Workshop at the 7th Symposium on Advances in Approximate Bayesian Inference (non-archival), 2025 1–30

What Actually Matters for Materials Discovery: Pitfalls and Recommendations in Bayesian Optimization

Tristan Cinquin* *Tübingen AI Center, University of Tübingen*
Stanley Lo *University of Toronto*
Felix Strieth-Kalthoff *University of Wuppertal*
Alán Aspuru-Guzik *University of Toronto, Vector Institute*
Geoff Pleiss *University of British Columbia, Vector Institute*
Robert Bamler *Tübingen AI Center, University of Tübingen*
Tim G. J. Rudner *New York University*
Vincent Fortuin *Helmholtz AI, TU Munich*
Agustinus Kristiadi *Vector Institute*

1. Introduction

Materials discovery underpins advances in many fields (Tamasi et al., 2022; Strieth-Kalthoff et al., 2024a), but involves time-consuming and resource-intensive lab work (Abolhasani and Kumacheva, 2023). Bayesian optimization (BO; Močkus, 1975; Garnett, 2023) is an encouraging method to focus this effort towards promising candidates (Tom et al., 2024) by using a probabilistic surrogate model to approximate the objective (physical measurements) and to balance exploration and exploitation across vast molecular design spaces.

Despite BO's goal of minimizing costly evaluations, many previous works do not discuss the iterative nature of method design and model selection during which data is already used (Riquelme et al., 2018; Li et al., 2024a). In real world settings, optimizing the surrogate model is impossible as it requires the costly experimental data that BO aims to minimize and thus requires domain-specific understanding of the features and surrogate design space. We systematically investigate this design space for BO in materials discovery, examining surrogate models, hyperparameters, molecular features, and fine-tuning strategies, and aim to provide a concise overview of the design choices and to identify the major factors of BO performance. Through empirical studies on eight real-world tasks, we identify several key pitfalls in current methodologies, challenging some of the current de-facto wisdom (see Figure 1): (i) GPs with stationary covariance functions can perform poorly if not carefully initialized; and, **well-initialized GPs typically outperform BNNs**; (ii) **BNNs** exhibit high sensitivity to hyperparameter configurations, but this **can be improved by using GP-based functional priors**; (iii) **Learned** and generic hand-crafted features **outperform expert-designed features**; and (iv) **Simple (non-Bayesian) fine-tuning** of features derived from foundation models with new data at each BO iteration **significantly enhances performance**. Based on these findings, we formulate a set of recommendations for practitioners and provide a course correction for future research on this topic.

* Correspondance to: tristan.cinquin@uni-tuebingen.de

© T. Cinquin *et al.*

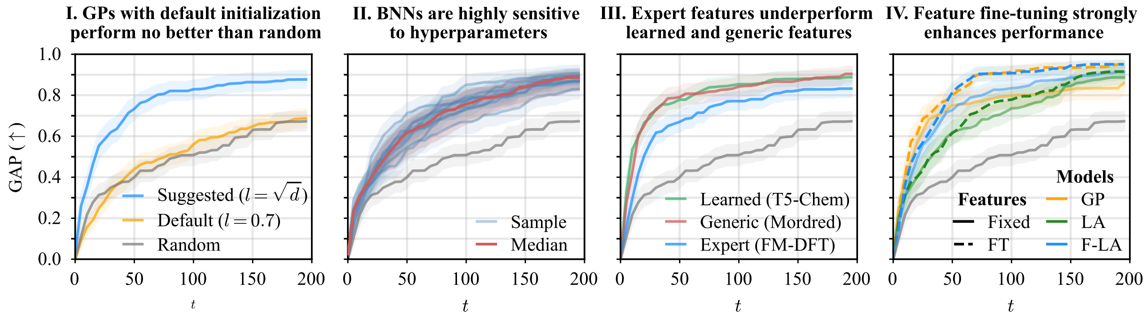

Figure 1: We find that (I) GPs are sensitive to some hyperparameters, but (II) BNNs even more so, and well-initialized GPs generally outperform BNNs. (III) Foundation model features are often better than hand-crafted ones, especially (IV) when additionally fine-tuned.

2. Experimental Setup

This section outlines datasets, surrogates, features and metrics, with details in Appendix C.

Dataset Our experiments span six materials discovery tasks from Kristiadi et al. (2024) and two drug discovery tasks from Graff et al. (2021). Materials tasks optimize redox potential (REDOXMER) and solvation energy (SOLVATION) of battery electrolytes (Agarwal et al., 2021), the fluorescence oscillator strength of lasers (LASER; Strieth-Kalthoff et al., 2024c), the power conversion efficiency (PCE) of photovoltaic materials (PHOTOVOLTAICS; Lopez et al., 2016), and the $\pi - \pi^*$ transition wavelength of organic photoswitches (PHOTOSWITCH; Griffiths et al., 2022). Drug discovery tasks (KINASE, AMPC, D4) minimize docking scores.

Surrogate models We use GP and BNN surrogates. GPs use a Tanimoto kernel for fingerprint features and Matérn-5/2 otherwise (Snoek et al., 2012; Griffiths et al., 2023). BNNs use a two-layer network with a linearized Laplace approximate posterior (Immer et al., 2021). We also use FSP-Laplace (Laplace BNN with a GP prior; Cinquin et al., 2024) and a Bayesian fine-tuned MolFormer with Laplace-approximated LoRA weights (Kristiadi et al., 2024). We use the Thompson sampling (Thompson, 1933) acquisition function.

Features We use three feature types: (i) data-driven (MolFormer and T5-Chem; Ross et al., 2022; Christofidellis et al., 2023), (ii) generic hand-crafted (Morgan fingerprints and Mordred descriptors; Morgan, 1965; Moriwaki et al., 2018), and (iii) expert hand-crafted (density functional theory (FM-DFT) properties, force-field energy and conjugation degree).

Metrics We measure BO performance via the optimal value over time and the GAP score (Jiang et al., 2020), which converts objectives to maximization tasks with values in $[0, 1]$ and averages across all tasks. Figures report means and standard errors over five random seeds.

3. An In-Depth Empirical Study of BO for Materials Discovery

3.1. All you need is a well-initialized GP surrogate

We first analyze key design choices for GP surrogates. We use MolFormer features as they were shown to perform best by Kristiadi et al. (2024). Starting from a reference configuration

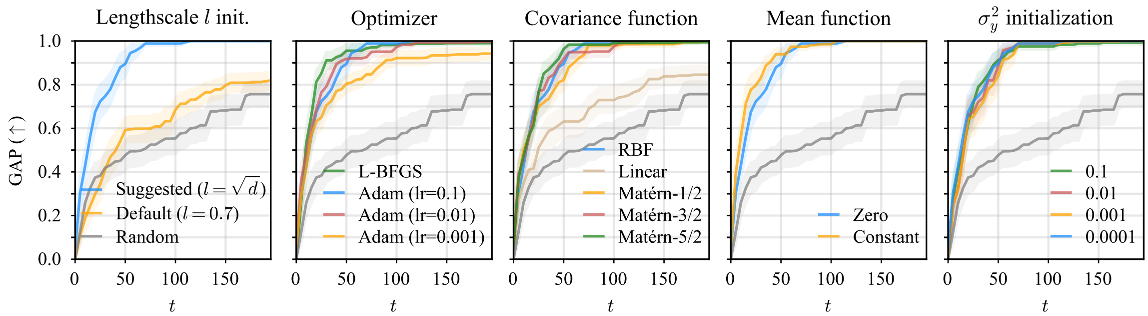

Figure 2: GAP metric for Gaussian process surrogates across design choices (for MolFormer features). Lengthscale initialization is crucial, and the choice of optimizer and covariance function matter to some extent, while the mean function and observation noise do not.

(Figure 2, blue line), we perform an ablation study varying one factor at a time. GAP scores are presented in Figure 2 and best reward plots in Figure D.12.

Proper lengthscale initialization is key. For stationary covariance functions, using GPyTorch's default lengthscale initialization ($l = 0.691$) for marginal likelihood optimization lead to poor results, barely outperforming random selection. Following Hvarfner et al. (2024), initializing $l \propto \sqrt{d}$ where d is the feature dimension, enables stationary covariance functions to maintain meaningful correlations over long ranges and significantly improves performance.

The choice of optimizer matters. We find that the optimizer used for marginal likelihood optimization of GP parameters significantly affects BO performance. While Adam (Kingma and Ba, 2015) is standard in GPyTorch, it converges slowly unlike quasi-Newton methods such as L-BFGS (Liu and Nocedal, 1989) which use line search. L-BFGS obtains higher BO objective values in fewer iterations than Adam. Furthermore, poorly tuned Adam learning rates strongly degrade BO performance, making L-BFGS a safer choice.

The prior covariance should be Matérn, but smoothness doesn't matter. Matérn covariance functions perform well across tasks, with little sensitivity to the smoothness parameter ν (RBF is Matérn with $\nu \to \infty$), while linear covariance performs poorly.

Mean function & observation noise initialization do not matter much. We standardize targets when training the surrogates, a zero mean function should be the correct model specification. Indeed, we find that using a constant or zero mean function does not make a difference. Observation noise initialization also has no significant impact.

3.2. Your BNN is probably not well tuned

Second, we explore key BNN design choices, evaluating the impact of individual components using a reference configuration (blue line, Figures D.4 to D.7) and performing an ablation.

BNNs with weight space priors are particularly sensitive to hyperparameters. We find that BNNs are highly sensitive to hyperparameters, particularly to activation functions, learning rates, and weight decay (see Laplace, Figures 3 and D.4). Observation noise initialization, however, has little effect. No single setup is optimal across all tasks—e.g., tanh works best for AmpC but worst for Redoxmer and Solvation (see Figure D.13).

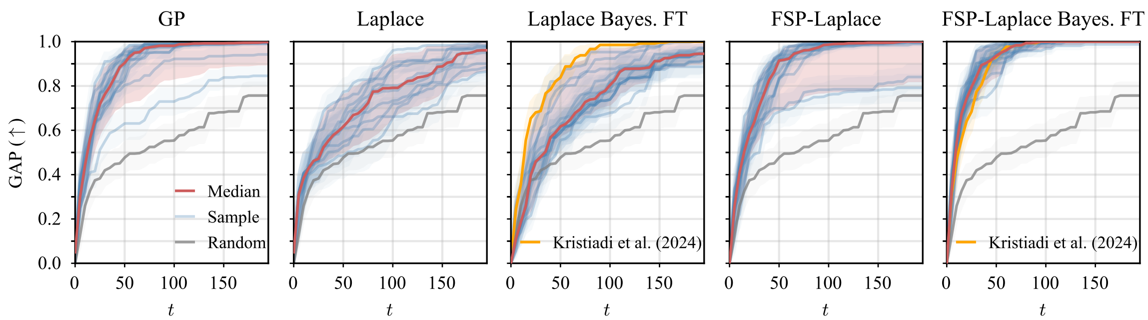

Figure 3: GAP metrics for GP, Laplace, Laplace Bayesian LoRA, FSP-Laplace and FSP-Laplace Bayesian LoRA surrogates. Each blue line is the GAP score for one instance of hyperparameters, the red line is the median, and the red interval is the $[0.05, 0.95]$ inter-quantile range. GP generally outperform BNNs and are less sensitive to hyperparameters. This behavior can be recovered in BNNs using GP priors with FSP-Laplace.

BNNs with GP priors are more robust. We here initialize the lengthscales as recommended in Section 3.1 and use L-BFGS for GP hyperparameter tuning. Unlike standard BNNs (i.e., Laplace), BNNs with GP priors using FSP-Laplace significantly improves robustness of BO performance across hyperparameters (Figure 3). However FSP-Laplace remains sensitive to learning rates and underperforms with linear covariance (Figure D.6), which is similarly ineffective for GPs. No single setup is optimal across all tasks (Figure D.14).

Comparing surrogate models. The sensitivity of surrogates to hyperparameters complicates comparisons, as better-performing configurations may remain undetected. GPs are robust when lengthscale initialization and optimizer are properly set. In contrast, Laplace BNNs are highly sensitive and mostly underperform GPs. FSP-Laplace improves performance relative to Laplace and is competitive with the best GP surrogate models.

