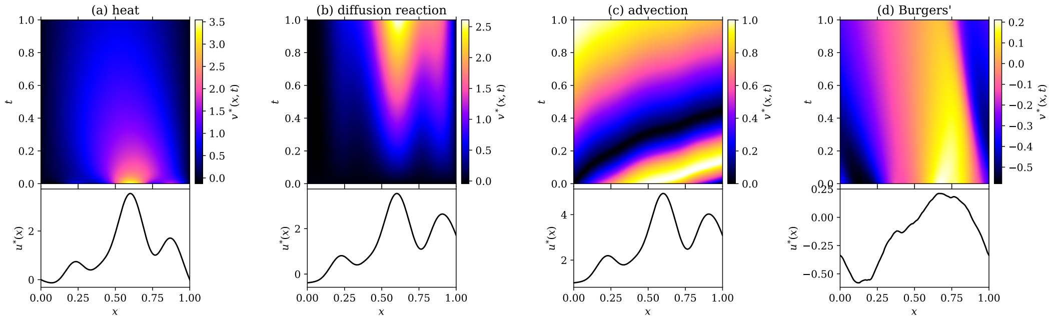


Figure 3: Out-of-distribution (OOD) optima u^* and corresponding $v^* = \mathcal{A}(u^*)$ for each PDE test problem.

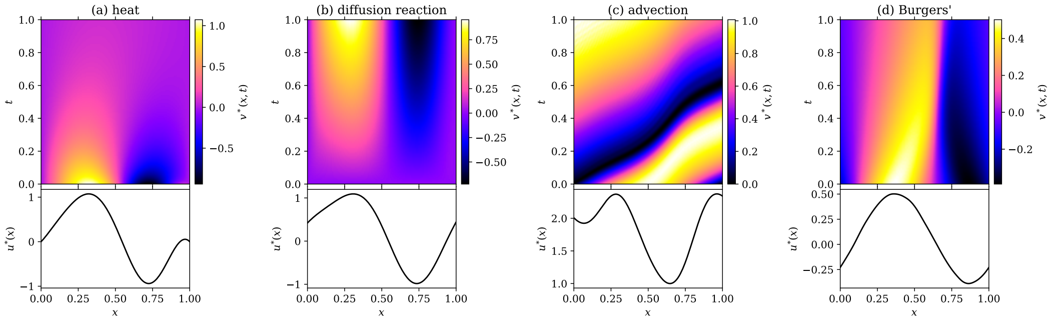


Figure 4: In-distribution (ID) optima u^* and corresponding $v^* = \mathcal{A}(u^*)$ for each PDE test problem.

D.6 Discussion of uncertainty quantification in PINNs and PINOs

There have been several recent investigations into quantifying uncertainty in PINN models and neural operator models. Uncertainty quantification in PINNs has been thoroughly reviewed in, for example, [31, 65–67]. Uncertainty quantification investigations for neural operators are relatively less developed than PINNs, but have been explored in [5, 66–72]. Our work makes use of PINO priors, deterministic and probabilistic, for deep operator ensemble and high-order Gaussian processes posteriors for Bayesian optimization. We are not aware of any other prior works that have investigated probabilistic or uncertainty-aware PINOs, for BO or otherwise.

E Implementation details

PDE test problem definitions. We consider four two-dimensional linear and nonlinear parametric partial differential equations (PDEs): the heat, diffusion-reaction, advection, and Burgers' equations. All problems are summarized in Table 1. As a reminder, provided a given input function u, the PDE solution can be written generally as $v = \mathcal{A}(u)$ for unknown operator $\mathcal{A}$, where $\mathcal{A}(u)$ satisfies the initial/boundary value problem expressed generally in eq. (1) and specifically in Table 1. Bayesian optimization seeks a neural operator approximation $\hat{\mathcal{A}} \approx \mathcal{A}$. We leverage knowledge of the initial value problems to give us a prior for BO using physics-informed training via eq. (3). The specific forms of the physics-informed loss functions for each problem may be found in Appendix D, Equations (13) to (16).

