OPT2025: 17th Annual Workshop on Optimization for Machine Learning

LEONARDBO: Fast and Prior-Driven Bayesian Optimization without Surrogate Modeling

Efe Mert Karagözlü EKARAGOZ@CS.CMU.EDU
Conor Igoe CIGOE@CS.CMU.EDU
Barnabás Póczos BAPOCZOS@CS.CMU.EDU
Jeff Schneider SCHNEIDE@CS.CMU.EDU
Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

In many real-world black-box optimization problems, practitioners know that the maximizer exists in a rather small subset of the search space, yet most common Bayesian Optimization (BO) frameworks do not allow them to input their prior knowledge over the maximizer. In addition, although the goal of BO is only to find the optimizer, BO surrogate models typically model the distribution of the whole latent function, which may introduce a computational burden. Motivated by these, we propose LEONARDBO, a novel approach to BO in which we only update the distribution of the argmax directly given the new observation in the surrogate modeling step, using a neural network to learn to do such updates. This not only enables custom priors over the optimum, but also results in $\mathcal{O}(n)$-time updates in the number of samples, in contrast to exact Gaussian Process (GP) updates with $\mathcal{O}(n^3)$-time. We analyze our method's performance empirically on synthetic functions as well as a real scientific problem where large language models (LLMs) can provide useful priors.

1. Introduction

Bayesian Optimization (BO) enables the optimization of costly-to-evaluate functions via sequential observations. In practice, the latent function f is typically modeled as a Gaussian process (GP) with a specified prior mean and kernel. Practitioners seldom tune the kernel and usually adopt standard choices such as the radial basis function (RBF) or Matérn kernels. This practice induces a prior–mismatch problem: the expert's belief about the maximizer (argmax) of f, denoted $\mathbf{x}^* = \arg\max_{\mathbf{x}} f(\mathbf{x})$, often conflicts with the GP prior. (Here, we assume a unique maximizer.) Because optimization ultimately targets $\mathbf{x}^*$, a mismatch in its distribution affects performance directly. Indeed, RBF kernels put substantial prior mass in the corners of the search space, resulting in boundary over-exploration. [21].

We propose to train a neural network that maps a prior over $\mathbf{x}^*$ and a new observation to the corresponding posterior. The network can replicate GP updates for a standard kernel while starting from an arbitrary prior over the argmax. Defining this prior is generally more transparent than engineering a bespoke kernel and mitigates the mismatch while retaining desirable GP properties such as smoothness. We call our method **LEONARDBO—Le**arning Updates **on Ar**gmax **D**istributions for **B**ayesian **O**ptimization.

To illustrate what it means to update the x^* distribution directly, consider optimizing the yield of a chemical reaction by adjusting the ratio of two inputs. A chemist may initially believe that the optimum lies somewhere in the middle of the range, since extreme ratios are known to perform

poorly; this belief can be represented directly as a prior distribution over the optimizer location x^*. As experiments are run, this distribution is updated: for instance, if an evaluation at ratio 0.2 yields much lower output than one at 0.6, the posterior shifts mass away from 0.2 and toward more promising regions. This example highlights what it means to place and update a prior on *where the maximizer is*, without modeling the full function.

Acquisition functions that depend solely on the posterior of $\mathbf{x}^*$ still remain computable. The obvious example is Thompson Sampling. [26] With slight modifications, Predictive Entropy Search [6], ϵ-Greedy [9], and Top-Two algorithms [20] also remain accessible.

Contributions. We (1) review relevant literature in Appendix A; (2) quantify the prevalence and severity of prior mismatch in real-world tasks in Appendix B; (3) introduce LEONARDBO, which replaces GP updates with neurally learned updates on argmax distributions, and present a neural architecture tailored to discrete domains; (4) evaluate LEONARDBO on synthetic benchmarks; and (5) show the use of large language models (LLMs) as informative priors in a scientific BO task.

2. Problem Formulation

A Gaussian process (GP) with kernel $k(\cdot,\cdot)$ and prior mean $\mu(\cdot)$ receives data as pairs $\{(\mathbf{x}_t, y_t)\}_{t=1}^{T}$, where $\mathbf{x}_t \in [0,1]^d$, $y_t \in \mathbb{R}$, and T is fixed by a sampling budget or stopping criterion. After receiving each observation, the GP computes a distribution $\mathcal{P}_t$ over the possible underlying functions f. $\mathcal{P}_t$ gives the entire state of the GP at time t. Note that the whole state of the GP can also be represented by its observational data at time t, $\mathcal{D}_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^{t}$.

The posterior $\mathcal{P}_t$ induces a distribution p_t over the maximizer $\mathbf{x}^* = \arg\max_{\mathbf{x}} f(\mathbf{x})$, since $\mathbf{x}^*$ is a deterministic function of the random function f. Here, we assume that the maximum is unique, hence the result of the argmax operation is not a set. Note that we can approximate the distribution p_t by drawing sample functions from $\mathcal{P}_t$ and finding their maximizers, which is essentially Thompson sampling. [26]

Because BO ultimately seeks the maximizer $\mathbf{x}^*$, downstream decisions depend on the argmax distribution $p_t(\mathbf{x}^*)$ rather than on the entire GP posterior. (We note, however, that access to the full state may still be required to *compute* future argmax distributions; our aim is to approximate this update using a reduced representation.) If one could update this distribution using only (i) the current prior $\pi_{t+1}(\mathbf{x}^*) := p_t(\mathbf{x}^*)$, (ii) the new observation $(\mathbf{x}_t, y_t)$, and (iii) a compact summary $C_t \subset \mathcal{D}_t$ of bounded size, the computational cost would reduce to $\mathcal{O}(1)$ per step (and $\mathcal{O}(n)$ overall) while the memory footprint would remain constant.

The role of C_t is to retain salient historical information that the pair $(\pi_{t+1}, (\mathbf{x}_t, y_t))$ alone may not capture—thereby preventing the update rule from "forgetting" context that is predictive of $\mathbf{x}^*$.

Computability of the argmax update. A natural question is whether a mapping

$$\mathcal{M} : \Delta([0,1]^d) \times [0,1]^d \times \mathbb{R} \times \mathbb{R}^{|C_t|} \to \Delta([0,1]^d)$$

exists such that $\mathcal{M}[\pi_t(\mathbf{x}^*), \mathbf{x}_t, y_t, C_t] = p_t(\mathbf{x}^*)$. The prior over $\mathbf{x}^*$ contains less information than the full function posterior, so exact computation may be impossible; however, $\mathcal{M}$ can be learned or approximated statistically. Hence, we parameterize $\mathcal{M}$ with a neural network and learn it from data.

We draw N sample functions from the reference GP, run BO for T steps using acquisition function $\alpha(\cdot)$, and at each step approximate $\pi(\mathbf{x}^*)$ and $p(\mathbf{x}^*)$ with M Monte-Carlo samples. This yields NT input–output pairs for $\mathcal{M}$. We minimise the expected Kullback–Leibler divergence

$$\ell = \mathbb{E}[KL(\hat{p}_t(\mathbf{x}^*;\theta)||p_t(\mathbf{x}^*))],$$

where $\hat{p}_t(\mathbf{x}^*;\theta) = \mathcal{M}[\pi_t(\mathbf{x}^*), \mathbf{x}_t, y_t, C_t]$, and the expectation is taken with respect to the GP states encountered by following the acquisition $\alpha(\cdot)$. Note that $\mu(\cdot)$ and $k(\cdot,\cdot)$ specifies the GP.