3.3. Expert features underperform general features

Generic and data-driven features outperform expert features. Hand-crafted generic (fingerprints, Mordred) and data-driven (MolFormer, T5-Chem) features outperform hand-crafted expert features (FM-DFT, force field energy, degree of conjugation) on average across surrogates (see Figure 4). Surprisingly, expert features perform poorly even where expected to be informative (e.g., force field energy for SOLVATION, KINASE; Figures D.8 to D.10)).

Feature performance varies by task. Mordred is the most robust, followed by MolFormer and T5-Chem. Feature concatenation rarely degrades BO performance and mostly matches the best individual feature, indicating limited complementarity. For GP and FSP-Laplace it also reduces variability across task, making "all features" a consistent top performer, likely due to the automatic-relevance determination in the GP prior which selects revelent features.

ROGI-XD generally correlates with average cumulative regret but not task difficulty. The roughness index (ROGI-XD; see Appendix A.5; Aldeghi et al., 2022) measures objective function variability and correlates with average cumulative regret (Pearson correlation ≥ 0.5 on 5 of 8 tasks; Figure D.2), but it does not consistently capture BO

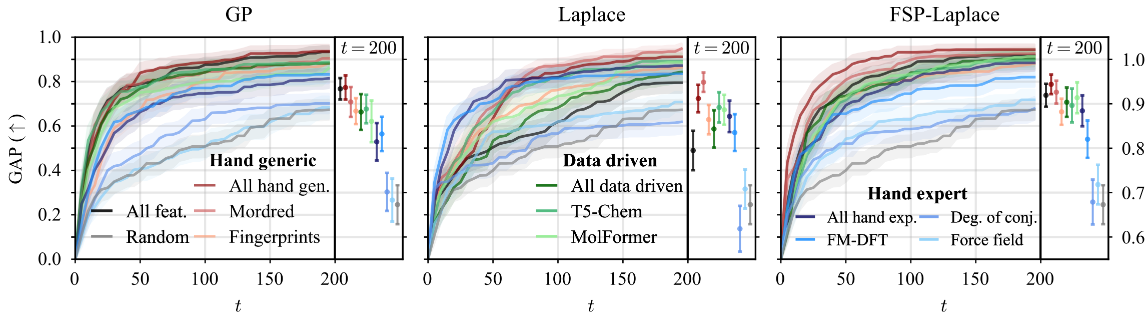

Figure 4: GAP score for GP, Laplace and FSP-Laplace surrogates across different types of features. We find that generic hand-crafted (fingerprints, Mordred) and data-driven (MolFormer, T5-Chem) features perform better on average than hand-crafted expert features (FM-DFT, force field, degree of conjugation). Concatenating a group of features generally performs as well as the best feature in the group. The shared legend is split across panels.

difficulty nor sensitivity to hyperparameters. For instance, D4, the hardest task, has among the lowest ROGI scores, while Photovoltaics, which is challenging, has the highest.

3.4. Fine-tuning has a positive impact on performance

Since ROGI-XD correlates with cumulative regret, and Graff et al. (2023) shows that fine-tuning reduces ROGI, we investigate LoRA and Bayesian LoRA fine-tuning for BO. At each iteration, LoRA fine-tuning (FT) updates the MolFormer model h_ϕ using observations $\mathcal{D}_t$, then generates a new dataset $\mathcal{F}_t$ of features using h_ϕ, later used to compute the surrogate's posterior $p(g \mid \mathcal{F}_t)$ (see Algorithm 1). Bayesian LoRA fine-tuning (Bayes. FT; Yang et al., 2024; Kristiadi et al., 2024) directly uses a fine-tuned MolFormer as a surrogate model, thus jointly training the LoRA and regression head, and incorporates posterior uncertainties via a Laplace approximation on *both* (vs. just g in LoRA FT). Bayesian LoRA FT is more expensive due to posterior estimation over LoRA weights.

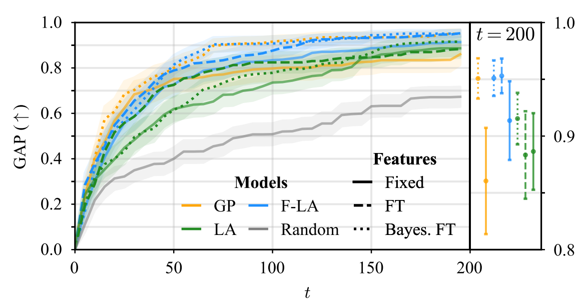

Figure 5: GAP score for GP and BNN surrogates with fixed and fine-tuned MolFormer features. Fine-tuning significantly improves performance, especially for GP and FSP-Laplace. We include a comparison with the best performing fixed features of Figure 4 (Fixed best).

Fine-tuning strongly improves BO performance. LoRA FT consistently improves BO performance and reduces variance over fixed features (see Figure 5). Fine-tuned GPs significantly outperform standard GPs, especially on challenging datasets (AmpC, D4; Figure D.11). FSP-Laplace also benefits substantially, while Laplace sees smaller gains. Bayesian LoRA FT performs comparably to LoRA FT overall, improving in some cases (e.g., FSP-Laplace on Redoxmer) but underperforming in others (e.g., Photoswitch).

Fine-tuning improves robustness to hyperparameters. Bayesian LoRA FT reduces sensitivity to hyperparameters compared to standard Laplace models (see Figure 3). Learning rate remains critical, but weight decay, activation functions, and LoRA rank have little effect (see Figure D.5). Using a GP prior further enhances robustness by reducing sensitivity to covariance (Figures D.7 and D.16), though learning rate sensitivity persists. Notably, hyperparameters from Kristiadi et al. (2024) consistently perform well (orange line, Figure 3).

4. A Practical Recipe for Bayesian Optimization in Materials Discovery

For reliable BO, use Matérn GP surrogates. We recommend GP surrogates with Matérn covariance functions and automatic-relevance determination, as they are fast, reliable, and highly competitive (Figure 3). Initialize lengthscales to $l \propto \sqrt{d}$, covariance function variance to $\sigma_f^2 = 1$, and use a zero mean function. While it had little impact in our experiments, observation noise σ_y^2 can follow an SNR-based heuristic (Deisenroth et al., 2020). Optimize hyperparameters with L-BFGS with the `strong_wolfe` line search strategy, using constraints to prevent extreme lengthscale values.

Algorithm 1 A Recipe for effective materials discovery with BO using LoRA feature fine-tuning

Require: Molecule set $\mathcal{X}$, objective f, *stationary* GP surrogate g, foundation model h_ϕ, acquisition function α, dataset $\mathcal{D}_1 = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$, budget T.

for $t = 1$ to T **do**
- Fine-tune h_ϕ on $\mathcal{D}_t$ with LoRA
- Build dataset $\mathcal{F}_t = \{(h_\phi(\boldsymbol{x}_i), y_i) \mid (\boldsymbol{x}_i, y_i) \in \mathcal{D}_t\}$
  ▷ Standardize features & labels
- Compute posterior $p(g \mid \mathcal{F}_t)$
- Find $\boldsymbol{x}_{t+1} = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \alpha(p(g(h_\phi(\boldsymbol{x})) \mid \mathcal{F}_t))$
- Evaluate objective $y_{t+1} = f(\boldsymbol{x}_{t+1})$
- Augment dataset $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(\boldsymbol{x}_{t+1}, y_{t+1})\}$
- Remove $\boldsymbol{x}_{t+1}$ from molecule set $\mathcal{X} = \mathcal{X} \setminus \{\boldsymbol{x}_{t+1}\}$

end for

Return: Best molecule $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{D}_{T+1}} f(\boldsymbol{x})$

Improve BO performance by fine-tuning features. For improved BO performance, we recommend using data-driven features from a foundation model (e.g., MolFormer) and LoRA fine-tuning at every BO step (see Algorithm 1). This significantly improves performance over fixed features while remaining computationally affordable. If fine-tuning is not feasible, concatenating fingerprint and Mordred features is a strong alternative, though with higher variance (see Figure 4). Additional features can also be included since for GPs more features do not seem to hurt BO performance but rather improve on some tasks (see Figure 4). We also advise standardizing features before BO and normalizing labels before fitting surrogates.

5. Conclusion

In this work, we systematically analyzed key design choices of surrogate models and molecular features in Bayesian optimization for materials discovery. We found that (1) default-initialized GPs perform hardly any better than random search, (2) BNNs are highly sensitive to hyperparameters, (3) learned and generic molecular features outperform expert-designed features, and (4) simple fine-tuning of molecular representations significantly improves performance, making costly Bayesian fine-tuning unnecessary. Based on our results, we identified the design choices that matter for cost-effective BO in materials discovery, namely using a simple but well-initialized surrogate model with simple feature fine-tuning.

Bibliography

Milad Abolhasani and Eugenia Kumacheva. The rise of self-driving labs in chemical and materials sciences. *Nature Synthesis*, 2(6):483–492, 2023.

Garvit Agarwal, Hieu Doan, Lily Robertson, and Rajeev Assary. Discovery of energy storage molecular materials using quantum chemistry-guided multiobjective bayesian optimization. *Chemistry of Materials*, 33, 10 2021. doi:10.1021/acs.chemmater.1c02040.

Ankit Agrawal and Alok Choudhary. Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science. *Apl Materials*, 4(5), 2016.

Matteo Aldeghi, David E Graff, Nathan Frey, Joseph A Morrone, Edward O Pyzer-Knapp, Kirk E Jordan, and Connor W Coley. Roughness of molecular property landscapes and its impact on modellability. *Journal of Chemical Information and Modeling*, 62(19):4660–4671, 2022.

Nicholas H Angello, Vandana Rathore, Wiktor Beker, Agnieszka Wołos, Edward R Jira, Rafał Roszak, Tony C Wu, Charles M Schroeder, Alán Aspuru-Guzik, Bartosz A Grzybowski, et al. Closed-loop optimization of general reaction conditions for heteroaryl suzuki-miyaura coupling. *Science*, 378(6618):399–405, 2022.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3(null):397–422, March 2003. ISSN 1532-4435.

The GPyOpt authors. GPyOpt: A bayesian optimization framework in python. `http://github.com/SheffieldML/GPyOpt`, 2016.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020. URL `http://arxiv.org/abs/1910.06403`.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. Chemberta: large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020.

Dimitrios Christofidellis, Giorgio Giannone, Jannis Born, Ole Winther, Teodoro Laino, and Matteo Manica. Unifying molecular and textual representations via multi-task language modelling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Tristan Cinquin, Marvin Pförtner, Vincent Fortuin, Philipp Hennig, and Robert Bamler. FSP-laplace: Function-space priors for the laplace approximation in bayesian deep learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=83vxe8alV4`.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux–effortless Bayesian deep learning. In *NeurIPS*, 2021.

Marc Peter Deisenroth, Yicheng Luo, and Mark van der Wilk. A practical guide to gaussian processes, 2020. URL https://infallible-thompson-49de36.netlify.app.

Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

Roman Garnett. *Bayesian Optimization.* Cambridge University Press, 2023.

Jose Garrido Torres, Sii Hong Lau, Pranay Anchuri, Jason Stevens, Jose Tabora, Jun Li, Alina Borovika, Ryan Adams, and Abigail Doyle. A multi-objective active learning platform and web app for reaction optimization. *ChemRxiv*, 2022. doi:10.26434/chemrxiv-2022-cljcp.

David Graff, Eugene Shakhnovich, and Connor Coley. Accelerating high-throughput virtual screening through molecular pool-based active learning. *Chemical Science*, 12, 04 2021. doi:10.1039/D0SC06805E.

David E Graff, Edward O Pyzer-Knapp, Kirk E Jordan, Eugene I Shakhnovich, and Connor W Coley. Evaluating the roughness of structure–property relationships using pretrained molecular representations. *Digital Discovery*, 2(5):1452–1460, 2023.

