# Sparse Bayesian Optimization

**Sulin Liu**[*,1] **Qing Feng**[*,2] **David Eriksson**[*,2] **Benjamin Letham**[2] **Eytan Bakshy**[2]

[1]Princeton University
[2]Meta

## Abstract

Bayesian optimization (BO) is a powerful approach to sample-efficient optimization of black-box objective functions. However, the application of BO to areas such as recommendation systems often requires taking the interpretability and simplicity of the configurations into consideration, a setting that has not been previously studied in the BO literature. To make BO applicable in this setting, we present several regularization-based approaches that allow us to discover sparse and more interpretable configurations. We propose a novel differentiable relaxation based on homotopy continuation that makes it possible to target sparsity by working directly with $L_0$ regularization. We identify failure modes for regularized BO and develop a hyperparameter-free method, sparsity exploring Bayesian optimization (SEBO) that seeks to simultaneously maximize a target objective and sparsity. SEBO and methods based on fixed regularization are evaluated on synthetic and real-world problems, and we show that we are able to efficiently optimize for sparsity.

## 1 Introduction

Bayesian optimization (BO) is a technique for efficient global optimization that is used for parameter optimization across a wide range of applications, including robotics [25, 5], machine learning pipelines [21, 33, 36], internet systems [24, 16], and chemistry [17, 15]. In many applications, including those just mentioned, it is preferable for the optimized parameters to be sparse. One reason to prefer sparsity is that it increases interpretability, a consideration that has recently attracted a great deal of attention in machine learning [9, 29]. Interpretability is necessary for humans to be able to understand and evaluate the outputs of complex systems—the types of systems to which BO is often applied. In policy optimization, sparsity of the control policy provides a natural way for human decision-makers to gain insight into the behavior of the system, and identify potential issues [37, 20]. Besides interpretability, sparsity can also be beneficial by producing systems that are easier to deploy and maintain, reducing the "tech debt" of machine learning systems [30]. In chemistry, a sparse solution may require fewer reagents and steps to synthesize a compound.

Sparsity in machine learning is often achieved via regularization, such as $L_1$ regularization used by the lasso [35], the group norm penalty used by the group lasso [43], and $L_0$ regularization which directly targets setting elements to zero [44]. The purpose of regularization in machine learning is typically to improve accuracy by reducing generalization error [14]. In our setting, sparsity is a separate goal; interpretable sparse configurations will generally not improve the optimization objective, and in fact, may come at some cost to other metrics. A central aspect of this work is to efficiently learn these trade-offs and offer practitioners a way to balance sparsity and other metrics.

Sparsity in BO is an important topic that has not yet been addressed in the literature. Past work has used regularization in acquisition function optimization or modeling, but not for the purpose of sparsity in design parameters. Our work provides a thorough and broad treatment of sparsity in BO that fills in this gap. The main contributions of this paper are:

[*]Equal contribution

2022 NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems.

1. We study different approaches for incorporating sparse regularization into BO, and provide negative theoretical results showing that previously studied forms of regularization can fail to optimize for certain levels of sparsity, regardless of the regularization coefficient.
2. We draw connections between multi-objective BO and acquisition function regularization, and show how multi-objective BO can be used for automatic selection of the regularization coefficient. We refer to this as the SEBO ("Sparsity Exploring Bayesian Optimization") method.
3. We develop a novel relaxation strategy for optimizing directly for $L_0$ sparsity, and show that it significantly outperforms the typical $L_1$ penalty in our context.
4. We provide the first results on achieving sparsity via BO, in a range of synthetic functions and on three real-world tasks (in systems configuration and AutoML). We also show the breadth of our method by using it to achieve different forms of sparsity such as feature-level and group sparsity.
5. We provide a new high-dimensional benchmark problem designed to emulate trade-offs found in real-world recommender systems, and show how such systems benefit from increased sparsity.

In Section 2, we first describes two natural approaches for incorporating sparse regularization into acquisition function optimization, both of which can fail to optimize for some levels of sparsity. Then we discuss a relationship between sparse BO and multi-objective BO, and describe how we can use methods from multi-objective BO to simultaneously optimize for all levels of sparsity. We further describe how we are able to directly optimize with $L_0$ regularization. We demonstrate the usefulness of our methods by applying them to a set of synthetic and real-world benchmarks in Section 3.

## 2 Methodology

Consider a Bayesian optimization problem with $f$ being the objective function to optimize and $\mathbf{x} \in \mathbb{R}^D$ being the parameters. (See Appendix S1 for a background review of BO). We use a penalty term $\xi(\mathbf{x})$ to measure the sparsity of $\mathbf{x}$. The penalty may be an $L_0$ norm to target feature-level sparsity, $\xi(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}^s\|_0$, or can be adjusted for different forms of sparsity such as group sparsity. Here, $\mathbf{x}^s$ is a user-defined point anywhere in the domain that the user wishes to regularize towards. For instance, it can be an existing system configuration, and the regularization can help with reducing changes to the system, for the purpose of interpretability, maintaining system consistency, and safety.