Optimization objectives. Our objective is to obtain $u^* = \text{argmin}_u\, f(\mathcal{A}(u))$ where f is a figure-of-merit functional, u is an input function to the PDE, and u^* is the unknown optimum. In composite optimization, we assume that f is known. For the purposes of testing PIBO, we choose f to be

$$f(v; v^*) = \|v - v^*\|^2_{L^2(\Omega_v)}, \tag{19}$$

where $v^* = \mathcal{A}(u^*)$ is a planted optimal output for some chosen u^*. Note that while v^* is provided, $\mathcal{A}$ and u^* are unknown to the optimization process. This scheme allows us to choose target optima that are either in-distribution (ID) or out-of-distribution (OOD) to the dataset provided during physics-informed training (i.e., $\mathcal{D}^{pi}$), in order to probe the efficacy of the physics-informed prior. The OOD and ID optimal u^* are sampled randomly from Gaussian processes with varying parameters, with some additional constraints (e.g., boundary conditions or periodicity). The OOD and ID u^*, and corresponding $v^* = \mathcal{A}(u^*)$, for each test problem are provided in Figure 3 and Figure 4 in the appendix. More information on how the ID and OOD u^* are sampled for each test problem are provided in Appendix D. Note that this paper assumes that u^* and v^* are noiseless and fully observed within the domains indicated in Table 1.

Discretization and problem size. While up to this point we have treated all inputs and outputs to the PDEs and operators as continuous functions, the functions must be discretized for numerical evaluation. For ground-truth PDE solutions, we sample $x \in [0, 1]$ and $t \in [0, 1]$ on a uniformly-spaced regular grid with $d \times d$ total points. For heat, diffusion-reaction, and advection we take $d = 128$, while for Burgers' we take $d = 101$. We may then encode u and solve for v along this spatiotemporal grid to collect $\mathbf{u} = [u(x_1), u(x_2), ..., u(x_d)] \in \mathbb{R}^d$, and $\mathbf{v} = [v(x_1, t_1), v(x_2, t_1), ..., v(x_d, t_1); v(x_1, t_2), ..., v(x_d, t_d)] \in \mathbb{R}^{d \times d}$. The figure-of-merit function eq. (19) is evaluated along this same discretization, i.e., $f(v; v^*) \approx f(\mathbf{v}; \mathbf{v}^*) = \frac{1}{d^2} \sum_{i,j} \|\mathbf{v}_{ij} - \mathbf{v}^*_{ij}\|^2$. Descriptions of the ground-truth solvers for each test problem are provided in Appendix D.

Baselines. We consider several baseline models. These include standard baselines like random sampling (RS) and single-objective Gaussian process (SO-GP). We will also consider a composite high-order GP (CO-HOGP) [48] and a composite deep ensemble (CO-DE) without pre-training. Finally, the last baseline will be to directly exploit the pre-trained mean operator (E-PTMO). In other words, we obtain the physics-informed operator model $\hat{\mathcal{A}}_\theta$ via eq. (7), then directly run gradient descent over it to obtain a number of candidates in parallel $\{u_c : u_c \sim \text{argmin}_u f(\hat{\mathcal{A}}_\theta(u)), c = 1, ..., C\}$. Then, we evaluate the candidates on the ground-truth in the order indicated by the predicted figure-of-merit values, e.g., $y_i = f(\mathcal{A}(u'_i))$ for $u'_i \in \text{argsort}(f(\hat{\mathcal{A}}_\theta(u_1)), ..., f(\hat{\mathcal{A}}_\theta(u_C)))$.

Models and hyperparameters. For all Gaussian process (GP and HOGP) models, we use an RBF kernel. For pre-trained physics-informed prior models (PTMO and PTOE) and deep ensemble (DE) models, we use the separable operator network (SepONet) [7] as base model, with $m = 20$ ensemble members when applicable. All hyperparameters, optimization details, and physics-informed training considerations are provided in Appendix F. Ablation studies with alternate kernels, base models, and number of ensemble members are provided in Appendix H.