Ryan-Rhys Griffiths, Jake L Greenfield, Aditya R Thawani, Arian R Jamasb, Henry B Moss, Anthony Bourached, Penelope Jones, William McCorkindale, Alexander A Aldrick, Matthew J Fuchter, et al. Data-driven discovery of molecular photoswitches with multioutput gaussian processes. *Chemical Science*, 13(45):13541–13551, 2022. doi:10.1039/D2SC04306H.

Ryan-Rhys Griffiths, Leo Klarner, Henry B. Moss, Aditya Ravuri, Sang Truong, Samuel Stanton, Gary Tom, Bojana Rankovic, Yuanqi Du, Arian Jamasb, Aryan Deshwal, Julius Schwartz, Austin Tripp, Gregory Kell, Simon Frieder, Anthony Bourached, Alex Chan, Jacob Moss, Chengzhi Guo, Johannes Durholt, Saudamini Chaurasia, Felix Strieth-Kalthoff, Alpha A. Lee, Bingqing Cheng, Alán Aspuru-Guzik, Philippe Schwaller, and Jian Tang. GAUCHE: A library for Gaussian processes in chemistry. In *NeurIPS*, 2023.

Riley Hickman, Priyansh Parakh, Austin Cheng, Qianxiang Ai, Joshua Schrier, Matteo Aldeghi, and Alán Aspuru-Guzik. Olympus, enhanced: Benchmarking mixed-parameter and multi-objective optimization in chemistry and materials science. *ChemRxiv*, 2023a.

Riley Hickman, Malcolm Sim, Sergio Pablo-García, Ivan Woolhouse, Han Hao, Zeqing Bao, Pauric Bannigan, Christine Allen, Matteo Aldeghi, and Alán Aspuru-Guzik. Atlas: A Brain for Self-driving Laboratories, September 2023b.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Carl Hvarfner, Erik Orm Hellsten, and Luigi Nardi. Vanilla Bayesian optimization performs great in high dimensions. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20793–20817. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hvarfner24a.html.

Florian Häse, Loïc M. Roch, Christoph Kreisbeck, and Alán Aspuru-Guzik. Phoenics: A bayesian optimizer for chemistry. *ACS Central Science*, 4(9):1134–1145, 2018.

Alexander Immer, Maciej Jan Korzepa, and M. Bauer. Improving predictions of bayesian neural nets via local linearization. In *International Conference on Artificial Intelligence and Statistics*, 2021. URL https://api.semanticscholar.org/CorpusID:221172984.

Shali Jiang, Henry Chai, Javier Gonzalez, and Roman Garnett. BINOCULARS for efficient, nonmyopic sequential experimental design. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4794–4803. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/jiang20b.html.

Donald Jones, Matthias Schonlau, and William Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 12 1998. doi:10.1023/A:1008306431147.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, and Vincent Fortuin. Promises and pitfalls of the linearized laplace in bayesian optimization, 2023. URL https://arxiv.org/abs/2304.08309.

Agustinus Kristiadi, Felix Strieth-Kalthoff, Marta Skreta, Pascal Poupart, Alan Aspuru-Guzik, and Geoff Pleiss. A sober look at LLMs for material discovery: Are they actually good for Bayesian optimization over molecules? In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 25603–25622. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/kristiadi24a.html.

Yucen Lily Li, Tim G. J. Rudner, and Andrew Gordon Wilson. A study of bayesian neural network surrogates for bayesian optimization. In *The Twelfth International*

Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=SA19ijj44B.

Yucen Lily Li, Tim GJ Rudner, and Andrew Gordon Wilson. A study of Bayesian neural network surrogates for Bayesian optimization. In *ICLR*, 2024b.

Qiaohao Liang, Aldair E Gongora, Zekun Ren, Armi Tiihonen, Zhe Liu, Shijing Sun, James R Deneault, Daniil Bash, Flore Mekki-Berrada, Saif A Khan, et al. Benchmarking the performance of Bayesian optimization across multiple experimental materials science domains. *npj Computational Materials*, 7(1):188, 2021.

Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(1–3):503–528, August 1989. ISSN 0025-5610.

Steven Lopez, Edward Pyzer-Knapp, Gregor Simm, Trevor Lutzow, Kewei Li, Laszlo Seress, Johannes Hachmann, and Alán Aspuru-Guzik. The harvard organic photovoltaic dataset. *Scientific Data*, 3, 09 2016. doi:10.1038/sdata.2016.86.

David John Cameron MacKay. Bayesian methods for adaptive models. 1992. URL https://api.semanticscholar.org/CorpusID:123141880.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, pages 13153–13164, 2019.

Alexander Mathiasen, Hatem Helal, Paul Balanca, Adam Krzywaniak, Ali Parviz, Frederik Hvilshøj, Blazej Banaszewski, Carlo Luschi, and Andrew William Fitzgibbon. Reducing the cost of quantum chemical data by backpropagating through density functional theory. *arXiv preprint arXiv:2402.04030*, 2024.

J. Močkus. *On Bayesian Methods for Seeking the Extremum*, pages 400–404. Springer Berlin Heidelberg, Berlin, Heidelberg, 1975. ISBN 978-3-662-38527-2. doi:10.1007/978-3-662-38527-2_55. URL https://doi.org/10.1007/978-3-662-38527-2_55.

Harry L. Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5: 107–113, 1965. URL https://api.semanticscholar.org/CorpusID:62164893.

Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. Mordred: a molecular descriptor calculator. *Journal of cheminformatics*, 10:1–14, 2018.

Emre Onal, Klemens Flöge, Emma Caldwell, Arsen Sheverdin, and Vincent Fortuin. Gaussian stochastic weight averaging for bayesian low-rank adaptation of large language models. In *Sixth Symposium on Advances in Approximate Bayesian Inference - Non Archival Track*, 2024. URL https://openreview.net/forum?id=LZrCBQBCzl.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

Mayk Caldas Ramos, Shane S Michtavy, Marc D Porosoff, and Andrew D White. Bayesian optimization of catalysts with in-context learning. *arXiv preprint arXiv:2304.05341*, 2023.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

Guillermo Restrepo. Chemical space: limits, evolution and modelling of an object bigger than our universal library. *Digital Discovery*, 1(5):568–585, 2022.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyYe6k-CW.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Skdvd2xAZ.

Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *Nature Machine Intelligence*, 4(12):1256–1264, 2022. doi:10.1038/s42256-022-00580-7.

Lars Ruddigkeit, Ruud Deursen, Lorenz Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52, 10 2012. doi:10.1021/ci300415d.

Stefan P. Schmid, Ella Miray Rajaonson, Cher Tian Ser, Mohammad Haddadnia, Shi Xuan Leong, Alan Aspuru-Guzik, Agustinus Kristiadi, Kjell Jorner, and Felix Strieth-Kalthoff. If optimizing for general parameters in chemistry is useful, why is it hardly done? In *AI for Accelerated Materials Design - NeurIPS 2024*, 2024.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

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

Eduardo Soares, Victor Shirasuna, Emilio Vital Brazil, Renato Cerqueira, Dmitry Zubarev, and Kristin Schmidt. A large encoder-decoder family of foundation models for chemical language. *arXiv preprint arXiv:2407.20267*, 2024a.

Eduardo Soares, Victor Shirasuna, Emilio Vital Brazil, Renato Cerqueira, Dmitry Zubarev, and Kristin Schmidt. A large encoder-decoder family of foundation models for chemical language, 2024b. URL `https://arxiv.org/abs/2407.20267`.

Felix Strieth-Kalthoff, Han Hao, Vandana Rathore, Joshua Derasp, Théophile Gaudin, Nicholas H Angello, Martin Seifrid, Ekaterina Trushina, Mason Guy, Junliang Liu, et al. Delocalized, asynchronous, closed-loop discovery of organic laser emitters. *Science*, 384 (6697):eadk9227, 2024a.

Felix Strieth-Kalthoff, Han Hao, Vandana Rathore, Joshua Derasp, Théophile Gaudin, Nicholas H Angello, Martin Seifrid, Ekaterina Trushina, Mason Guy, Junliang Liu, et al. Delocalized, asynchronous, closed-loop discovery of organic laser emitters. *Science*, 384 (6697):eadk9227, 2024b.

Felix Strieth-Kalthoff, Han Hao, Vandana Rathore, Joshua Derasp, Théophile Gaudin, Nicholas Angello, Martin Seifrid, Ekaterina Trushina, Mason Guy, Junliang Liu, Xun Tang, Masashi Mamada, Wesley Wang, Tuul Tsagaantsooj, Cyrille Lavigne, Robert Pollice, Tony Wu, Kazuhiro Hotta, Leticia Bodo, and Alán Aspuru-Guzik. Delocalized, asynchronous, closed-loop discovery of organic laser emitters. *Science (New York, N.Y.)*, 384:eadk9227, 05 2024c. doi:10.1126/science.adk9227.

Matthew J Tamasi, Roshan A Patel, Carlos H Borca, Shashank Kosuri, Heloise Mugnier, Rahul Upadhya, N Sanjeeva Murthy, Michael A Webb, and Adam J Gormley. Machine learning on a robotic platform for the design of polymer–protein hybrids. *Advanced Materials*, 34(30):2201809, 2022.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. ISSN 00063444. URL `http://www.jstor.org/stable/2332286`.

Gary Tom, Stefan P Schmid, Sterling G Baird, Yang Cao, Kourosh Darvish, Han Hao, Stanley Lo, Sergio Pablo-García, Ella M Rajaonson, Marta Skreta, et al. Self-driving laboratories for chemistry and materials science. *Chemical Reviews*, 124(16):9633–9732, 2024.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Adam X. Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=FJiUyzOF1m`.

Yunxing Zuo, Mingde Qin, Chi Chen, Weike Ye, Xiangguo Li, Jian Luo, and Shyue Ping Ong. Accelerating materials discovery with Bayesian optimization and graph deep learning. *Materials Today*, 51:126–135, 2021.

Appendix

Appendix A. Background

A.1. Materials discovery

Materials discovery is the daunting task of searching through the chemical space of 10^{200} materials (Restrepo, 2022) for candidates with desirable properties. Materials discovery generally requires the design, synthesis, purification, characterization, and testing of the candidate material, which involves a plethora of design parameters and observable metrics. Often, there is a lack of data and poor understanding of the structure-property relationships in materials discovery (Agrawal and Choudhary, 2016) which makes it challenging. As a consequence, the traditional approach of trial-and-error following human intuition or design-of-experiments (i.e. combinatorial search) is too slow and inefficient to explore the vast landscape (Shahriari et al., 2015; Tom et al., 2024). To remedy this, material scientists have turned to Bayesian optimization because of its data efficiency, versatility in black-box model selection, uncertainty quantification, and recent success in chemical optimization tasks (Tom et al., 2024; Angello et al., 2022).

A.2. Bayesian optimization

Let $f : \mathcal{X} \mapsto \mathcal{Y}$ be an unknown function that is expensive to evaluate. Bayesian Optimization (BO) is a sequential optimization method to find the maximum $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$. BO requires (i) a probabilistic surrogate model g that approximates the objective f, (ii) an observation model $p(y \mid g(\boldsymbol{x}))$ for the data, (iii) a prior $p(g)$ for the surrogate, and (iv) an acquisition function α which guides the selection of the evaluation points. At each iteration, α uses the surrogate model's posterior $p(g \mid \mathcal{D}_t) \propto p(\mathcal{D}_t \mid g)\, p(g)$ on past (potentially noisy) function evaluations $\mathcal{D}_t = \{(\boldsymbol{x}_i, y_i)\}$ to balance between *exploration* in regions where f is unobserved and *exploitation* in regions where f is known to yield promising candidates. When applied to materials discovery, the search space is a discrete set of molecules $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^M$, and we only need to evaluate α at all unlabeled candidate molecules to find its maximum. This simplifies BO and is embarrassingly parallel. We describe discrete BO in Algorithm 2.