**Regularization in acquisition functions** Perhaps the most straightforward approach for adding regularization to acquisition optimization is simply to add a regularization penalty directly to the acquisition function. This parallels the approach taken in regression with techniques like ridge regression and the lasso. Given a penalty term $\xi(\mathbf{x})$, we then maximize

$$\alpha_{\text{ER}}(\mathbf{x}; \lambda) = \alpha(\mathbf{x}) - \lambda\xi(\mathbf{x}) \tag{1}$$

to select the next point for evaluation. We refer to this approach as *external regularization* (ER).

An alternative approach for adding regularization to the acquisition optimization is to add it directly to the objective function. In this approach, instead of using the posterior of $f$ to compute the acquisition function, we compute the acquisition for the posterior of a regularized function:

$$g(\mathbf{x}; \lambda) = f(\mathbf{x}) - \lambda\xi(\mathbf{x}). \tag{2}$$

We refer to this as *internal regularization* (IR). In Appendix S2, we provide more details on the differences of the two approaches and how they behave under the most commonly used expected improvement (EI) acquisition function.

There are two fundamental challenges with both of the regularization approaches. The first is that they both have a regularization coefficient $\lambda$ that must be set. In sparse BO, if there is a known desired level of sparsity, $\lambda$ can be swept to find a value that produces candidates with the desired level of sparsity. But often times, the desired level of sparsity is typically not known *a priori* in real applications. When there is a trade-off between interpretability and system performance, the desired level of interpretability will depend on what that trade-off looks like. In practice, we thus wish to identify the best-achievable objective at any particular level of sparsity. The second challenge is that, due to theoretical limitations of ER and IR identified in our analysis (Propositions 1 and 2 in Appendix S2), we may not be able to find the entire objective vs. sparsity trade-off using ER or IR, no matter how $\lambda$ is swept. Depending on the problem, it may be that the sparsity levels of interest cannot be explored via either regularization strategy. Next, we show that both of these challenges can be addressed by viewing sparse BO from the lens of multi-objective BO.

**Sparse BO as Multi-Objective BO** In this section we introduce the Sparsity Exploring Bayesian Optimization method (SEBO), which takes a multi-objective approach to sparse BO. Rather than considering $\xi$ as a penalty applied to the objective, we consider $f$ and $-\xi$ to each be objectives that we wish to maximize. Casting sparse BO as Multi-Objective BO (MOO) of the objective and sparsity has several advantages. It provides a solution for setting the regularization coefficient $\lambda$, e.g. $f - \lambda\xi$, which is required as in the classical regularized regression setting. The coefficient $\lambda$ decides the desired level of sparsity which is often unknown in real applications. When there is a trade-off between interpretability and system performance, the desired level of interpretability will depend on what that trade-off looks like. Instead, we can use MOO methods such as Expected Hypervolume Improvement (EHVI) to select points that maximize performance for all levels of sparsity, or equivalently, maximize sparsity for all levels of performance, explicitly optimizing for the entire regularization path. The goal of MOO is to identify the optimum for every level of sparsity, which enables decision makers to make an informed trade-off between interpretability and other considerations of system performance.

In our experiments, we use the EHVI acquisition function. Here, the hypervolume improvement is defined with respect to a worst-case reference point $\mathbf{r} = [r_f, r_\xi]$, can be set to estimates for the minimum and maximum values of $f$ and $\xi$ respectively. Given a set of observations $X^{\text{obs}} = \{\mathbf{x}^1, \dots, \mathbf{x}^n\}$, the Pareto hypervolume of that set is defined as $V(X^{\text{obs}}) = \lambda_M \left( \bigcup_{i=1}^n \left( [r_f, r_\xi] \times [f(\mathbf{x}^i), \xi(\mathbf{x}^i)] \right) \right)$, where $\lambda_M$ denotes the Lebesgue measure. The expected hypervolume improvement is computed as

$$\alpha_{\text{SEBO}}(\mathbf{x}) = \mathbb{E}_f \left[ V(X^{\text{obs}} \cup \{\mathbf{x}\}) - V(X^{\text{obs}}) \right]. \tag{3}$$

We call this acquisition function SEBO, and explore its performance in combination with the $L_0$ sparse regularization, described next. This acquisition function is hyperparameter-free, and, as we will see, is highly effective for sparse BO.