(Physics-informed) Bayesian optimization and acquisition For all optimizations in the main text, we initialize the posterior models with a single sample $\mathcal{D}^{sv} = \{(u_0, v_0)\}$, sampled independently for each optimization trial, but reused for initialization of each method. We only accept a single new sample (i.e., $q = 1$) per iteration $\mathcal{D}^{sv} \leftarrow \mathcal{D}^{sv} \cup \{(u_c, v_c)\}$ for candidate u_c, under the assumption that ground-truth samples are expensive. We perform up to 30 BO iterations, with 10 independent trials (unique random seeds) per method. We use logarithmic expected improvement (LOGEI) as our default acquisition function. Ablation studies with alternative acquisition functions and number of accepted candidates are provided in Appendix H. For pre-training of physics-informed prior models, we sample a dataset of $N_u = 5,000$ input functions, e.g., $\mathcal{D}^{pi} = \{(u_i)_{i=1}^{5,000}\}$. Each $u \in \mathcal{D}^{pi}$ is always sampled "in-distribution (ID)"; please see details in Appendices D and F. All results in the main text utilizing the pre-trained physics-informed mean operator (PTMO) and operator ensemble (PTOE) used 100,000 iterations of pre-training. Meanwhile, the PTMO with fine-tuning was pre-trained for 5,000 iterations, and subsequently, after each ground-truth evaluation, was fine-tuned for an additional 5,000 iterations for 155,000 iterations of physics-informed training in total. The PTMO/PTOE was trained independently for all 10 optimization trials. We will present additional results as a function of pre-train iterations in Section 3.

F Model and hyperparameter specifications

F.1 Model choices

For single-objective (SO) BO cases, we choose a Gaussian process with RBF kernel, which is the default SingleTaskGP model from BoTorch. (In Appendix H we present ablation studies for other choices of kernels.) Inputs are the discretized functions $\mathbf{u}$ of size d and outputs are predicted scalar figure-of-merit values $\hat{y} \approx f(\hat{v})$. Moreover, inputs are normalized and outputs are standardized. We use BoTorch's built-in PyTorch likelihood fit with learning rate of 0.1, cosine annealing learning rate schedule, and 10,000 iterations.

For composite BO, we use either a higher-order GP (HOGP) or deep ensemble (DE). For the HOGP, we use an RBF kernel. We use the Matheson's rule sampling scheme with default settings from BoTorch. Inputs are the discretized functions $\mathbf{u}$ of size d and outputs are the predicted PDE solutions $\hat{\mathbf{v}}$ of size $d \times d$. Inputs are normalized and outputs are standardized.

The primary base neural operator model for physics-informed priors and composite BO we use is the separable operator network (SepONet) [7]. The SepONet enjoys efficient evaluation of spatiotemporal partial derivatives by leveraging an architecture inspired by separation of variables. Unless otherwise specified, for any single model, branch nets and trunk nets use 5 hidden layers with 64 hidden units per layer and output rank 64, and gelu activation functions. The branch net has an input dimension of d while the (two) trunk nets have input dimension of 1.

F.2 Physics-informed training

For both PTMO and PTOE strategies, we set the input function batch size to 128. Physics-informed training also requires the choice of collocation points for the numerical evaluation of the physics-informed residual integrals in eq. (3). We sample x and t in both the interior of the domain and its boundary independently each iteration using a uniform stratified sampler to approximate the integrals by Monte Carlo integration. Note that x and t are sampled separately uniformly 128 times to obtain two 128-length vectors and then meshed to obtain $N_c = 128 \times 128$ collocation point samples per iteration. Boundary points and initial condition points are sampled independently with 128×1 points per axis per iteration. We use a learning rate of 10^{-4} with a warm restart cosine decay learning rate scheduler in all cases. We vary the number of iterations used in pre-training. For PTOE, we default to $m = 20$ ensemble members.

F.3 Bayesian optimization hyperparameters

For SO-GP and PTMF-SO-GP we use the default BoTorch acquisition optimizer (L-BFGS-B), with 2000 raw samples and 128 restarts, applied to the analytic acquisition function. For CO-HOGP and PTMO-CO-HOGP we use 512 raw samples and 8 restarts, chosen to maintain computational tractability.

When ensembling, we default to $m = 20$ models. Ablation studies with different choices of m are shown in Appendix H. We use a custom acquisition routine that uses the AdamW optimizer with weight decay 10^{-8}, learning rate of 10^{-4}, and warm start cosine decay scheduler. We sample 128 samples uniformly and optimize all of them to find 128 candidates. For finetuning (PTMF-CO-DE-FT), we add all 128 candidates to the physics-informed dataset $\mathcal{D}^{pi} \leftarrow \mathcal{D}^{pi} \cup \{(u_c)_{c=1}^{128}\}$.

F.4 Hardware environment

All experiments utilizing GP models were run on an internal CPU cluster with 224 nodes and 6TB memory. All experiments utilizing HOGP models were run on 80GB A100 GPUs. Pre-trained physics-informed neural operator priors (PTMO) for GP and HOGP were trained on GPU (indicated below) and then loaded on CPU for use with PTMO-SO-GP and loaded on GPU for use with PTMO-CO-HOGP.