While the default surrogate models remain Gaussian processes (GPs) (Rasmussen and Williams, 2006), there has been a recent interest in Bayesian neural networks (BNNs), which have shown promising results (Kristiadi et al., 2024). GPs $g \sim \mathcal{GP}(m, k)$ are defined by mean $m : \mathcal{X} \mapsto \mathcal{Y}$ and covariance functions $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ which allow to specify prior knowledge about the objective function f, and BNNs are defined by a neural network $g_{\boldsymbol{w}}$. Common choices of acquisition function α include Thompson sampling (Thompson, 1933), expected improvement (Jones et al., 1998), and the upper-confidence bound (Auer, 2003). Finally, material scientist typically use the Gryffin (Häse et al., 2018), GPyOpt (authors, 2016), BoTorch (Balandat et al., 2020), EDBO+ (Garrido Torres et al., 2022), and Atlas (Hickman et al., 2023b) Python packages for their materials discovery campaigns.

A.3. Bayesian Neural networks

Let $g_{\boldsymbol{w}} : \mathcal{X} \mapsto \mathcal{Y}$ be a neural network with parameters $\boldsymbol{w} \in \mathcal{W} \subset \mathbb{R}^p$. Given data $\mathcal{D}$, Bayesian neural networks are defined in terms of a likelihood $p(\mathcal{D} \mid \boldsymbol{w})$ and a weight-space prior $p(\boldsymbol{w})$.

Algorithm 2 Bayesian Optimization for materials discovery

Require: Molecule candidate set $\mathcal{X}$, objective function f, surrogate model g, acquisition function α, initial dataset $\mathcal{D}_1 = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$, budget T.

for $t = 1$ to T **do**
  Compute posterior $p(g \mid \mathcal{D}_t)$
  Find candidate $\boldsymbol{x}_{t+1} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \alpha(p(g \mid \mathcal{D}_t))$
  Evaluate objective $y_{t+1} = f(\boldsymbol{x}_{t+1})$
  Augment dataset $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(\boldsymbol{x}_{t+1}, y_{t+1})\}$
  Remove $\boldsymbol{x}_{t+1}$ from molecule set $\mathcal{X} = \mathcal{X} \setminus \{\boldsymbol{x}_{t+1}\}$
end for
Return: Best molecule $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathcal{D}_T} f(\boldsymbol{x})$

BNNs recast training as approximating the (intractable) posterior $p(\boldsymbol{w} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{w})\, p(\boldsymbol{w})$ thus naturally capturing epistemic uncertainty due to learning with finite data. Crucially, the posterior yields predictions consistent with the data within its support, while being uncertain far away from the data. Multiple approximate inference algorithms exist, either based on a set of samples (e.g., MCMC) or a parametric distribution (e.g., variational inference and Laplace approximations).

Laplace approximations. The Laplace (MacKay, 1992) approximates the neural network's posterior by a Gaussian $q(\boldsymbol{w} \mid \mathcal{D}) = \mathcal{N}\left(\boldsymbol{w}^*, \boldsymbol{\Lambda}^{-1}\right)$ where $\boldsymbol{w}^*$ is found by maximizing the log-posterior $\log p(\boldsymbol{w} \mid \mathcal{D})$ (i.e., standard regularized neural network training) and $\boldsymbol{\Lambda} = -\nabla^2_{\boldsymbol{w}} \log p(\boldsymbol{w} \mid \mathcal{D})\big|_{\boldsymbol{w}=\boldsymbol{w}^*}$ is the Hessian. To avoid computing prohibitively expensive Hessians of the neural network with respect to its parameters, it is common to first linearize the neural network around the posterior mean $\boldsymbol{w}^*$ $g^{\text{lin}}_{\boldsymbol{w}}(\boldsymbol{x}) = g_{\boldsymbol{w}^*}(\boldsymbol{x}) + \boldsymbol{J}_{\boldsymbol{w}^*}(\boldsymbol{x})(\boldsymbol{w} - \boldsymbol{w}^*)$ where $\boldsymbol{J}_{\boldsymbol{w}^*}(\boldsymbol{x}) = \nabla_{\boldsymbol{w}} g_{\boldsymbol{w}}(\boldsymbol{x})|_{\boldsymbol{w}=\boldsymbol{w}^*}$ is the Jacobian, before computing $\boldsymbol{\Lambda}$ under this approximation (Immer et al., 2021). The resulting posterior predictive is $p(g^{\text{lin}}_{\boldsymbol{w}}(\boldsymbol{x}) \mid \mathcal{D}) = \mathcal{N}\left(g_{\boldsymbol{w}^*}(\boldsymbol{x}), \boldsymbol{J}_{\boldsymbol{w}^*}(\boldsymbol{x})\boldsymbol{\Lambda}^{-1}\boldsymbol{J}_{\boldsymbol{w}^*}(\boldsymbol{x})^\top\right)$ thus adding an uncertainty estimate around the standard neural network's prediction $g_{\boldsymbol{w}^*}(\boldsymbol{x})$. Since the precision $\boldsymbol{\Lambda} \in \mathbb{R}^{p\times p}$ is typically prohibitively large, it is often approximated by its diagonal or a product of Kronecker factors (Ritter et al., 2018). Laplace approximations scale to large datasets and models, and have been shown to provide well-calibrated uncertainty estimates (Immer et al., 2021; Daxberger et al., 2021).

The Laplace also provides an approximation to the marginal likelihood allowing for model selection. Making the dependance of the prior on its hyperparameters $\boldsymbol{\theta}$ explicit, the marginal likelihood is approximated by

$$\log p(\mathcal{D} \mid \boldsymbol{\theta}) \approx \log p(\mathcal{D} \mid \boldsymbol{w}^*) + \log p(\boldsymbol{w}^* \mid \boldsymbol{\theta}) + \frac{p}{2}\log 2\pi - \frac{1}{2}\log\det\boldsymbol{\Lambda} \tag{A.1}$$

We can then maximize the marginal likelihood by gradient ascent to find the neural network's prior parameters.

Function-space priors for Laplace approximations. While the choice of prior strongly affects the resulting posterior distribution, meaningful prior beliefs over neural network weights are very difficult to specify due to non-interpretability of the weights. FSP-Laplace

(Cinquin et al., 2024) allows to specify meaningful prior beliefs using a GP prior within the framework of Laplace approximations. Specifically, the posterior approximation remains $q(\boldsymbol{w} \mid \mathcal{D}) = \mathcal{N}\left(\boldsymbol{w}^*, \boldsymbol{\Lambda}^{-1}\right)$ but the posterior mean $\boldsymbol{w}^*$ under the GP prior $g \sim \mathcal{GP}\left(m, k\right)$ is found by maximizing $R_{FSP}(\boldsymbol{w}) = \log p(\mathcal{D} \mid \boldsymbol{w}) - {}^1\!/_2 \, \|g_{\boldsymbol{w}} - m\|^2_{\mathbb{H}_k}$ where $\|\cdot\|_{\mathbb{H}_k}$ is the RKHS norm under the kernel k, and the posterior precision is given by $\boldsymbol{\Lambda} = -\left.\nabla^2_{\boldsymbol{w}} R_{FSP}(\boldsymbol{w})\right|_{\boldsymbol{w}=\boldsymbol{w}^*}$.

A.4. Molecular foundation models

Molecular foundation models are a class of large-scale neural networks pre-trained on vast datasets of molecular structures (e.g., SMILES) and their properties (Chithrananda et al., 2020; **?**; Soares et al., 2024a). These models leverage the transformer architecture, which has demonstrated strong performance in natural language processing and is increasingly applied to generate molecular data with robust accuracy Mathiasen et al. (2024).

The transformer architecture captures long-range dependencies in sequential data using K-head self-attention. Given an input sequence $\boldsymbol{X} \in \mathbb{R}^{T \times N}$, it computes

$$\begin{aligned} \boldsymbol{O} &= [\boldsymbol{H}_1, \ldots, \boldsymbol{H}_K]\boldsymbol{W}_O^\top \in \mathbb{R}^{T \times O} \\ \boldsymbol{H}_i &= \mathrm{softmax}\left(\frac{1}{\sqrt{D}}(\boldsymbol{X}\boldsymbol{Q}_i^\top)(\boldsymbol{X}\boldsymbol{K}_i^\top)^\top\right)(\boldsymbol{X}^\top \boldsymbol{V}_i) \in \mathbb{R}^{T \times D} \end{aligned}$$

where $[\ldots]$ is the column-wise stacking operator, $\boldsymbol{W}_O \in \mathbb{R}^{O \times KD}$ and $\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i \in \mathbb{R}^{D \times N}$ are linear projections. Composing a multi-head self-attention layer with normalization layers and residual connections yields a transformer module, many of which are then stacked to form a transformer model.

Molecular foundation models are pre-trained using self-supervised masked language modeling on large generic datasets such as MoleculeNet Wu et al. (2018). In this approach, some tokens in the input sequence are randomly masked, and the model is trained to reconstruct these masked tokens. Foundation models are then fine-tuned to specific applications where typically less data is available. Due to their very large size, fine-tuning all the model parameters is prohibitively expensive. As a solution, parameter-efficient fine-tuning (PEFT) only updates a subset of the parameters, keeping the rest frozen (Houlsby et al., 2019; Hu et al., 2022).

Low-rank adaptation (LoRA) (Hu et al., 2022) is a popular PEFT method that introduces low-rank updates to a subset of parameters (typically attention weights), significantly reducing computational and memory requirements. Given a frozen pre-trained weight $\boldsymbol{W}^* \in \mathbb{R}^{D \times N}$, LoRA adds low-rank factors $\boldsymbol{A} \in \mathbb{R}^{Z \times N}$ and $\boldsymbol{B} \in \mathbb{R}^{Z \times D}$ in a new weight $\boldsymbol{W} = \boldsymbol{W}^* + \boldsymbol{B}^\top \boldsymbol{A}$. Z is typically small (e.g., $Z = 8$) such that the factors only introduce a small amount of new parameters.

Bayesian LoRA approximates the neural network's posterior with respect to the LoRA weights either using the Laplace approximation (Yang et al., 2024) or SWAG (Onal et al., 2024; Maddox et al., 2019).

A.5. ROGI-XD

The roughness index (ROGI-XD) quantifies the difficulty of a materials discovery task, with higher values indicating greater optimization complexity. Given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$

of molecular feature and objective value pairs, we define ROGI as follows. The labels y_i are normalized such that all pairwise differences in objective values fall within the range $[0, 1]$.

ROGI is computed by applying complete-linkage clustering at varying distance thresholds $t \in [0, 1]$, ensuring that all points within a cluster are separated by at most t. At each threshold, we compute the standard deviation of objective values across clusters. As t increases, this standard deviation decreases monotonically. The area under this curve defines ROGI:

$$\mathrm{ROGI} = \int_0^1 2(\sigma_0 - \sigma_t)dt \tag{A.2}$$

where σ_t denotes the standard deviation at threshold t (Aldeghi et al., 2022; Graff et al., 2023). Intuitively, when similar inputs exhibit large variations in objective values, dispersion across clusters decreases sharply at low thresholds, leading to a higher ROGI. This metric has been used to compare model performance across datasets and feature representations.

However, Graff et al. (2023) showed that ROGI systematically underestimates roughness in high-dimensional feature spaces. As dimensionality increases, normalized pairwise distances become more concentrated near 1, reducing ROGI values even when the underlying structure remains complex. To correct for this, ROGI-XD replaces the pairwise distance with $1 - \log N_{\text{clusters}} / \log N$, where N_{clusters} is the number of clusters at a given step in the dendrogram and N is the dataset size. This adjustment makes ROGI-XD invariant to both dimensionality and dataset size, yielding a more robust measure of roughness across different molecular representations.

Appendix B. Related work

Various surrogate models for BO in materials discovery have been proposed, but GPs remain the *de facto* practitioners' choice (e.g., Zuo et al., 2021; Griffiths et al., 2022; Strieth-Kalthoff et al., 2024b; Schmid et al., 2024). Recently, Bayesian neural networks and even foundation models have also been used for materials discovery (e.g., Kristiadi et al., 2024; Ramos et al., 2023). However, such expressive surrogate models bring increased complexity due to the ever-growing number of moving parts. In the present work, we deliberately pause and attempt to answer which moving parts practitioners and researchers should focus on.