Existence of a likelihood function. One might ask whether a likelihood function $\mathcal{L}_t(\mathbf{x}^*)$ exists such that $p_t(\mathbf{x}^*) = \pi_t(\mathbf{x}^*)\,\mathcal{L}_t(\mathbf{x}^*)$. However, if we try to estimate $\mathcal{L}_t(\mathbf{x}^*)$ by dividing the posterior by the prior, the estimate also depends on the order of observations, implying that no likelihood function in this simple multiplicative form exists. For a more detailed discussion, see Appendix C.

However, across random BO realizations, an *average* likelihood

$$\hat{\mathcal{L}}(\mathbf{x}^*)[\mathbf{x}, y] \;=\; \mathbb{E}[p_t(\mathbf{x}^*)/\pi_t(\mathbf{x}^*)]$$

may still be well defined. Learning this quantity regularizes the updates of network by discouraging highly variable updates for identical observations.

Equivariance properties. If the domain were infinite, the average likelihood would satisfy

- *Positional equivariance:* $\hat{\mathcal{L}}(\mathbf{x}^*)[\mathbf{x}, y] = \hat{\mathcal{L}}(\mathbf{x}^* - \mathbf{x})[\mathbf{0}, y]$;
- *Rotation invariance:* $\hat{\mathcal{L}}(\mathbf{x}_1^*)[\mathbf{x}, y] = \hat{\mathcal{L}}(\mathbf{x}_2^*)[\mathbf{x}, y]$ whenever $\|\mathbf{x}_1^* - \mathbf{x}\| = \|\mathbf{x}_2^* - \mathbf{x}\|$,

because the underlying kernel is stationary and depends only on distances between points. Although the practical domain is finite, we retain these symmetry assumptions in our model design.

3. The Neural Network

The preceding discussion indicates that the prior $\pi(\mathbf{x}^*)$ contains information about $p(\mathbf{x}^*)$ that cannot be expressed solely through a rotation- and shift-invariant likelihood $\mathcal{L}(\mathbf{x}^*)$. Nevertheless, such a likelihood may capture most of the update, with $\pi(\mathbf{x}^*)$ providing a modest correction.

Guided by this intuition, we train a network to predict an *argmax-likelihood* term $\mathcal{L}(\mathbf{x}^*)$ from the inputs $(\mathbf{x}, y)$ and recover the posterior by multiplication. Working in log space,

$$\log \hat{p}(\mathbf{x}^*) \;=\; \log \pi(\mathbf{x}^*) \;+\; \mathrm{NN}(\mathbf{x}, y),$$

where NN is required to be shift-invariant and permutation-symmetric in $\mathbf{x}$. This is enforced by modelling $\mathcal{L}(\mathbf{x}^*)[\mathbf{x}, y] = \rho(\|\mathbf{x}^* - \mathbf{x}\|)\,[y]$, and learning the radial function ρ with a small network.

To allow limited, non-symmetric adjustments, we introduce a latent representation z that depends on both the prior and the observation history:

$$\begin{aligned}
\log \hat{p}(\mathbf{x}^*) &= \text{Direct-}\hat{p}\text{-Decoder}(z;\theta_1) \\
&\quad + \left(\log \pi(\mathbf{x}^*) + \rho\text{-to-}\mathcal{L}\text{-Transform}(\rho\text{-Decoder}(z,\theta_2), \mathbf{x}_t)\right), \\
z &= \text{Direct-}\pi\text{-Encoder}(\log \pi(\mathbf{x}^*), \theta_3) + y\text{-Encoder}(y_t, \theta_4) \\
&\quad + y\text{-Encoder}(y_t - y_{t-1}^*; \theta_5)
\end{aligned}$$

y_{t-1}^*, the previous best observation is the context C_t that we use here. The latent encoders and decoders employ convolutional layers to preserve local structure. Parameters θ_1 and θ_3 are initialized at zero, limiting the direct influence of the prior and bounding posterior asymmetry at the start of training. For details of the architecture and training, check Appendix D.

4. Experiments

We begin by describing the training procedure. We generated transition data from an exact GP following Thompson Sampling (TS) on a discrete grid. We considered domains of up to 3 dimensions, each being the unit hypercube. Each axis was discretized into 11 grid points ranging from 0 to 1 in increments of 0.1, resulting in 11, 121, and 1331 total grid points depending on dimensionality.

After each GP update in the BO loop, we generated $M = 500{,}000$ Thompson samples to approximate the argmax distribution and saved it. This was done for $N = 2000$ parallel episodes and $T = 30$ timesteps. The GP used zero prior mean, an RBF kernel with lengthscale 0.2, noise variance 0.01, and outputscale 1.0. We trained networks of appropriate size to match task difficulty. (76k parameters for 1D, 438k for 2D, and 5.9M for 3D.) Code for the experiments and training of models can be found online.[1]

4.1. Synthetic Experiments

Here, we again considered functions on grids of 11 points per dimension, sampled from GPs with hyperparameters given exactly in the previous section. We had $N = 1000$ runs for $T = 30$ timesteps, each run having an independent function drawn from the GP. Then, we replaced the GP with our neural networks to predict the posterior over the argmax and ran TS using the predicted (approximate) posterior. The resulting instantaneous and simple regret curves are in Figure 1. The networks match GP performance reasonably and even outperform it early in simple regret.

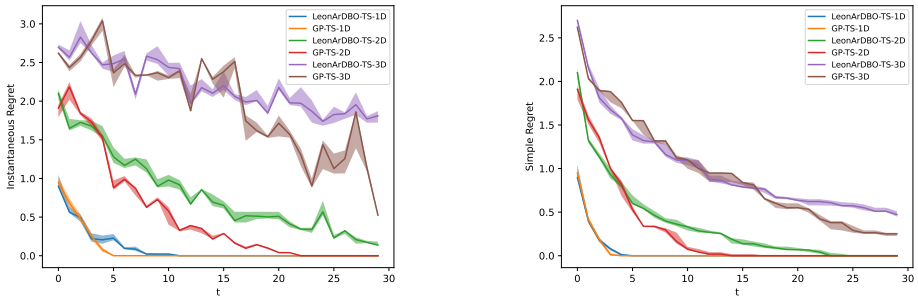

Figure 1: Instantaneous and simple regret for three dimensions using GP and LEONARDBO under Thompson Sampling. For all figures, solid lines show the median; error bars indicate 95% bootstrap confidence intervals. GP results use 2000 runs; LEONARDBO uses 1000.