Out-of-distribution (OOD) experiments For out-of-distribution (OOD) experiments utilizing deep ensembles (DE), random sampling (RS), or exploited PTMO (E-PTMO), any given optimization was run on a single 80GB A100 GPU within an internal cluster.

In-distribution (ID) experiments For in-distribution (ID) experiments utilizing deep ensembles (DE), random sampling (RS), or exploited PTMO (E-PTMO), any given optimization was run on a single 32GB V100 GPU within an internal cluster.

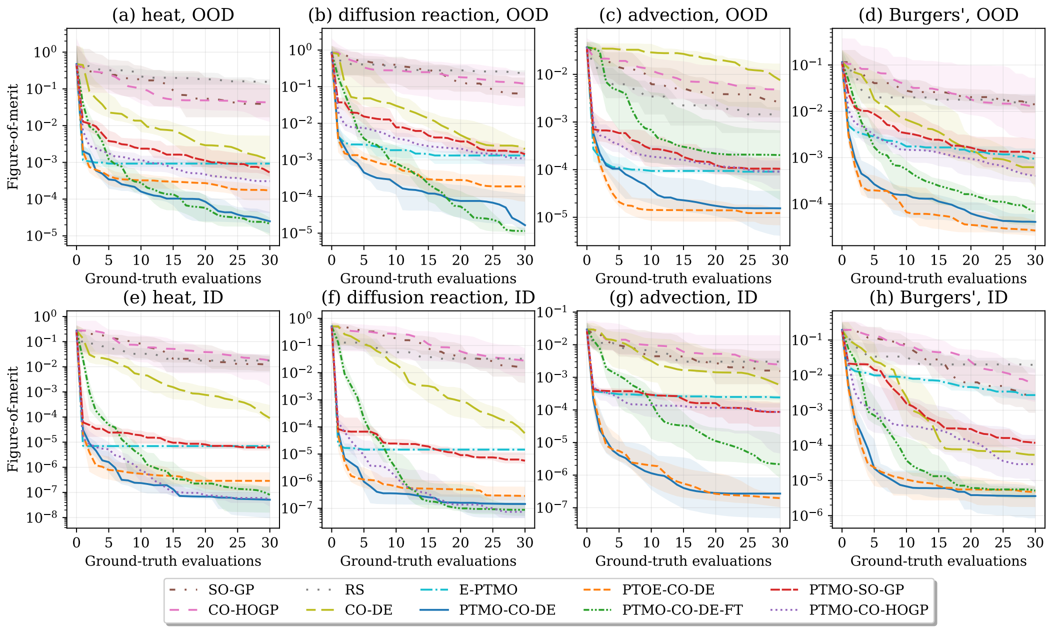


Figure 5: Out-of-distribution (OOD), (a)-(d), and in-distribution (ID), (e)-(h), test results for single-objective (SO) and composite (CO) Bayesian optimization using Gaussian process (GP), high-order GP (HOGP), and deep ensemble (DE) uncertainty models. We supply a pre-trained physics-informed mean operator (PTMO) or operator ensemble (PTOE) as an inductive bias to define a prior. We also explore fine-tuning (FT) the PTMO, random sampling (RS), and exploiting (E) the PTMO.

G Extended results

We start this appendix by presenting an extended version of Figure 1 including optimization curves for the fine-tuning (FT) and ensemble (PTOE)-based methods, see Figure 5.

G.1 Optimization run time

The optimization run time for the CO-DE, E-PTMO, PTMO-CO-DE, PTOE-CO-DE, and PTMO-CO-DE-FT methods from the main text (originally from Figure 1) are presented in Figure 6, where each curve represents the figure-of-merit versus runtime after 30 ground-truth evaluations. In the E-PTMO, PTMO-CO-DE, and PTOE-CO-DE models, the prior was pre-trained with 100,000 physics-informed iterations. For the PTMO-CO-DE-FT (fine-tuned) results, the prior was pre-trained with 5,000 physics-informed iterations, and then trained for an additional 5,000 physics-informed iterations after each new ground-truth evaluation. As discussed in Appendix F.4, the out-of-distribution (OOD) and in-distribution (ID) experiments were conducted on an 80GB A100 GPU and a 32GB V100 GPU, respectively. The type of GPU is clarified in the x-axis of each problem instance, which explains the roughly factor 2-3 difference in overall runtime between the OOD and ID problems. Nevertheless, the relative results between different models are consistent.