Principled efforts in evaluating BO algorithms have only been made recently. Li et al. (2024b) studied various surrogate models for BO in general. Kristiadi et al. (2023) discussed the deficiencies of NN-based surrogates. However, they did not study materials discovery problems. While Hickman et al. (2023a) provided a chemistry-focused benchmark, they focused on providing the benchmarking library itself. Meanwhile, Liang et al. (2021); Griffiths et al. (2023) focused on evaluating non-BNN surrogates for materials discovery. While BNNs are known to be notoriously sensitive to hyperparameters, discussions about this issue in sequential decision-making tasks are largely absent, with the notable exceptions of Li et al. (2024a) for BO and Riquelme et al. (2018) for bandits. Our work not provides only the performance evaluation of BO surrogates for materials discovery in the age of foundation models, but also actionable recommendations for practitioners.

Appendix C. Extended experimental setup description

C.1. Datasets

We conduct our experiments on well-established benchmarks: six materials discovery tasks from Kristiadi et al. (2024) and two additional drug discovery tasks, AmpC and D4, from Graff et al. (2021). The materials discovery tasks include minimizing the redox potential (Redoxmer) and solvation energy (Solvation) of battery electrolytes (Agarwal et al., 2021), maximizing the fluorescence oscillator strength of lasers (Laser; Strieth-Kalthoff et al., 2024c), maximizing the power conversion efficiency (PCE) of photovoltaic materials (Photovoltaics; Lopez et al., 2016), and maximizing the $\pi - \pi^*$ transition wavelength of organic photoswitches (Photoswitch; Griffiths et al., 2022). For drug discovery, the tasks Kinase, AmpC, and D4 aim to minimize docking scores, a critical metric for evaluating potential binders (Graff et al., 2021).

C.2. Features

We consider three types of features: data-driven, generic hand-crafted, and expert hand-crafted features.

Data-driven features We use 784-dimensional embeddings extracted by average pooling of the final layer of the MolFormer (Ross et al., 2022) and Chemistry-T5 (T5-chem; Christofidellis et al., 2023) models. T5-Chem uses the just-smiles prompting strategy from Kristiadi et al. (2024).

Generic hand-crafted features These features include the 1024-bit Morgan fingerprints with radius 3 (Morgan, 1965) as well as all the molecular descriptors from the Mordred Python package (Moriwaki et al., 2018).

Expert hand-crafted features We use the FM-DFT (foundation model predicted density functional theory) properties, force-field energy, and the maximum degree of conjugation. The FM-DFT features are predicted using the SMI-TED model (Soares et al., 2024b), fine-tuned on the QM9 dataset (Ramakrishnan et al., 2014; Ruddigkeit et al., 2012), which achieves state-of-the-art performance on QM9. The Merck molecular force-field energy (force field), computed using RDKit, is particularly relevant for tasks like Solvation, Kinase, AmpC, and D4. Both FM-DFT and force-field energy are general molecular properties applicable across all datasets. Finally, the maximum degree of conjugation, a simple computation, is expected to be especially relevant for datasets similar to Photovoltaics, Laser, Photoswitch, and Redoxmer.

C.3. Acquisition function

In all experiments, we use the Thompson sampling (Thompson, 1933) acquisition function for its simplicity. Kristiadi et al. (2024) found that Thompson sampling and expected improvement perform similarly on average, making it a reasonable default choice.

C.4. Bayesian optimization initialization

At the beginning of the BO campaign, we standardize the features of all candidates in $\mathcal{X}$ to have zero mean and unit variance, and standardize labels in $\mathcal{D}_t$ before training the surrogate models. We start the BO with an initial dataset of 10 function evaluations drawn uniformly from the set of candidates $\mathcal{X}$.

C.5. Surrogate models

Gaussian process surrogates For GP surrogates, we use the Tanimoto covariance function (Griffiths et al., 2023) when using fingerprint features and otherwise a Matérn-5/2 covariance function with automatic relevance determination (ARD), as it is generally considered a more realistic default choice for BO than RBF (Snoek et al., 2012; Garnett, 2023). When combining fingerprints with other features, we apply Tanimoto to the fingerprint components and Matérn-5/2 to the rest. We include a scale parameter controlling the variance of the prior and constrain the lengthscales to the range $[10^{-3}, 10^3]$ to prevent numerical instabilities when using the aggressive L-BFGS optimizer.

Bayesian neural networks For the BNN, we use a two-layer multi-layer perceptron with 50 hidden units each and tanh activations. We approximate the posterior using the linearized Laplace with the Kronecker-factored posterior covariance approximation (Ritter et al., 2018; Immer et al., 2021). We pose a Gaussian prior on the weights $p(\boldsymbol{w}) = \prod_{l=1}^{L} \mathcal{N}\left(\mathbf{0}_l, \sigma_{p,l}^2 \boldsymbol{I}_l\right)$, with an independent variance parameter $\sigma_{p,l}^2$ per layer. We fit the prior parameters and the likelihood's observation noise by maximizing the marginal likelihood after training the neural network (i.e., the so-called posthoc Laplace approximation). We also run experiments with FSP-Laplace (Cinquin et al., 2024), which approximates the posterior using the Laplace but under a GP prior, using the same neural network architecture, and the same prior as the GP surrogates.

Bayesian LoRA We consider a Bayesian fine-tuned MolFormer network (Ross et al., 2022) applying the Laplace approximation to its LoRA weights, following Kristiadi et al. (2024). We use the same hyperparamters as Kristiadi et al. (2024) for both simple LoRA and Bayesian LoRA fine-tuning.

C.6. Metrics

We measure BO performance by tracking the optimal value over time and by using the GAP score (Jiang et al., 2020) to aggregate over all BO tasks. The GAP score is computed by converting each task to a maximization problem, normalizing the objective values to the range $[0, 1]$, and averaging over all tasks. In each figure, we report the mean and standard error of the score across 5 repetitions with different random seeds.

C.7. Software

We use the BOTorch (Balandat et al., 2020) Python package for the Bayesian optimization experiments and GPyTorch (Gardner et al., 2018) for GP models. When using fingerprints, we use the Tanimoto covariance function from Gauche (Griffiths et al., 2023). PyTorch (Paszke et al., 2019) serves as our framework for neural networks. We retrieve Chemical-T5, MolFormer, and SMI-TED models from Hugging Face using the Transformers library and use the PEFT package for LoRA. We implement Bayesian LoRA fine-tuning using code from https://github.com/wiseodd/lapeft-bayesopt (Kristiadi et al., 2024). For the Laplace approximation, we rely on the Laplace-PyTorch package (Daxberger et al., 2021).

C.8. Hardware

We ran our experiments using Intel Xeon Gold CPUs and Nvidia 2080Ti GPUs with 11GB of memory.

Appendix D. Additional experimental results

D.1. Results for ROGI

The roughness index (ROGI-XD; see detailed description in Appendix A.5) quantifies objective function variability and correlates with test error (Aldeghi et al., 2022). We compute ROGI for each dataset under different feature representations, using Tanimoto distance for fingerprints and Euclidean distance otherwise.

We find that the choice of features significantly influences ROGI, sometimes leading to large variations for the same dataset (see Figure D.1). For example, in Solvation, T5-Chem features yield a ROGI of $\sim$0.06, while the degree of conjugation results in a ROGI of $\sim$0.18.

ROGI-XD generally correlates with average cumulative regret but does not always accurately reflect task difficulty. In our experiments, it exhibits a Pearson correlation of at least 0.5 with average cumulative regret in five out of eight tasks (see Figure D.2). However, it fails to fully capture optimization complexity. For instance, D4, the most difficult task with the highest cumulative regret, has among the lowest ROGI scores (0.080–0.115). In contrast, Photovoltaics, which is both challenging and highly variable, has the highest ROGI. These results suggest that while ROGI captures variability in molecular representations, it does not reliably explain Bayesian optimization (BO) difficulty or sensitivity to hyperparameter choices.

We find that fine-tuning typically reduces ROGI-XD, with notable decreases observed in Photovoltaics, Redoxmer, and Laser (see Figure D.3). However, lower ROGI-XD does not always lead to better BO performance. For example, despite a significant reduction in ROGI for Photovoltaics, BO performance remains unchanged. Conversely, D4 and Solvation exhibit stable ROGI values yet show substantial improvements with fine-tuning. In contrast, Laser and AmpC display a clearer relationship between ROGI reduction and improved BO performance. These findings indicate that while ROGI-XD provides insight into objective variability, it is not a reliable predictor of BO performance across all tasks.

D.2. Additional Bayesian optimization results

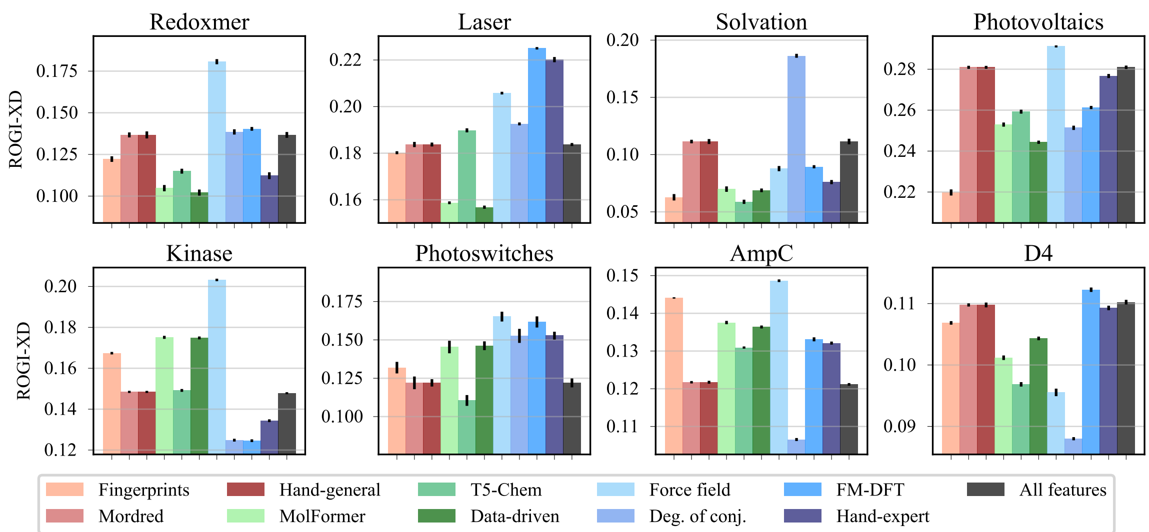

Figure D.1: Different feature representations induce different roughness index (ROGI-XD).

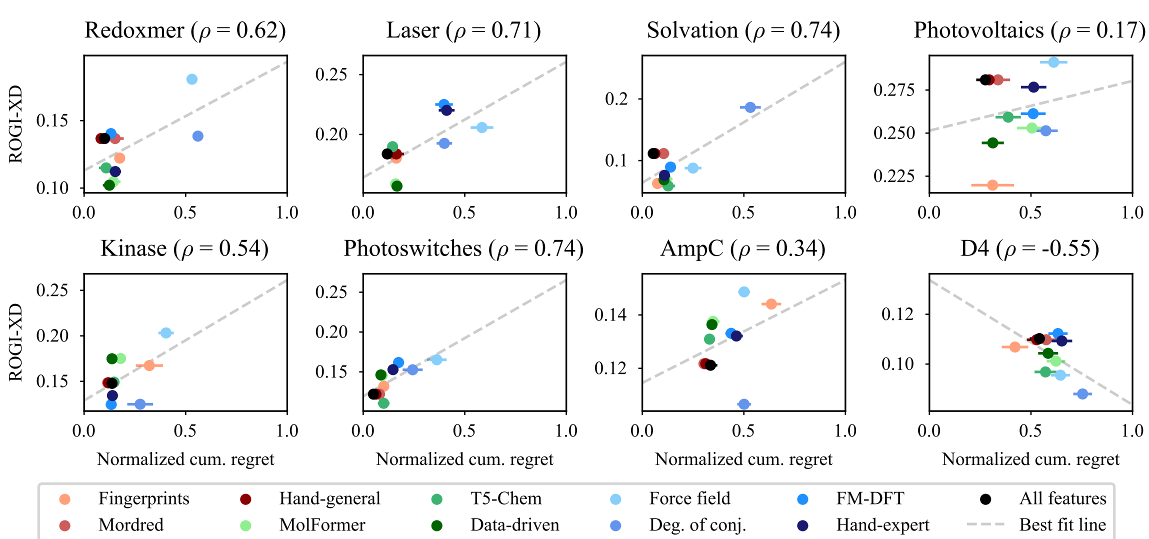

Figure D.2: Roughness index (ROGI-XD) shows a strong positive Pearson correlation ($\rho > 0.5$) with the cumulative regret of the GAP score on 5/8 datasets.