Runtime Analysis. All runtime experiments were performed on Intel® Xeon® Gold 6152 CPUs (2.10 GHz). The LEONARDBO–TS–3D model requires on average 0.0195s to update the distribution, C_t, and draw a TS. For variational GP methods, we optimized the ELBO and selected inducing points following (1) greedy variance reduction [2] and (2) greedy improvement reduction (GIR) [15], as recommended for global accuracy and optimization, respectively. Two standard strategies were used to determine the number of inducing points: (1) warm-start with 10 exact GP steps and use 10 inducing points; (2) warm-start with 5 steps and use half of the data as inducing

1. https://github.com/EMKaragozlu/LeonArDBO

points (ScaleInd). Figure 2 shows comparable regret across methods, with LEONARDBO being orders of magnitude faster. Note, however, that all experiments were run on CPU, which may inflate variational GP runtimes.

More experimental results can be found Appendix E and F.

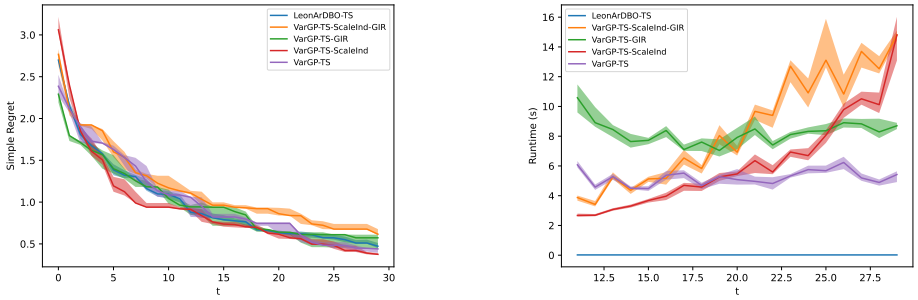

Figure 2: Regret (left) and runtime (right) comparisons between LEONARDBO-TS and variational baselines. 1000 parallel episodes were used. The figure excludes the warm-start phase.

4.2. LEONARDBO for LLM-Enhanced BO

Here, we consider the Buckminsterfullerene adducts experiment from the Olympus package. [7] The aim is to maximize the mole fraction of a desired chemical product by controlling three process conditions (temperature, reaction time and ratio of sultine to C60). We discretize the space and employ our 3D model and the GP it was trained to copy.

To feed LEONARDBO, we asked gpt-4o-mini with temperature 0.7 for five possible optimizer locations and the probability that the true optimizer is in the vicinity of one of these, and repeated this for five samples. We then constructed a Gaussian mixture prior. (We accounted for the probability that it is not in any vicinity, by adding a uniform to the mixture.) To evaluate, we draw $N = 1000$ samples from the built-in Bayesian neural network emulator from Olympus trained on actual data from [11] to simulate the results. The results in Figure 3 indicate that LLMs can provide an informative prior that benefits algorithms capable of exploiting argmax priors, especially early on; conversely, eliciting a full functional prior from an LLM is considerably more challenging. The standard GP baseline lacks a mechanism to incorporate such information and therefore lags behind.

5. Conclusion

This work introduced and tested LEONARDBO, a Bayesian optimization framework that updates only the distribution of the maximizer and skips the usual surrogate modeling. For future work, a continuous version of LEONARDBO could be developed using optimal transport, flow-based, or diffusion-based update mechanisms.

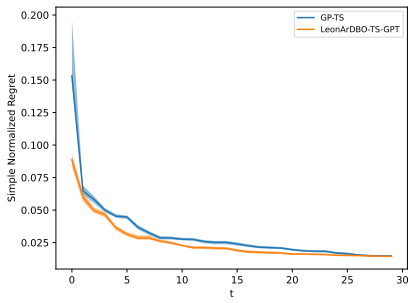

Figure 3: Simple normalized regret of the GPT-informed LEONARDBO and vanilla GP with TS.

Acknowledgments

This work was supported in part by the U.S. Army Futures Command under Contract No. W519TC-23-C-0030.

References

[1] Thang D. Bui, José Miguel Hernández-Lobato, and Richard E. Turner. Streaming sparse gaussian process approximations. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 3299–3307, 2017.

[2] David R Burt, Carl Edward Rasmussen, and Mark Van Der Wilk. Convergence of sparse variational inference in gaussian processes regression. *Journal of Machine Learning Research*, 21(131):1–63, 2020.

[3] Paul E Chang, Nasrulloh Loka, Daolang Huang, Ulpu Remes, Samuel Kaski, and Luigi Acerbi. Amortized probabilistic conditioning for optimization, simulation and inference. *arXiv preprint arXiv:2410.15320*, 2024.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[5] James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 282–290, 2013.

[6] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, 27, 2014.

[7] 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, May 2023.

[8] Carl Hvarfner, Frank Hutter, and Luigi Nardi. Prior-weighted acquisition functions for faster bayesian optimization (π-bo). Technical Report 2204.00032, arXiv, 2022. arXiv:2204.00032 [cs.LG].

[9] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[10] Agustinus Kristiadi, Felix Strieth-Kalthoff, Marta Skreta, Pascal Poupart, Alán Aspuru-Guzik, and Geoff Pleiss. A sober look at llms for material discovery: Are they actually good for bayesian optimization over molecules? *arXiv preprint arXiv:2402.05015*, 2024.

[11] Stefan Langner, Florian Häse, José Darío Perea, Tobias Stubhan, Jens Hauch, Loïc M Roch, Thomas Heumueller, Alán Aspuru-Guzik, and Christoph J Brabec. Beyond ternary opv: high-throughput experimentation and self-driving laboratories optimize multicomponent systems. *Advanced Materials*, 32(14):1907801, 2020.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[13] 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.

[14] Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance bayesian optimization. *arXiv preprint arXiv:2402.03921*, 2024.

[15] Henry B Moss, Sebastian W Ober, and Victor Picheny. Inducing point allocation for sparse gaussian processes in high-throughput bayesian optimisation. In *International Conference on Artificial Intelligence and Statistics*, pages 5213–5230. PMLR, 2023.

[16] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. In *Journal of Machine Learning Research Workshop and Conference Proceedings*, volume 1, pages 114–123, 2005.

[17] 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.

[18] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[19] Simone Rossi, Markus Heinonen, Edwin Bonilla, Zheyang Shen, and Maurizio Filippone. Sparse gaussian processes revisited: Bayesian approaches to inducing-variable approximations. In *International Conference on Artificial Intelligence and Statistics*, pages 1837–1845. PMLR, 2021.

[20] Daniel Russo. Simple bayesian algorithms for best arm identification. In *Conference on learning theory*, pages 1417–1418. PMLR, 2016.

[21] Eero Siivola, Aki Vehtari, Jarno Vanhatalo, Javier Gonzalez, and Michael Riis Andersen. Correcting boundary over-exploration deficiencies in bayesian optimization with virtual derivative sign observations. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2018. doi: 10.1109/MLSP.2018.8516936.

[22] Mikko Siivola, Arno Solin, and Samuel Kaski. Correcting boundary bias in bayesian optimization with virtual derivative observations. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 4701–4710. PMLR, 2018.

[23] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18 (NeurIPS)*, pages 1257–1264. MIT Press, 2006.

[24] Jasper Snoek, Kevin Swersky, Rich Zemel, and Ryan Adams. Input warping for bayesian optimization of non-stationary functions. In *International conference on machine learning*, pages 1674–1682. PMLR, 2014.

[25] Artur Souza, Luigi Nardi, Leonardo B Oliveira, Kunle Olukotun, Marius Lindauer, and Frank Hutter. Bayesian optimization with a prior for the optimum. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*, pages 265–296. Springer, 2021.