We observe that the CO-DE and and PTMO-CO-DE-FT methods allow for immediately exploitation starting at $t \approx 0$ seconds. Meanwhile, the E-PTMO, PTMO-CO-DE, and PTOE-CO-DE have more substantial upfront pre-training costs. Nevertheless, the pre-training cost is generally only a fraction of the total PIBO runtime after 30 ground-truth evaluations. For example, the PTMO-CO-DE pre-training represents only about $0.25\times$ the total runtime. The PTOE-CO-DE has about a factor 3 increase in pre-training time despite using the same number of iterations. This is a result of training the $m = 20$ ensemble members in parallel. After pre-training, the PTMO/PTOE models exhibit rapid reduction in the figure-of-merit with high ground-truth sampling efficiency, leading to the PTMO-CO-DE and PTMO-CO-DE-FT exhibiting an optimal cost balance between model training time, sample efficiency, and figure-of-merit performance. Furthermore, if pre-trained priors are recycled for multiple optimizations, the upfront pre-training cost is arguably amortized.

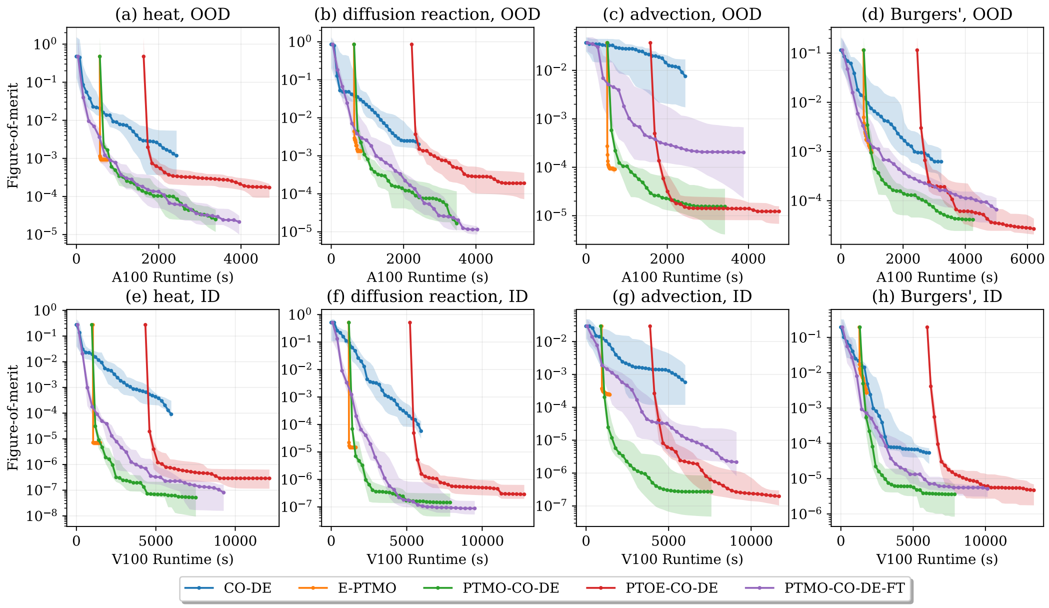


Figure 6: Figure-of-merit as measured in GPU runtime for the deep ensemble and E-PTMO PIBO methods, including PTMO/PTOE pre-training time.

Note that we omit the RS, SO-GP, and PTMO-SO-GP due to poor sample efficiency and final figure-of-merit. We omitted the HOGP models here due to their optimization runtime and performance in terms of figure-of-merit were far inferior to the corresponding DE models. On GPU the HOGP optimizations ran for approximately 6 hours (21,600 seconds).

G.2 Performance of physics-informed prior models versus number of pre-train iterations

Comprehensive investigations into the performance of the E-PTMO, PTMO-CO-DE, PTOE-CO-DE, and PTMO-SO-GP methods versus the number of physics-informed pre-training iterations for all test problems are provided in Figures 7 to 10, respectively. The results from Figure 8 were condensed and presented in Figure 2 of Section 3 in the main manuscript to demonstrate monotonic scaling of figure-of-merit and sampling efficiency as pre-train iterations are increased. As discussed there, all methods relying on pre-trained physics-informed priors generally benefit in terms of sample efficiency and figure-of-merit as pre-training effort is increased. Nevertheless, the deep ensemble posterior allows for further exploration and refinement over the E-PTMO, especially on OOD data. The CO-DE, SO-GP, and CO-HOGP models are also included where applicable, for reference.