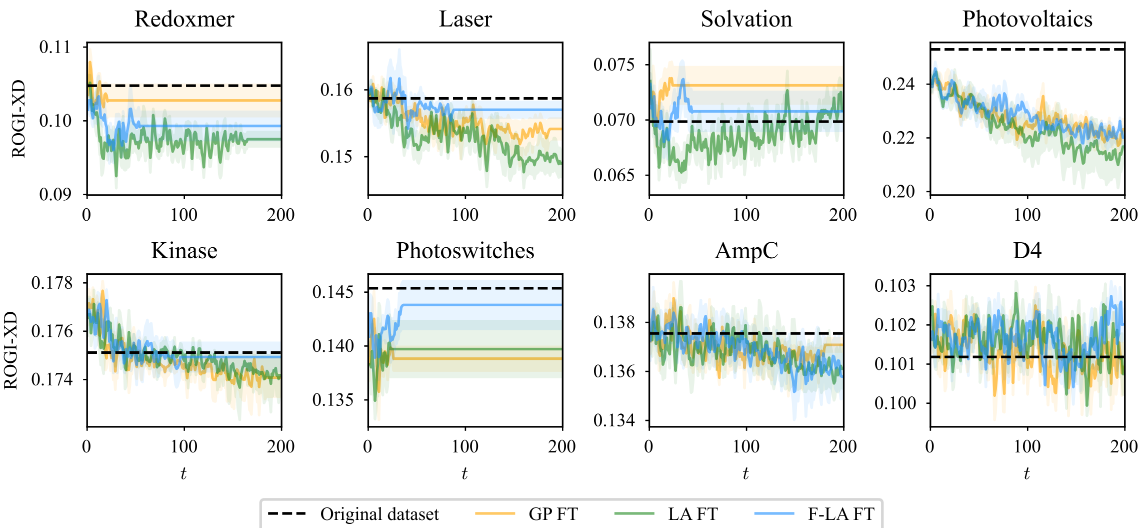

Figure D.3: Rolling average of the roughness index (ROGI) against the BO iteration during feature fine-tuning. Fine-tuning does not systematically decrease the ROGI.

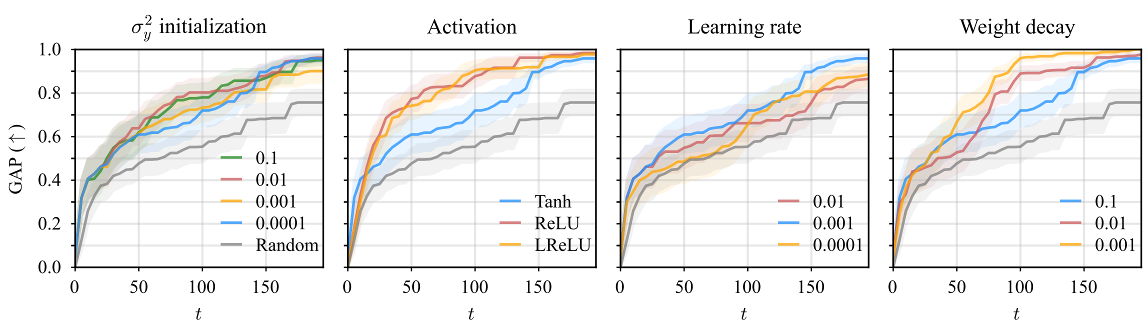

Figure D.4: GAP metric for Bayesian neural network surrogates (Laplace approximation) across different design choices (for MolFormer features). We see that the choice of activation, learning rate and weight decay have a strong influence on BO performance but not the initialization of the observation noise.

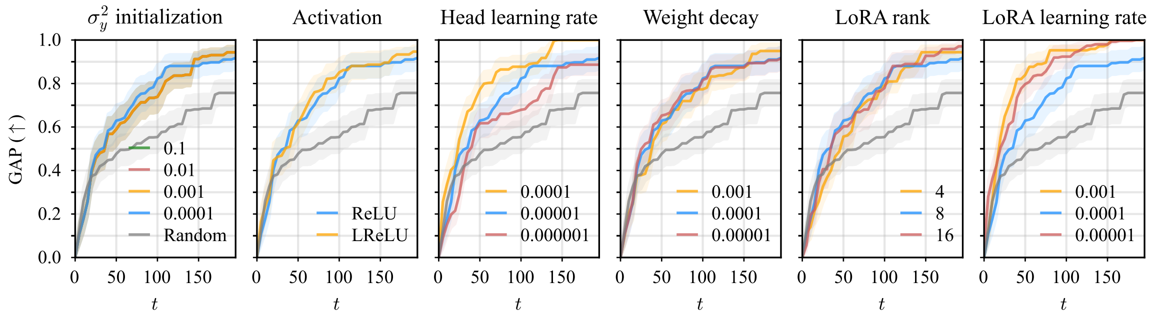

Figure D.5: GAP metric for Laplace Bayesian LoRA surrogates across different design choices (for MolFormer features). The choice of learning rate for the LoRA and regression head strongly influences the average BO performance but not the choice of weight decay, rank of the LoRA factors and activation function.

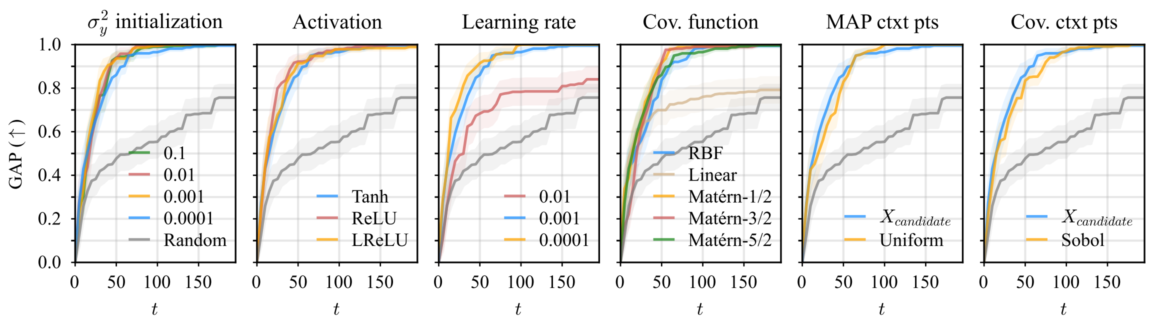

Figure D.6: GAP metric for FSP-Laplace surrogates across different design choices (for MolFormer features). We find BO performance is sensitive to the choice of learning rate and performs poorly for the linear covariance function, which is similarly ineffective for GP surrogates Figure 2, but not for other hyperparameters. Notably, average BO performance with FSP-Laplace is more stable to different choices of hyperparameters than with Laplace surrogates.

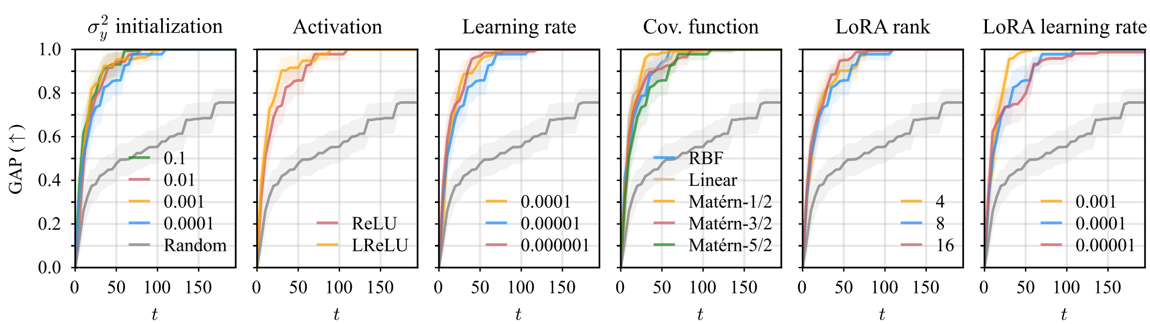

Figure D.7: GAP metric for FSP-Laplace Bayesian LoRA surrogates across different design choices (for MolFormer features). Average BO performance is sensitive to the choice of learning rate for the LoRA and regression head weights, but no longer to the choice of covariance function.

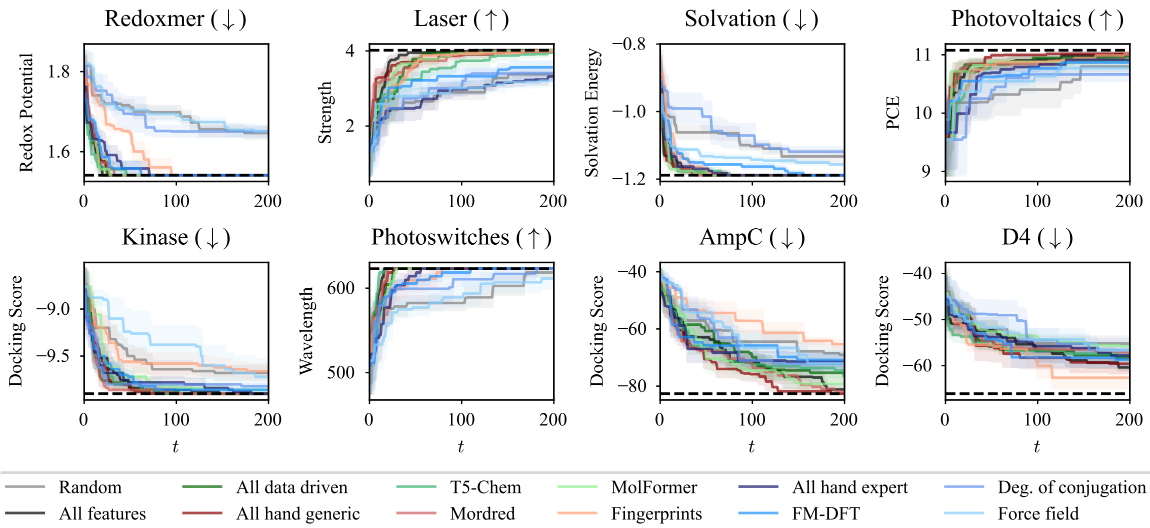

Figure D.8: Mean and standard deviation of the optimal value found during the BO at each time step across different features (for Gaussian process surrogate models). The dark dashed line shows the optimal value for each task.