[26] 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.

[27] Michalis K. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 567–574, Clearwater Beach, FL, USA, 2009. PMLR.

[28] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

[29] Christopher Williams and Carl Rasmussen. Gaussian processes for regression. In D. Touretzky, M.C. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995. URL https://proceedings.neurips.cc/paper_files/paper/1995/file/7cce53cf90577442771720a370c3c723-Paper.pdf.

Appendix A. Related Work

GPs, introduced by Williams and Rasmussen [29], are the standard surrogate in BO. We highlight four relevant directions.

Scalable GP inference. Exact GP updates scale cubically, limiting BO with many evaluations. Sparse and variational approximations use *inducing points*, starting from pseudo-point methods [16, 23] and later unified by Titsias [27]. Stochastic variational inference extends this to large data [5], while online [1] and Bayesian selection [19] adapt inducing sets dynamically. These achieve near-linear complexity with high accuracy. Random embeddings even allow real-time BO in billion-dimensional spaces [28].

Boundary over-exploration. Stationary kernels with constant means bias acquisition toward corners of bounded domains. Remedies include virtual derivative sign observations at boundaries [21], decaying mean/kernels, and acquisition penalties, all redirecting evaluations inward and improving efficiency.

Argmax priors in BO. Rather than default RBF kernels and constant means, practitioners often know likely optima locations. Approaches include input warping [24], priors over the optimum itself [25], and π-BO [8], which reweights acquisitions by a prior density while preserving convergence. These reduce sample complexity but not computational cost, as they still maintain full posteriors.

The work most closely related to ours is Chang et al. [3], which proposes a transformer-based meta-learning approach that learns predictive distributions for both data and latent variables, while also being able to encode priors over latents. One of their applications is BO, where the argmax location and maximum value are treated as latent variables. Their method additionally offers runtime improvements through amortization. However, (1) their network requires all observed data as input, even though this set already fully characterizes the surrogate model state, (2) their transformer architecture incurs a runtime of $\mathcal{O}(n^2)$, and (3) in sequential BO, the latents must be re-inferred as the dataset grows, further increasing computational cost. In contrast, with LEONARDBO we show that (1) retaining only the previous argmax distribution, rather than the full observation set, is sufficient to achieve low regret, (2) for a static dataset of size n, our architecture achieves $\mathcal{O}(n)$ total runtime by applying $\mathcal{O}(1)$ sequential updates, and (3) as a result, the cumulative runtime across a BO task is $\mathcal{O}(n)$, since previously inferred posteriors can be directly reused.

LLMs for BO. Pre-trained LLMs offer rich priors. Ramos et al. [17] use token probabilities and in-context learning to guide molecule optimization. Liu et al. [14] apply LLMs for warm starts, surrogates, and candidate generation. However, in-context updates are often unreliable [10], and true Bayesian integration may require costly retraining.

Appendix B. Prior Mismatch in the Wild

As noted by Siivola et al. [22], stationary kernels such as RBF combined with constant mean functions assign disproportionate prior mass to corner solutions. However, few studies quantify the extent of this mismatch in real-world tasks. We examine the phenomenon on five molecular optimisation problems from Liang et al. [13]: electrical conductivity, absorbance spectrum score, stability score, mechanical toughness, and shape score. The corresponding datasets contain 3–5 input dimensions and 100–600 candidate molecules (see Table 1 in [13]).

To characterise the severity of the bias, we formalise the notion of corners. A design point $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ is a corner in dimension i if $\mathbf{x}_i \in (\arg\max_{\mathbf{x}' \in \mathcal{X}} \mathbf{x}'_i) \cup (\arg\min_{\mathbf{x}' \in \mathcal{X}} \mathbf{x}'_i)$. If $\mathbf{x}$ is

Table 1: Empirical Investigation of the Prior Mismatch on the Corners

Dataset	$\pi(\mathbf{x}^* \in S_1)$	$\pi(\mathbf{x}^* \in S_2)$	$\pi(\mathbf{x}^* \in S_3)$	$\pi(\mathbf{x}^* \in S_4)$	$\pi(\mathbf{x}^* \in S_5)$	$C(\mathbf{x}^*)$
Crossed Barrel	99.7%	95.5%	76.6%	36.1%	0	2
P3HT	73.9%	69.8%	69.8%	50.3 %	13.4%	0
AutoAM	99.6%	90.1 %	76.3 %	35.0 %	0	1
Perovskite	93.4%	56.5%	56.5%	0	0	1
AgNP	23.4%	9.2%	0	0	0	0

a corner in j dimensions, we call it a j-corner; let S_j denote the set of all j-corners and define $C(\mathbf{x}) = \max\{j : \mathbf{x} \in S_j\}$.

Each input space is scaled to $[0, 1]^d$ and outputs are standardised. We initialise GPs with zero mean and an RBF kernel, adopting the prior hyperparameters of Liang et al. [13]: unit length- and output-scale and noise variance 0.01. Whereas Liang et al. [13] updated these hyperparameters online, we retain their prior values. For each task we estimate, from 10^7 GP samples, the probability π that the true optimiser $\mathbf{x}^*$ is a j-corner, and denote this by $\pi(\mathbf{x}^* \in S_j)$. Table 1 shows that GPs markedly overestimate the likelihood of high-order corners.

To summarise this overconfidence, we define

$$\mathbb{E}_{\tau \sim \text{real-world BO tasks}}\left[\pi\left(\mathbf{x}_\tau^* \in S_{C(\mathbf{x}_\tau^*)+1}\right)\right],$$

where $\mathbf{x}_\tau^*$ denotes the maximiser for task τ. A calibrated surrogate should yield a value below 0.5, as over- and underestimates balance. Across the five tasks this quantity reaches 64.1%, indicating pronounced overconfidence in boundary solutions.

Although our analysis is limited to chemically related tasks, we expect similar mismatches to arise in other domains. Practitioners often design search spaces assuming interior optima; if boundaries were suspected, the domain would typically be expanded at the outset.

Appendix C. On the Existence of Multiplicative Likelihood

Figure 4 demonstrates that the update depends on the order of observations, implying that no likelihood function in this simple multiplicative form exists.

Appendix D. Training and Architectural Details

As mentioned in the paper, the network consists of a Direct-$\hat{p}$-Decoder, a ρ-Decoder, a Direct-π-Encoder, and two y-Encoders.