H Ablation studies

In this appendix, we provide comprehensive ablation studies. In all studies, we consider the "advection, OOD" test problem only, presented in the main text results in Figure 1(c) with optimum provided in Figure 3(c). By and large, we find that PIBO is robust against making different choices for the acquisition function (Figure 11), the number of data points acquired per optimization iteration (Figure 12), the kernel type when using a Gaussian process based posterior predictive (Figure 13), and the number of ensemble members (Figure 14a) and the choice of physics-informed neural operator model (Figure 14b) when leveraging a deep ensemble posterior predictive.

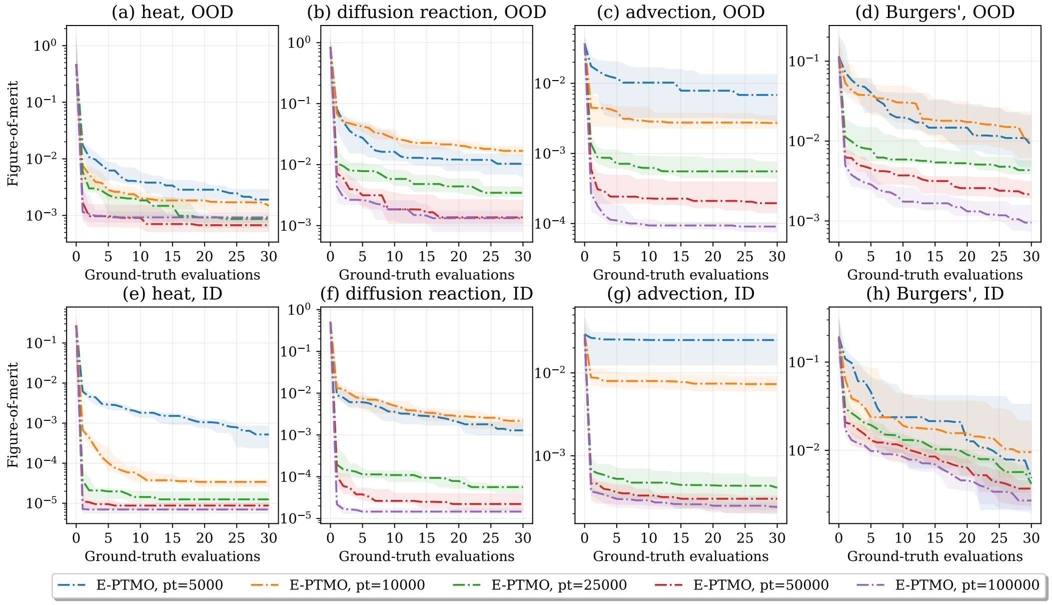


Figure 7: Effect of number of PTMO pre-train iterations on E-PTMO performance.