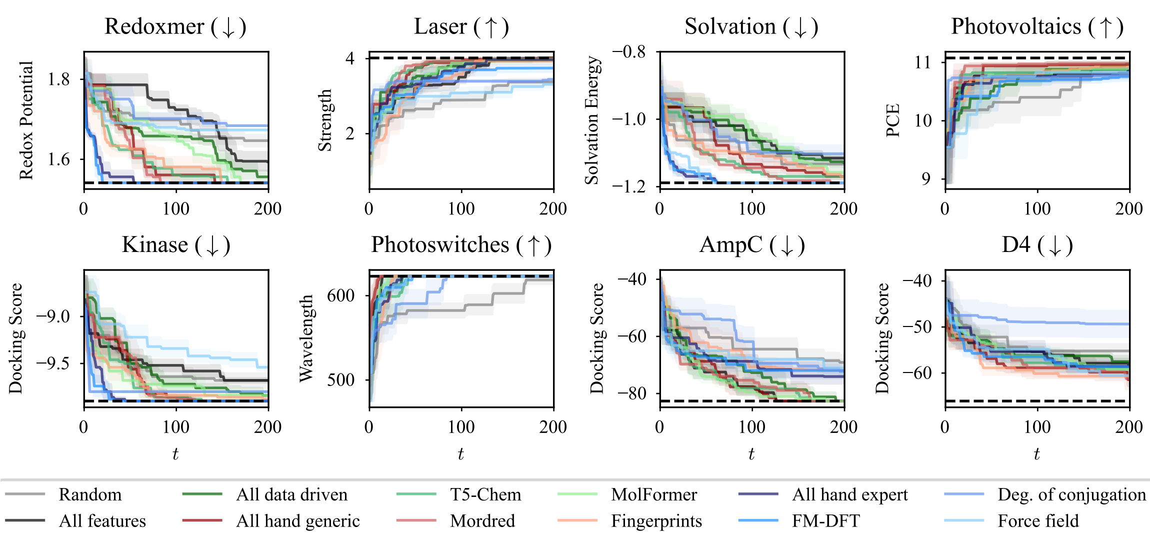

Figure D.9: Mean and standard deviation of the optimal value found during the BO at each time step across different features (for Laplace approximation surrogate models, Bayesian neural network). The dark dashed line shows the optimal value for each task.

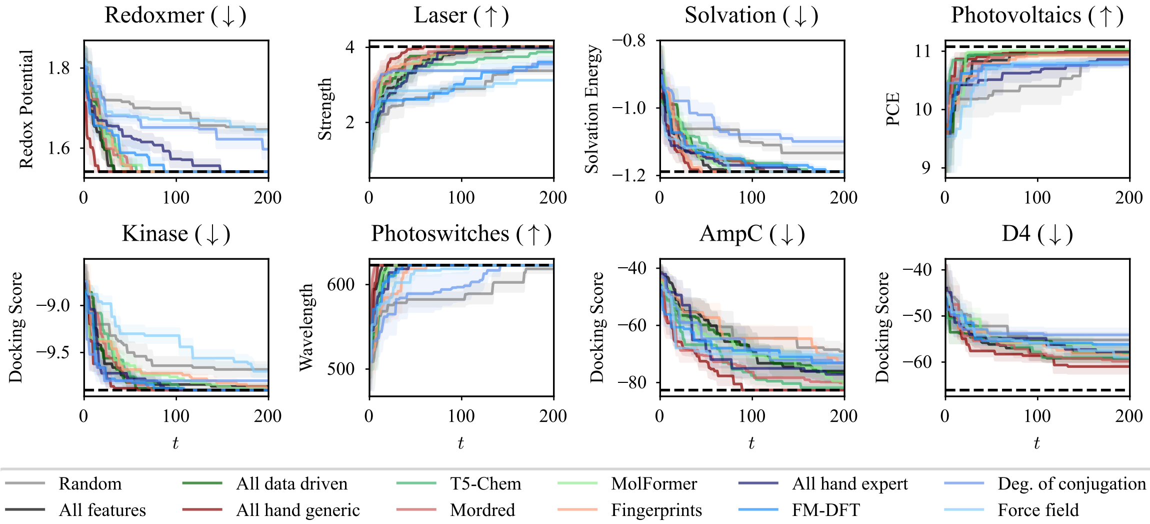

Figure D.10: Mean and standard deviation of the optimal value found during the BO at each time step across different features (for FSP-Laplace surrogate models, i.e., Bayesian neural network with a GP prior). The dark dashed line shows the optimal value for each task.

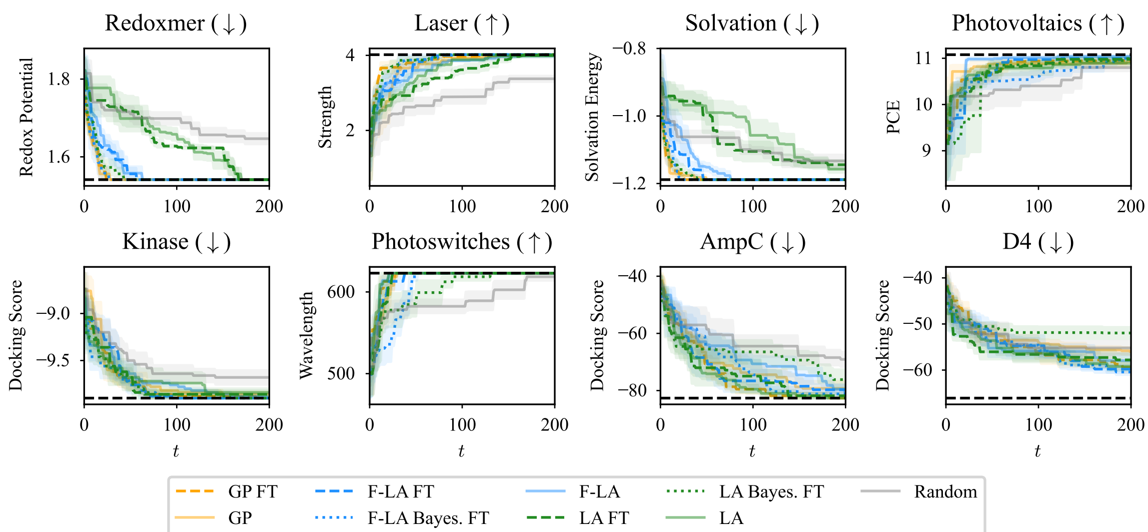

Figure D.11: Mean and standard deviation of the optimal value found during the BO at each time step for GP, Laplace and FSP-Laplace surrogate models and different feature fine-tuning strategies (fixed features, fine-tuned features and Bayesian fine-tuned surrogates). The dark dashed line shows the optimal value for each task.

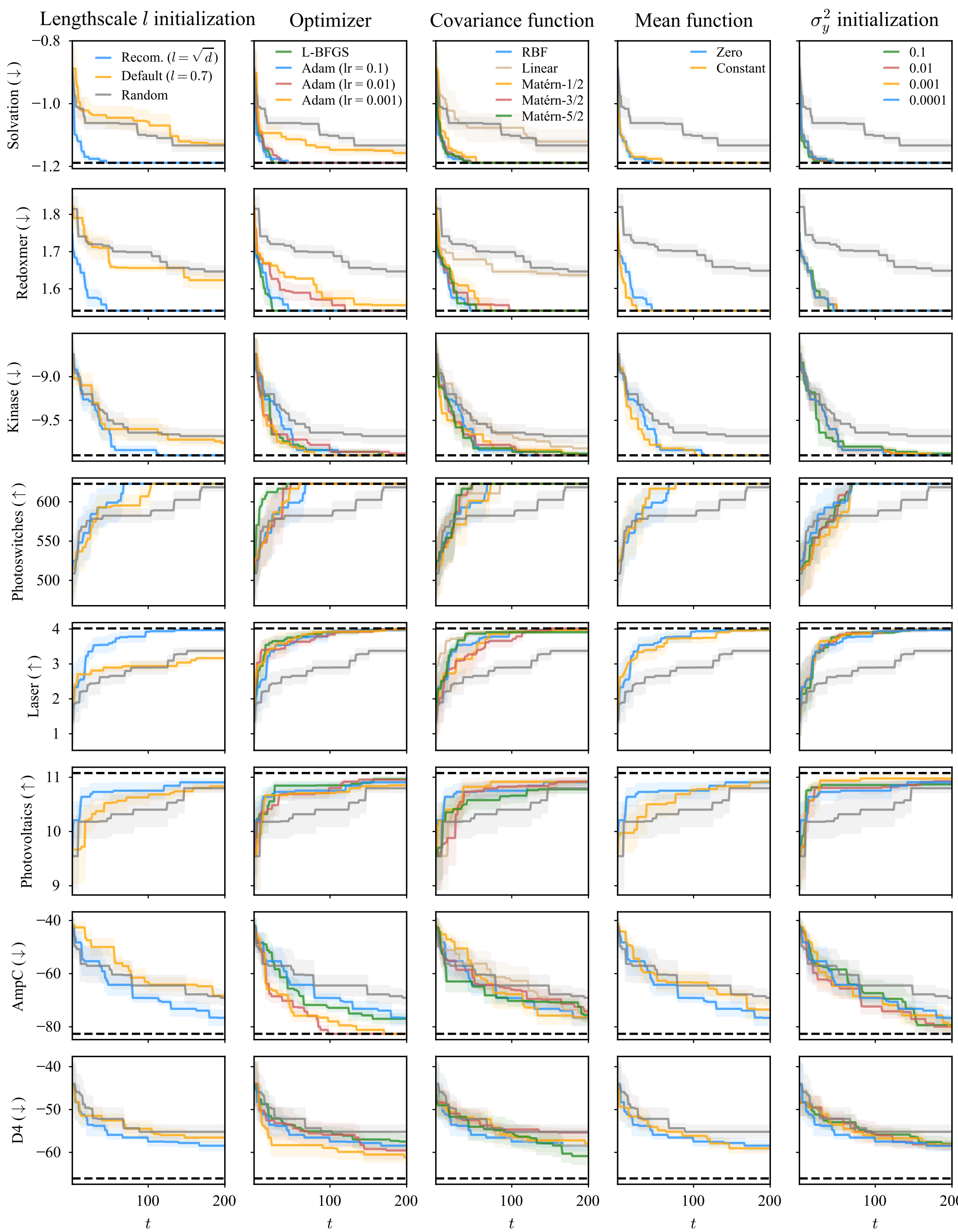

Figure D.12: Mean and standard deviation of the optimal value found during BO at each time step for GP surrogates across different design choices (for MolFormer features). The dark dashed line shows the optimal value for each task.

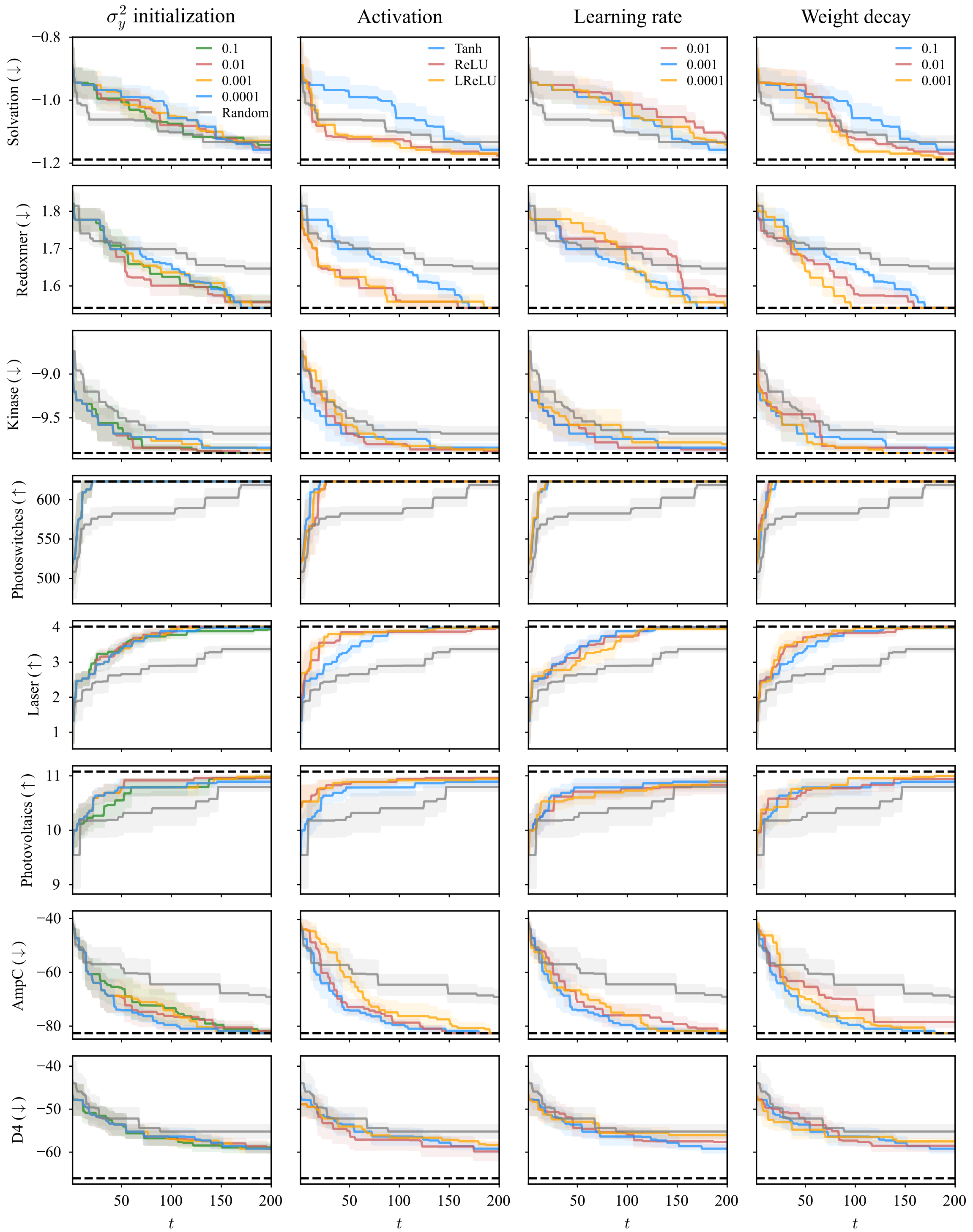

Figure D.13: Mean and standard deviation of the optimal value found during BO at each time step for Laplace surrogates across different design choices (for MolFormer features). The dark dashed line shows the optimal value for each task.