For our 3D model, the y-Encoders are MLPs with hidden layer sizes $[16, 24, 80, 160, 280]$, output size 400, and skip connections (inspired by [4]) from the input to the third hidden layer, and from the third hidden layer to the output. The Direct-π-Encoder is a convolutional neural network ([12]) with 4 repetitions of convolutional layer, batchnorm, ReLU, and dropout (with 0.5 probability) sequences. Also, there is a residual connection from start to the second layer and from second layer to the final layer. Each convolutional layer has kernel size 2 and 30 channels. In the end, there is a linear layer to map the last feature map to z, which has 400 dimensions. The Direct-$\hat{p}$-Decoder

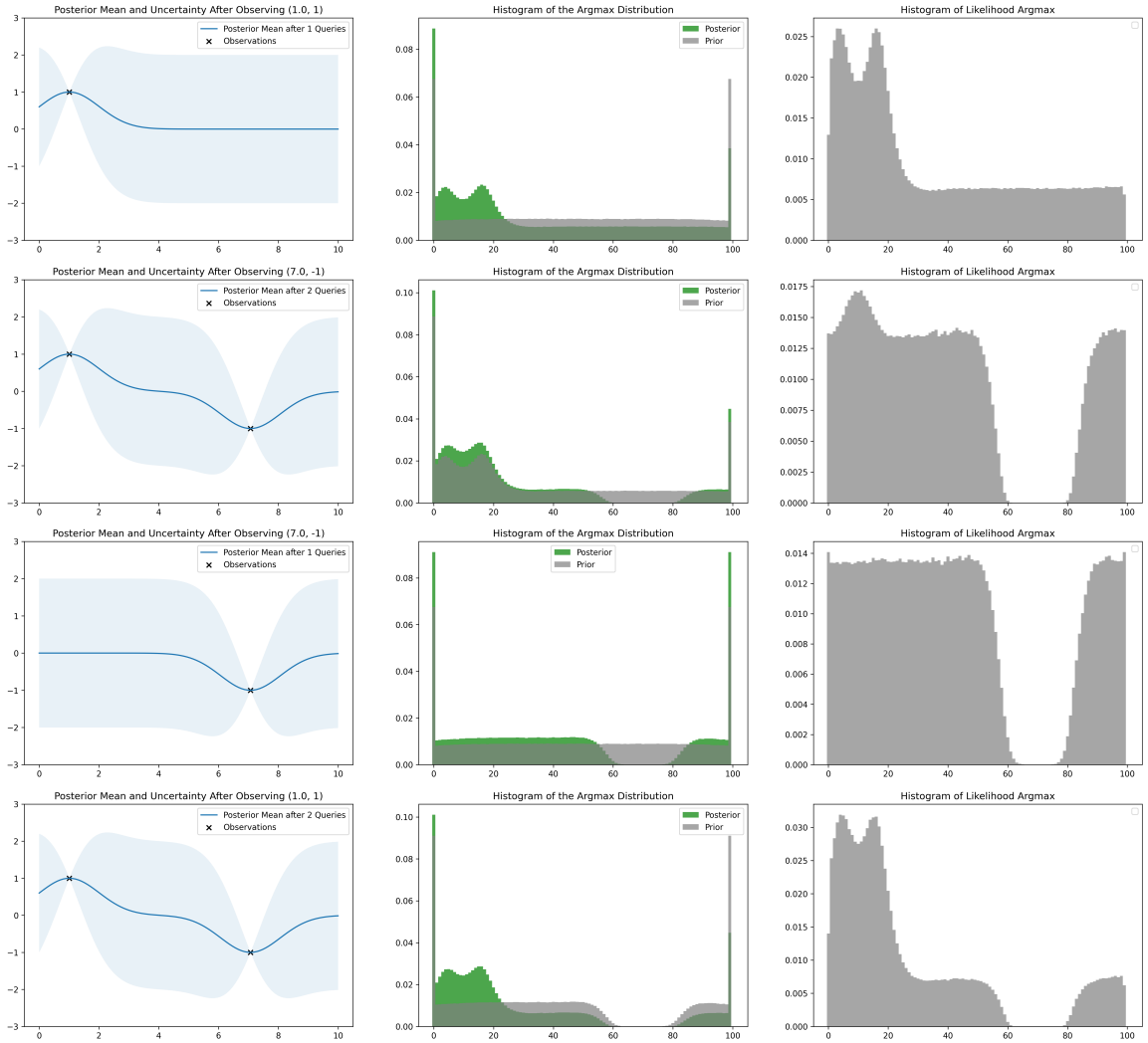

Figure 4: Left figures for the state of the GP, green is prior and gray is posterior in the middle figure, right figure is the supposed 'likelihood'. The top two rows show the states of a GP when it receives the samples (1, 1) and (7, -1) in sequence. The bottom two rows is the same with observation order reversed. 3M sample draws are used to compute empirical likelihoods.

is a similar CNN with 6 repetitions of convolutional, upsampling, 0.5 dropout, ReLU, and batch-norm layers. There is a skip from input to the third layer and from third to the final layer. All convolutional layers have kernel size 3 and 30 channels. In the very beginning, there is a linear layer to map z to appropriate size and shape. The ρ-Decoder is an MLP with hidden layer sizes $[360, 320, 280, 240, 200]$, skip connections from the input to the third layer, and from there to the output. It also has dropout with 0.5 probability for regularization.

Our 2D model uses the same building blocks but is similar. Its y-Encoders have the hidden layer sizes $[16, 24, 80, 160, 280]$, Direct-π-Encoder and Direct-$\hat{p}$-Decoder has 15 channels, the ρ-Decoder has hidden layers $[180, 160, 140, 120, 100]$, and z is 200 dimensional.

Our 1D model's y-Encoders have the hidden layer sizes $[4, 6, 20, 40, 70]$, Direct-π-Encoder and Direct-$\hat{p}$-Decoder has 7 channels, the ρ-Decoder has hidden layers $[90, 80, 70, 60, 50]$, and z is 100 dimensional.

For $t = 1$, we set $y^* = -10$. We use Adam optimizer with lr=0.001, and $(0.9, 0.999)$ beta values, for all models. We train for 500 epochs with early stopping. We apply weight decay of 1 to the parameters of Direct-π-Encoder and Direct-$\hat{p}$-Decoder, and 1e-4 to the rest. To ensure numerical stability, we add 1e-15 to probability densities we receive, and keep the same behaviour in evaluation. (This avoids 0 probability regions.)

Appendix E. Additional Experiments

The context C_t and distribution shift. All experiments operate under distribution shift, as approximation error places the system in states unseen during training. Although methods such as dataset aggregation [18] could mitigate this, we observed that a compact summary C_t of the history already stabilises learning. In preliminary runs without C_t, regret diverged after roughly ten timesteps. We hypothesised that supplying the network with the best observed value y^*_{t-1} would help, and indeed this modification resolved the divergence. Empirically, enlarging C_t or including the location $\mathbf{x}^*_{t-1}$ provided no additional benefit.

Convergence of regret. TS with GPs converges to zero regret, and we aim for our models to exhibit the same behaviour. To test longer horizons we generated new data with $N = 5000$, $T = 100$, $M = 100{,}000$, and lengthscale 1.0 on a 3D domain, and trained a 677k-parameter model. Though intermediate performance is weaker due to the smaller model, the regret converges to zero within 100 steps (Figure 5, left).

Out-of-distribution acquisitions. We also evaluated generalisation to acquisition functions that differ from those used in training. A model was trained on data produced by a random policy over the interval $[0, 10]$ (100 grid points) with lengthscale 1.0 in 1D. Despite the mismatch, the model performs reasonably under TS (Figure 5, right), suggesting it is possible to train on diverse or random policies and deploy the model with any acquisition function.