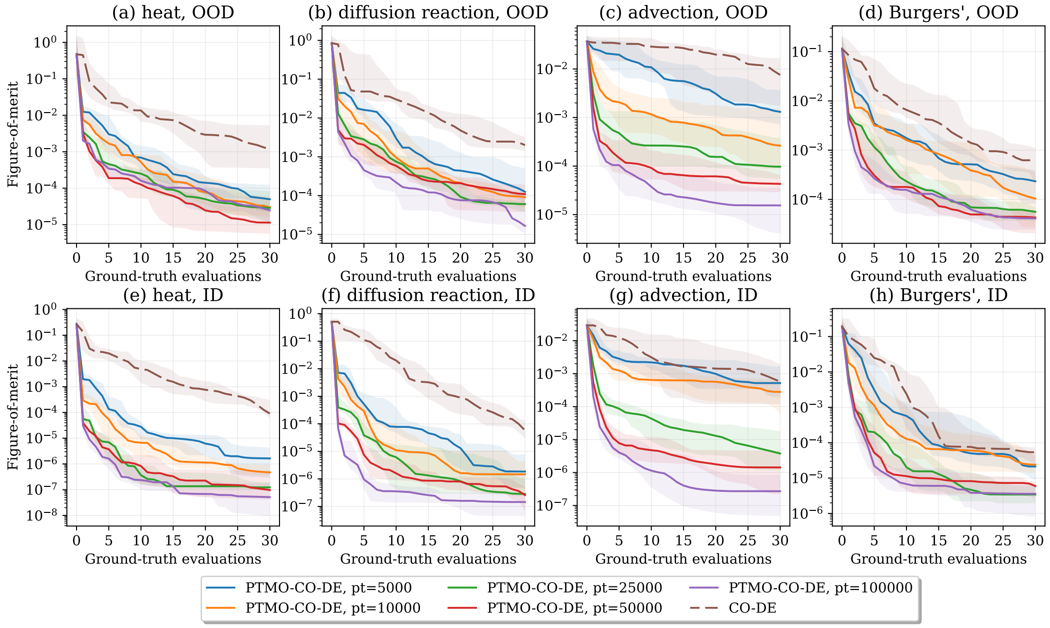


Figure 8: Effect of number of PTMO pre-train iterations on PTMO-CO-DE performance.

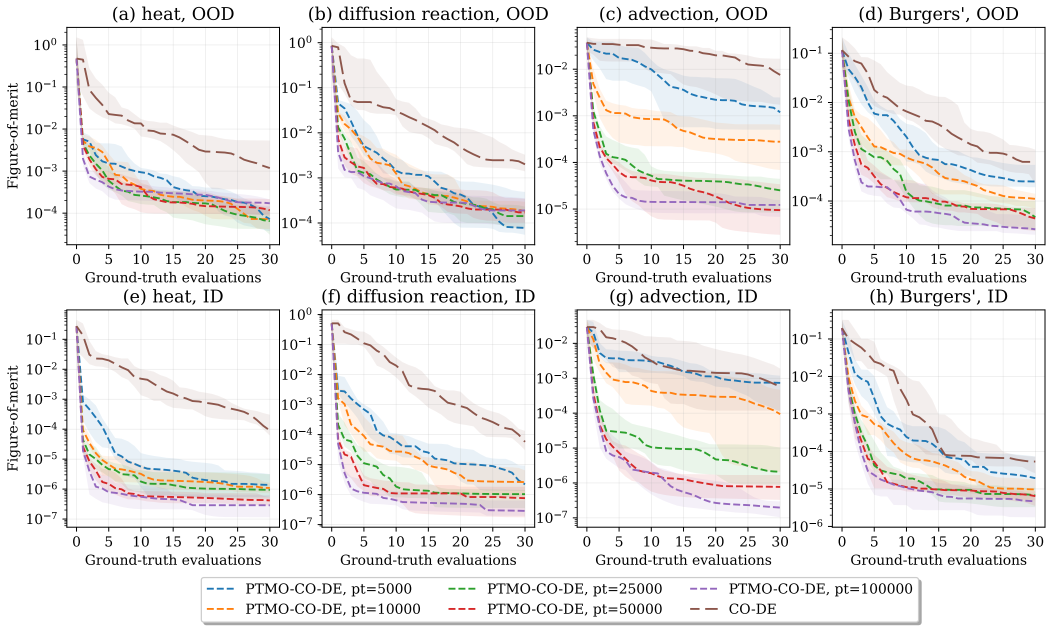


Figure 9: Effect of number of PTOE pre-train iterations on PTOE-CO-DE performance.