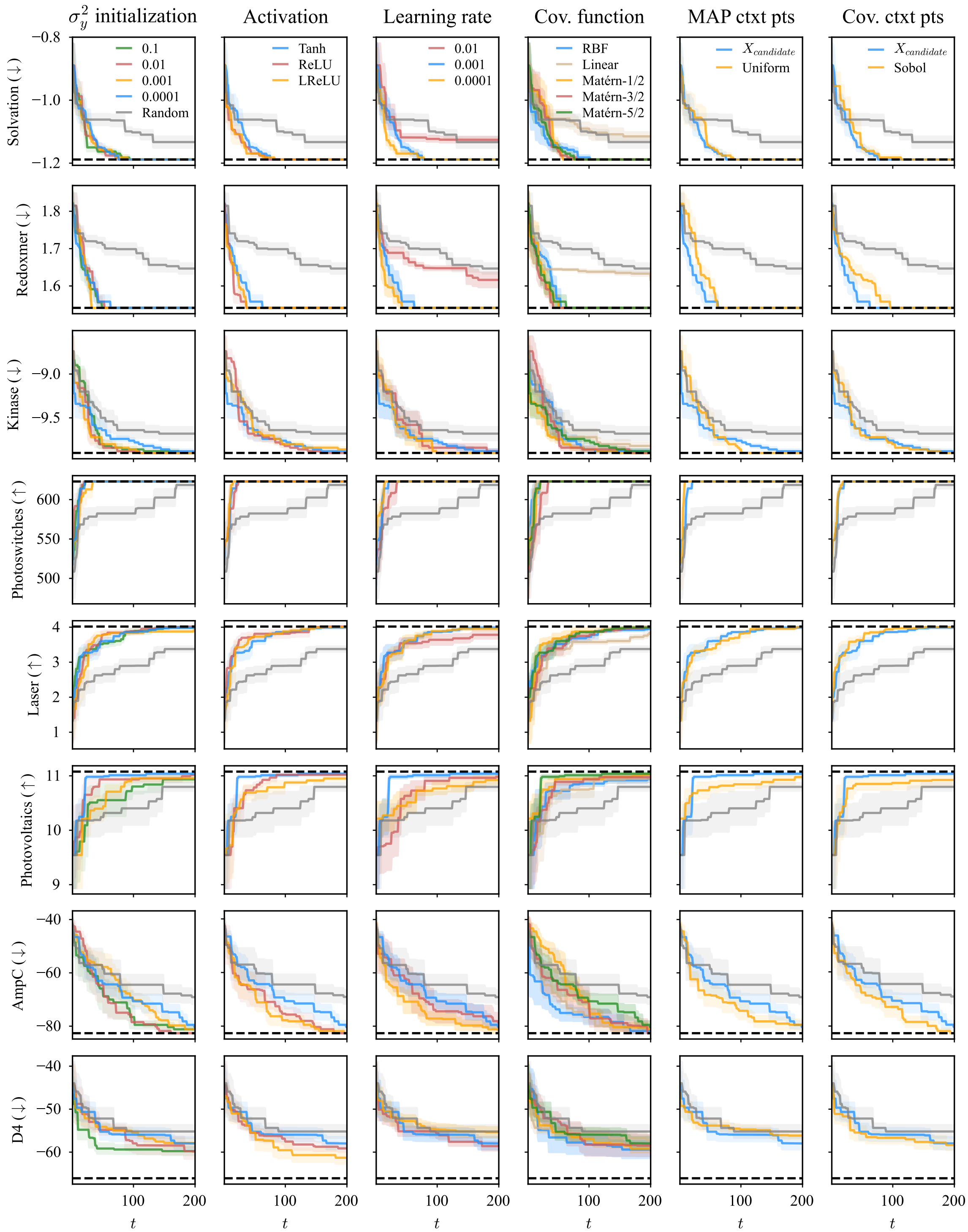

Figure D.14: Mean and standard deviation of the optimal value found during BO at each time step for FSP-Laplace surrogates across different design choices (for MolFormer features). The dark dashed line shows the optimal value for each task.

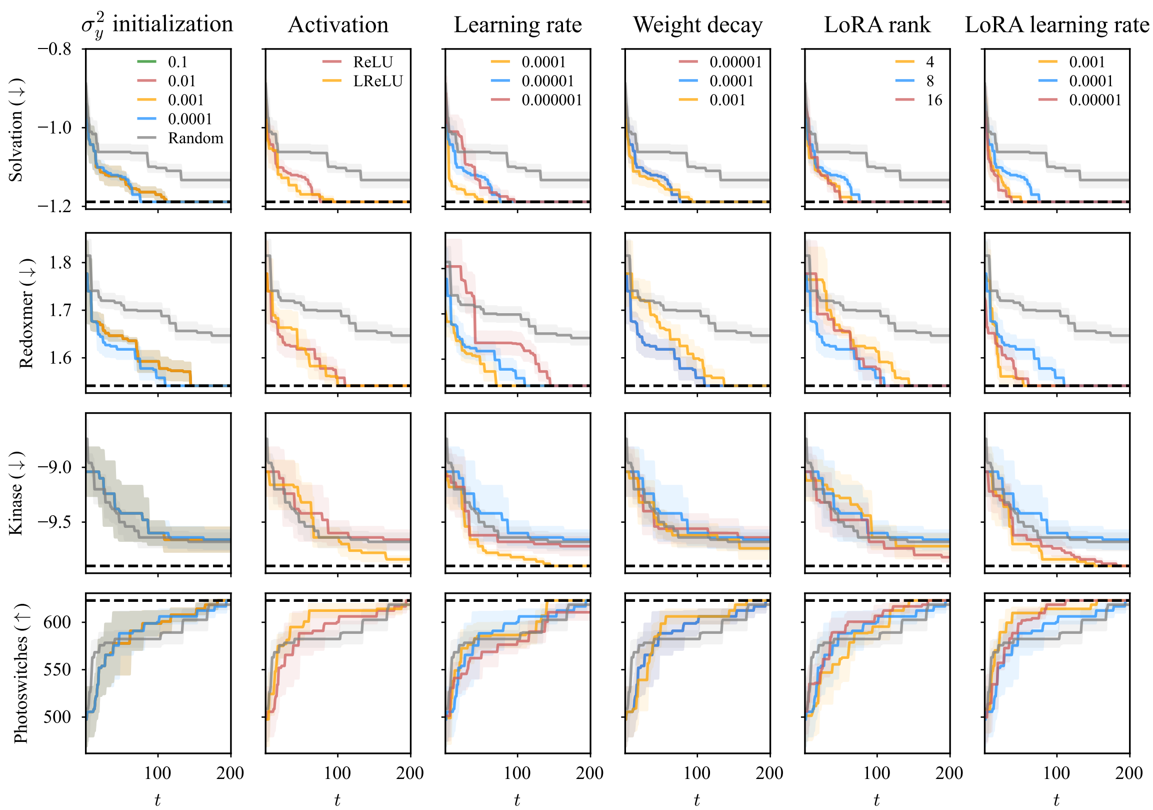

Figure D.15: Mean and standard deviation of the optimal value found during BO at each time step for Laplace Bayesian LoRA surrogates across different design choices (for MolFormer features). The dark dashed line shows the optimal value for each task.

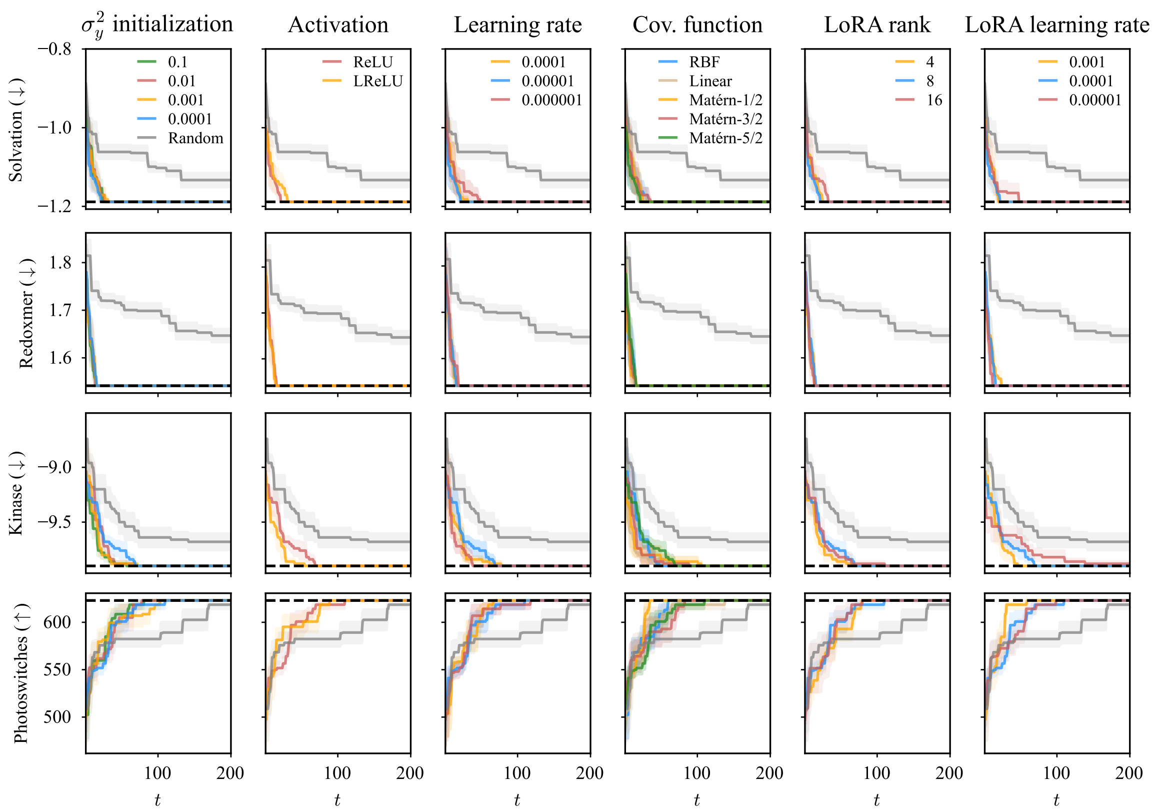

Figure D.16: Mean and standard deviation of the optimal value found during BO at each time step for FSP-Laplace Bayesian LoRA surrogates across different design choices (for MolFormer features). The dark dashed line shows the optimal value for each task.