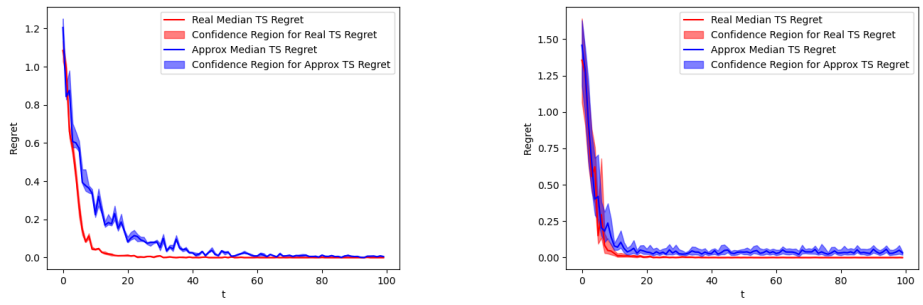

Figure 5: Regret curves for (left) long-horizon 3D experiments and (right) OOD generalization in 1D. 1000 and 100 parallel episodes are used, respectively.

Details for the GPT experiment. We min-max scaled the domain and add a Gaussian with covariance 0.5 around each guess. Note that the third response did not have any confidence reported, so we dropped it. Finally, we added uniform distributions based on the probability that the LLM assigns to the event that optimum is not near any of the suggested points. Finally, if the LLM uses phrases like "20-30%" we treated it as 25%.

Here is our prompt:

```
Hey, I want to optimize the production of o-xylenyl adducts of Buck-
minsterfullerenes.  You can vary three process conditions (tempera-
ture, reaction time and ratio of sultine to C60) to maximize the mole
fraction of the desired product.  Experiments are executed on a three
factor fully factorial grid with six levels per factor.  the ranges
for each condition are as follows:
  reaction_time:  [ 3, 31] hours
temperature:  [100, 150] degrees Celsius
sultine to C60 ratio:  [ 1.5, 6.0]
  Give 5 suggestions to try where you expect the minimizer to be close
to and format them in triples.  You can elaborate on your reasoning
and also express how likely you think the minimizer will be in the vicin-
ity of each of these points.  Also, give the likelihood for the op-
timizer is not in the vicinity of any of these points, and simply you're
not confident in your guess.
```