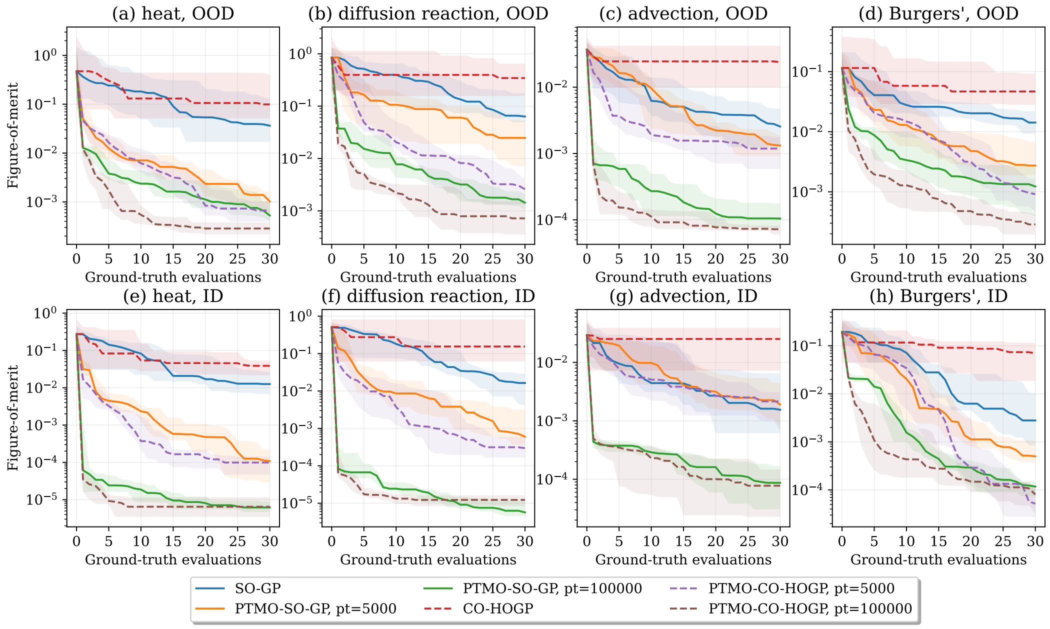
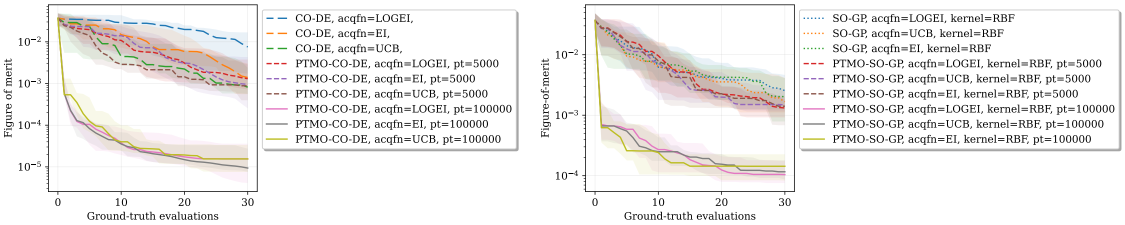


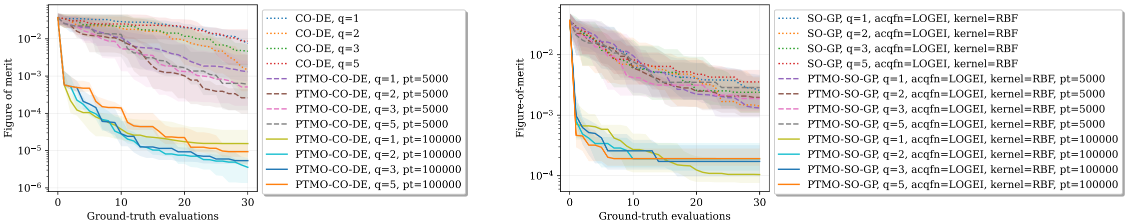
Figure 10: Effect of number of PTMO pre-train iterations on PTMO-SO-GP and PTMO- performance.



(a) Deep ensemble based posterior predictives

(b) Gaussian process based posterior predictives

Figure 11: Ablation study results for different acquisition functions.



(a) Deep ensemble based posterior predictives

(b) Gaussian process based posterior predictives

Figure 12: Ablation study results for number of acquired candidates q per BO iteration.

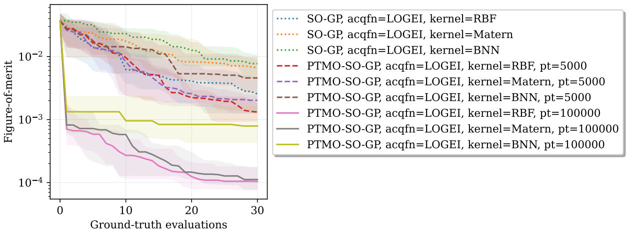
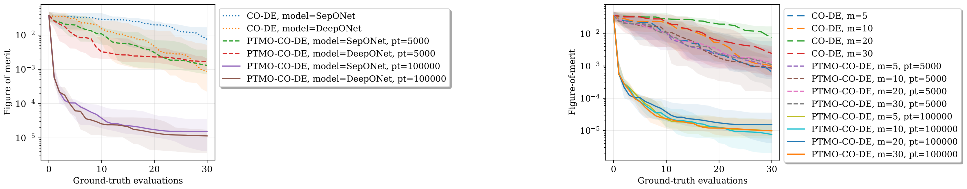


Figure 13: Ablation study results for different kernel types for GP based posterior predictives.



(a) Model type

(b) Number of ensemble members

Figure 14: Ablation study results for model type (left) and number of ensemble members (right) for deep ensemble based posterior predictives.