And here are the LLM generated responses:

```
1. To optimize the production of o-xylenyl adducts of
   ↪  Buckminsterfullerenes (C60), we can explore combinations
   ↪  of the three process conditions: reaction time,
   ↪  temperature, and sultine to C60 ratio. Given the
   ↪  constraints and ranges provided, I will suggest five sets
   ↪  of conditions that could lead to an optimal mole fraction
   ↪  of the desired product.

   ### Suggested Experimental Conditions

   1. **(Reaction Time: 3 hours, Temperature: 100C, Sultine to
   ↪  C60 Ratio: 1.5)**
      - **Reasoning:** This point represents the lower bounds of
      ↪  all factors. It could help establish a baseline for
      ↪  reaction efficiency at the minimum conditions. It's
      ↪  likely that the product yield will be low here, but it
      ↪  can inform how reaction time and temperature affect
      ↪  the rate of reaction.

   2. **(Reaction Time: 31 hours, Temperature: 150C, Sultine to
   ↪  C60 Ratio: 6.0)**
```

```
   - **Reasoning:** This combination tests the upper extremes
    ↪  of each factor. A long reaction time at a high
    ↪  temperature may favor the formation of the desired
    ↪  adducts, potentially increasing mole fraction
    ↪  significantly. However, it may also lead to side
    ↪  reactions or degradation, so it's a critical test
    ↪  point.

3. **(Reaction Time: 15 hours, Temperature: 125C, Sultine to
↪  C60 Ratio: 4.0)**
   - **Reasoning:** This midpoint setting balances between
    ↪  extremes and aims to capture a moderate reaction
    ↪  condition that may enhance selectivity toward the
    ↪  desired product. It's reasonable to hypothesize that
    ↪  this combination could yield a high mole fraction
    ↪  without pushing any individual factor to extremes.

4. **(Reaction Time: 20 hours, Temperature: 140C, Sultine to
↪  C60 Ratio: 5.0)**
   - **Reasoning:** This combination leans slightly towards
    ↪  the higher temperature and time but maintains a
    ↪  mid-range sultine to C60 ratio. It may help reveal the
    ↪  synergistic effects of temperature and time on the
    ↪  reaction kinetics while ensuring sufficient reactant
    ↪  concentration.

5. **(Reaction Time: 10 hours, Temperature: 120C, Sultine to
↪  C60 Ratio: 3.0)**
   - **Reasoning:** This condition tests a lower reaction
    ↪  time with moderate temperature and a balanced sultine
    ↪  ratio. It allows for evaluating how reducing the
    ↪  reaction time impacts yield while still keeping the
    ↪  other parameters favorable for product formation.

### Likelihood Analysis
- I estimate that the minimizer (optimal condition for
↪  maximum mole fraction) is likely to be **within the
↪  vicinity of the points** suggested, especially around
↪  points 3 and 4, where we balance temperature and reaction
↪  time effectively. Given the factorial nature of the
↪  experiment, we can anticipate that the optimal conditions
↪  would not stray too far from these balanced settings.
```

```
   - The likelihood that the optimizer is **not in the vicinity
    ↪  of any of these points** is relatively lower, but I would
    ↪  estimate it at around **20%**. This suggests that while
    ↪  our suggestions are grounded in reasoning based on the
    ↪  experimental design, there is still a possibility that
    ↪  the true optimum lies outside the proposed conditions.

   In conclusion, these suggested conditions should provide a
    ↪  productive starting point for your experiments, and I
    ↪  recommend monitoring the mole fraction closely to
    ↪  identify further opportunities for optimization based on
    ↪  initial results.

2. To optimize the production of o-xylenyl adducts of
    ↪  Buckminsterfullerenes (C60), we can explore a systematic
    ↪  approach to varying the three process conditions:
    ↪  reaction time, temperature, and the ratio of sultine to
    ↪  C60. Given the ranges for each condition and the
    ↪  complexity of the reaction, a factorial design is
    ↪  appropriate.

   ### Suggested Experimental Conditions

   Here are five suggestions for the experimental conditions,
    ↪  formatted as triples (reaction_time, temperature,
    ↪  sultine_to_C60_ratio):

   1. **(3 hours, 100°C, 1.5 ratio)**
      - **Reasoning:** This point represents the lower bounds
       ↪  for all factors. It may provide insights into the
       ↪  initial reaction kinetics and whether sufficient
       ↪  reaction occurs at low conditions.
      - **Likelihood of Minima:** This is less likely to yield
       ↪  high product formation, as both time and temperature
       ↪  are at their lower limits. However, it could be useful
       ↪  for understanding the reaction base line.

   2. **(31 hours, 150°C, 6.0 ratio)**
      - **Reasoning:** This condition represents the upper
       ↪  bounds for all factors, potentially leading to maximum
       ↪  product formation due to increased energy and
       ↪  prolonged reaction time.
```

```
   - **Likelihood of Minima:** High likelihood of being close
    ↪  to the optimal point due to the extensive reaction
    ↪  time and elevated temperature, which might enhance
    ↪  reactivity.

3. **(15.5 hours, 125°C, 3.75 ratio)**
   - **Reasoning:** This middle point offers a balanced
    ↪  approach, allowing for moderate conditions that can
    ↪  help identify the effect of each variable without
    ↪  being extreme.
   - **Likelihood of Minima:** Moderate likelihood; as a
    ↪  central point, it may yield valuable insights into the
    ↪  interaction effects between factors.

4. **(20 hours, 140°C, 6.0 ratio)**
   - **Reasoning:** This setting keeps the temperature high
    ↪  and the sultine to C60 ratio at its maximum, while
    ↪  reducing the reaction time, which may favor the
    ↪  formation of adducts without leading to excessive
    ↪  byproducts.
   - **Likelihood of Minima:** High likelihood of being close
    ↪  to the optimal point, as it combines elevated
    ↪  temperature and high reagent availability with a
    ↪  reasonable reaction time.

5. **(10 hours, 150°C, 1.5 ratio)**
   - **Reasoning:** This combination features maximum
    ↪  temperature with a minimal reagent ratio, which could
    ↪  drive the reaction effectively while limiting excess
    ↪  reagent that may produce side reactions.
   - **Likelihood of Minima:** Moderate likelihood; while the
    ↪  temperature is optimal, the low sultine ratio may
    ↪  limit product formation.

### Confidence and Likelihood Assessment

- **Overall Confidence:** I believe that the optimizer is
 ↪  likely to be in the vicinity of one or more of the
 ↪  suggested points, particularly the extremes (2 and 4) due
 ↪  to their high energy conditions and long reaction times.
 ↪  Given the nature of chemical reactions, extreme
 ↪  conditions often yield more product, although they can
 ↪  also lead to side reactions.
```

```
   - **Likelihood of Not Being Close:** There is still a chance
    ↪ that the optimal conditions lie outside the suggested
    ↪ points, particularly if there are nonlinear effects or
    ↪ interactions between the factors that are not captured by
    ↪ a linear model. Thus, I would estimate a **20% likelihood
    ↪ that the optimizer is not in the vicinity of any of these
    ↪ points**.

   This approach balances exploration of extremes while also
    ↪ considering mid-range conditions, which can help to
    ↪ locate the optimal settings for maximizing the mole
    ↪ fraction of the desired product.

3. To optimize the production of o-xylenyl adducts of
    ↪ Buckminsterfullerenes, we need to consider the three
    ↪ factors: reaction time, temperature, and the sultine to
    ↪ C60 ratio. Given that we are working with a three-factor
    ↪ fully factorial grid and that you have specified the
    ↪ ranges and levels for each factor, I will provide five
    ↪ suggested combinations to test, which I believe might
    ↪ yield favorable results based on the properties of the
    ↪ reactions involved.

   ### Suggested Combinations:

   1. **Combination 1**: (3 hours, 100°C, 1.5)
      - **Reasoning**: Starting with the lowest reaction time
       ↪ and temperature might help in observing the initial
       ↪ reaction kinetics. A low sultine to C60 ratio could
       ↪ favor the formation of the desired adduct without
       ↪ excessive side reactions.
      - **Likelihood of Minimizer**: There is a moderate chance
       ↪ that the minimizer is in this vicinity since it allows
       ↪ for the study of the reaction baseline.

   2. **Combination 2**: (15 hours, 125°C, 3.0)
      - **Reasoning**: This combination strikes a balance
       ↪ between moderate reaction time and temperature. The
       ↪ mid-level sultine to C60 ratio is also likely to
       ↪ enhance product formation while maintaining control
       ↪ over side products.
      - **Likelihood of Minimizer**: High likelihood, as it may
       ↪ represent an optimal setting for achieving a higher
       ↪ yield without pushing the system too far.
```

```
3. **Combination 3**: (31 hours, 150°C, 4.5)
   - **Reasoning**: This combination tests the upper limits
    ↪  of reaction time and temperature. A higher sultine to
    ↪  C60 ratio could also help in ensuring that the
    ↪  reaction goes to completion, though it risks side
    ↪  reactions.
   - **Likelihood of Minimizer**: Moderate likelihood, as
    ↪  extreme conditions can sometimes lead to unexpected
    ↪  results, but may also be suboptimal due to potential
    ↪  degradation of the fullerenes.

4. **Combination 4**: (20 hours, 140°C, 5.0)
   - **Reasoning**: This combination is slightly below the
    ↪  maximum temperature and at a high reaction time, which
    ↪  may favor product stability and yield. The sultine to
    ↪  C60 ratio is also elevated, which should encourage
    ↪  adduct formation.
   - **Likelihood of Minimizer**: High likelihood, as it
    ↪  balances time and temperature effectively, allowing
    ↪  for extended reaction time under favorable conditions.

5. **Combination 5**: (10 hours, 110°C, 6.0)
   - **Reasoning**: This combination tests a lower
    ↪  temperature with a high sultine to C60 ratio. It could
    ↪  be interesting to see if lower energy input with an
    ↪  excess of sultine could still yield high adduct
    ↪  production without excessive side reactions.
   - **Likelihood of Minimizer**: Moderate likelihood; while
    ↪  the temperature is lower, the high ratio may still
    ↪  lead to a good yield.

### Overall Confidence in Suggestions:
The likelihood that the optimizer is not in the vicinity of
 ↪  any of these points is relatively low, given that these
 ↪  combinations span a diverse range of the experimental
 ↪  space. However, there is always a possibility that
 ↪  optimal conditions could exist outside this grid,
 ↪  especially if there are nonlinear interactions among the
 ↪  factors or if the reaction dynamics are complex. Thus,
 ↪  while I am confident in the combinations suggested, I
 ↪  would still advise conducting exploratory runs beyond
 ↪  these points, especially if initial results indicate
 ↪  unexpected trends.
```

4. To optimize the production of o-xylenyl adducts of
   ↪ Buckminsterfullerenes, we can systematically explore the
   ↪ parameter space defined by your three factors: reaction
   ↪ time, temperature, and the ratio of sultine to C60. Given
   ↪ the ranges for each factor and the fully factorial grid
   ↪ design, we can suggest five specific combinations
   ↪ (triples) that are likely to yield favorable results
   ↪ based on typical chemical reaction behavior.

   ### Suggested Experimental Conditions:

   1. **(Reaction Time: 3 hours, Temperature: 100C, Sultine to
   ↪ C60 Ratio: 1.5)**
      - **Reasoning**: This combination represents the lower
      ↪ limits of all three factors. While it may lead to
      ↪ lower conversion rates, it allows for initial
      ↪ assessments of the reactivity and stability of the
      ↪ system under mild conditions.

   2. **(Reaction Time: 31 hours, Temperature: 150C, Sultine to
   ↪ C60 Ratio: 6.0)**
      - **Reasoning**: This combination represents the upper
      ↪ extremes of all factors. Longer reaction times and
      ↪ higher temperatures are likely to enhance reactivity
      ↪ and yield but may also lead to side reactions. This
      ↪ setup is useful to probe the limits of the system.

   3. **(Reaction Time: 15 hours, Temperature: 125C, Sultine to
   ↪ C60 Ratio: 4.5)**
      - **Reasoning**: This middle-ground combination balances
      ↪ time, temperature, and reactant ratio. It is often
      ↪ advantageous to explore such intermediate conditions
      ↪ to identify optimal yield without risking excessive
      ↪ degradation or unwanted byproducts.

   4. **(Reaction Time: 10 hours, Temperature: 150C, Sultine to
   ↪ C60 Ratio: 3.0)**
      - **Reasoning**: Here, were testing a high temperature
      ↪ with a moderate reaction time and reactant ratio. High
      ↪ temperatures can enhance the yield, and the moderate
      ↪ time allows for a compromise between efficiency and
      ↪ the risk of degradation.

   5. **(Reaction Time: 25 hours, Temperature: 100C, Sultine to
   ↪ C60 Ratio: 6.0)**

```
   - **Reasoning**: This combination keeps the temperature
    ↪ low while extending the reaction time and increasing
    ↪ the sultine to C60 ratio. It is expected to maximize
    ↪ product formation while minimizing the risk of side
    ↪ reactions at high temperatures.

### Likelihood Assessment:

- **Proximity of Minimizer**: I believe that the minimizer
 ↪ (the combination that maximizes the mole fraction of the
 ↪ desired product) could be reasonably expected to be near
 ↪ the middle-ground conditions, particularly suggestion #3.
 ↪ This is a typical region where many optimizations find
 ↪ favorable results due to a balance of reactivity,
 ↪ stability, and yield.

- **Likelihood of Not Being Close**: There is a moderate
 ↪ likelihood (around 20-30%) that the optimizer is not in
 ↪ the vicinity of any of these suggested points. Reaction
 ↪ behavior can be complex, and the actual optimal
 ↪ conditions may be influenced by specific kinetics or
 ↪ thermodynamic factors not fully captured in the initial
 ↪ range, or they may be affected by unforeseen interactions
 ↪ between the reactants.

Overall, I recommend starting with these suggested
 ↪ combinations and closely monitoring the results to refine
 ↪ further based on experimental feedback.

5. Optimizing the production of o-xylenyl adducts of
    ↪ Buckminsterfullerenes involves careful selection of
    ↪ reaction conditions. Given the specified ranges for the
    ↪ three factors:

   1. **Reaction Time:** [3, 31] hours
   2. **Temperature:** [100, 150] degrees Celsius
   3. **Sultine to C60 Ratio:** [1.5, 6.0]

   With a three-factor fully factorial design and six levels per
    ↪ factor, there are a total of 216 unique combinations to
    ↪ explore. However, I can suggest five specific
    ↪ combinations based on theoretical reasoning and general
    ↪ trends observed in similar chemical reactions.

   ### Suggested Points for Experimentation
```

```
1. **Point 1:** (3 hours, 100 °C, 1.5 ratio)
   - This combination features the minimum reaction time and
    ↪  temperature, which may promote rapid formation of
    ↪  intermediates but could lead to incomplete reactions.
    ↪  Its a good starting point for determining the baseline
    ↪  yield.
   - **Likelihood of being close:** Medium; this may help
    ↪  establish a lower bound for product yield.

2. **Point 2:** (31 hours, 150 °C, 6.0 ratio)
   - This point represents maximum extremes for all factors,
    ↪  potentially leading to high yields if the reaction is
    ↪  indeed thermally activated and the abundance of
    ↪  sultine drives the reaction to completion.
   - **Likelihood of being close:** High; the extended
    ↪  reaction time and high temperature could promote
    ↪  maximum reaction efficiency.

3. **Point 3:** (15 hours, 125 °C, 3.0 ratio)
   - This combination strikes a balance between reaction time
    ↪  and temperature, with a moderate sultine to C60 ratio.
    ↪  This is often a sweet spot in many reactions, allowing
    ↪  for good product formation without excessive
    ↪  degradation.
   - **Likelihood of being close:** Medium-High; its
    ↪  plausible that moderate conditions will yield
    ↪  favorable results.

4. **Point 4:** (15 hours, 150 °C, 1.5 ratio)
   - This condition has an elevated temperature but a lower
    ↪  sultine to C60 ratio. It could potentially favor the
    ↪  reaction kinetics but may not provide enough sultine
    ↪  for optimal yields.
   - **Likelihood of being close:** Medium; the temperature
    ↪  is high, but the low ratio could limit product
    ↪  formation.

5. **Point 5:** (31 hours, 100 °C, 3.0 ratio)
   - Here, the extended reaction time may allow for complete
    ↪  reaction at a lower temperature. This might help avoid
    ↪  thermal degradation of the product, making it a
    ↪  cautious choice for maximizing yield.
```

```
   - **Likelihood of being close:** Medium; while the
    ↪  extended time is beneficial, the low temperature may
    ↪  hinder reaction rate.

### General Confidence Assessment

Overall, I estimate that the optimizer is likely to be in the
 ↪  vicinity of one of these suggested points (around 60-70%
 ↪  confidence). However, there remains a significant chance
 ↪  (30-40%) that the true optimal set of conditions lies
 ↪  outside these points, especially if the non-linear
 ↪  interactions between factors play a significant role.

Applying a response surface methodology or a design of
 ↪  experiments approach after these initial trials could be
 ↪  beneficial to further refine the conditions and identify
 ↪  the true optimal parameters.
```

Appendix F. Additional Figures

Here, we provide log-scale versions of the regret figures in the main paper in Figure 6.

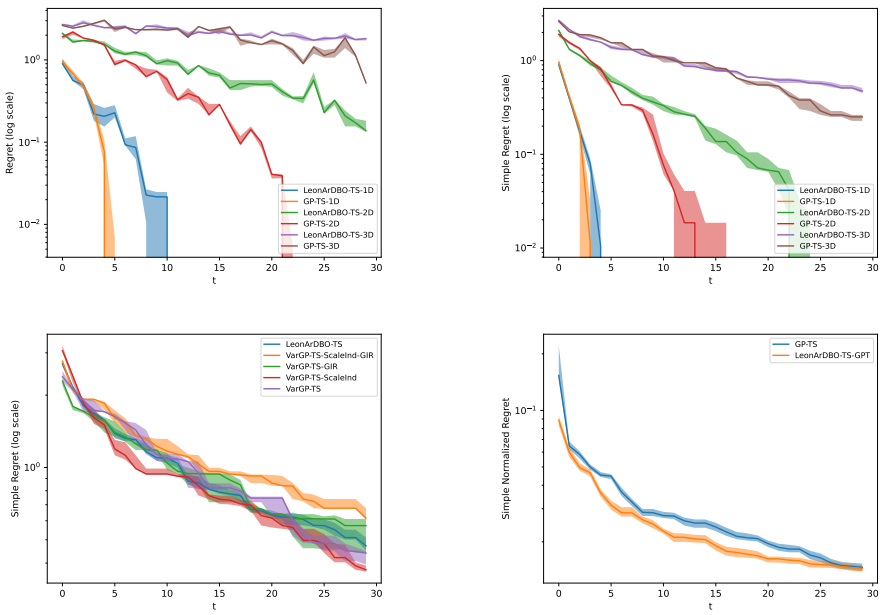

Figure 6: Regret curves for (left) long-horizon 3D experiments and (right) OOD generalization in 1D. 1000 and 100 parallel episodes are used, respectively.