**Optimizing Acquisition Functions with $L_0$ Sparsity** Optimizing acquisition functions with $L_0$ sparsity is challenging since the $L_0$ norm is discontinuous. We leverage the idea of homotopy continuation [1] that defines a homotopy $H(\mathbf{x}, a)$, where $H(\mathbf{x}, a_{\text{start}})$ corresponds to a problem that is easy to solve and $H(\mathbf{x}, a_{\text{end}})$ corresponds to the target problem. In particular, for $a > 0$ we define $H(\mathbf{x}, a) = \mathbb{E}_f[u([f(\mathbf{x}), \varphi_a(\mathbf{x})])]$ where $\varphi_a(\mathbf{x}) := D - \sum_{i=1}^D \exp\left(-0.5\,(\mathbf{x}_i/a)^2\right) \approx \|\mathbf{x}\|_0$. Note that under the assumption of a continuous utility $u(x)$ , we have that $\lim_{a \to 0^+} H(x, a) = \mathbb{E}_f[u([f(x), \|\mathbf{x}\|_0])]$, which corresponds to the original acquisition function with the $L_0$ norm.

We will start at some value $a_{\text{start}}$ large enough to make the acquisition function easy to optimize and decrease $a$ towards $a_{\text{end}} = 0$. Each time we change $a$ we re-optimize the acquisition function starting from the best solution found for the previous value of $a$. This idea is illustrated in Fig. 1 where we consider the 1D problem of using SEBO to optimize $f(x) = -x^2$ with an $L_0$ penalty $\|x - 0.5\|_0$ and assume $X^{\text{obs}} = \{0, 0.25, 0.75, 1.0\}$ have already been evaluated.

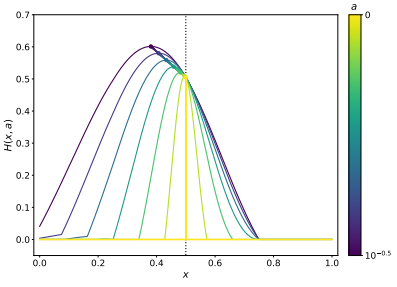


Figure 1: 1D problem to optimize $-x^2$ with an $L_0$ penalty $\|x - 0.5\|_0$. The global optimum of the acquisition function is given by the sparse point $x = 0.5$. Optimizing the acquisition function along the continuous homotopy path starting at $a_{\text{start}} = 10^{-0.5}$ allows us to eventually uncover find the true optimum of $x = 0.5$.

## 3 Experiments

We evaluate EI-IR, EI-ER and SEBO on two synthetic and three real-world problems. We compare performance to quasi-random search (Sobol), BO with a standard ARD Matérn-$5/2$ kernel and the EI acquisition function (GPEI), and SAASBO. Our experiments all have high-dimensional parameter spaces, so we use the SAAS GP model in [12] when optimizing with SEBO. We show the results using $L_0$ regularization for most problems except for the adaptive bitrate simulation (ABR) problem, where the group lasso is used to demonstrate that the methods can be applied to recover different forms of sparsity, such as group sparsity. Due to limited space, we defer the experiments on synthetic problems and ABR to Appendix S5.2. Ablation studies on $L_0$ v.s. $L_1$, and the importance of using homotopy continuation are included in Appendix S6.

**Ranking sourcing system simulation:** The task is to optimize a 25-dimensional retrieval policy in the sourcing component of recommender systems. We developed a simulation of a recommender sourcing system that simulates the quality and infrastructure load of recommendations produced by a particular sourcing policy. The sourcing system is modeled as a topic model, where each source has a different distribution over topics, and topics have different levels of relevance to the user. When two sources are (topically) similar to one another, they may obtain duplicate items, which will not improve recommendation quality. Our goal is thus to identify a retrieval policy that uses a minimal number of sources while still maximizing the ranking quality score, measured by a function of content relevance and infrastructure load. Our desired sparsity is to set parameters to 0, i.e., turning off the source.

**SVM machine learning hyperparameter tuning:** We consider the problem of doing joint feature selection and hyperparameter ($C$, $\varepsilon$, and $\gamma$) tuning for a support vector machine (SVM). The scale factor for each feature is in the continuous range $[0, 1]$, where sparsity means removing the feature with scale factor 0. For hyperparameters, we took $C \in [0.01, 1.0]$, $\varepsilon \in [0.01, 1.0]$, and $\gamma \in [0.001, 0.1]$, where the center of each interval was considered sparse as this is the default value in Sklearn. We used 100 features from the CT slice UCI dataset [10] and the goal was to minimize the RMSE on the test set. This produces a 103D optimization problem.

The objective-sparsity trade-offs of the two problems are shown in Fig. 2. Sobol and GPEI could not find sparse policies. In general, IR and ER are more effective than SAASBO in finding sparse policies, which shows regularization helps. SEBO-$L_0$ performs the best in optimizing the objective under different sparsity levels, dominating all other methods across different problems.

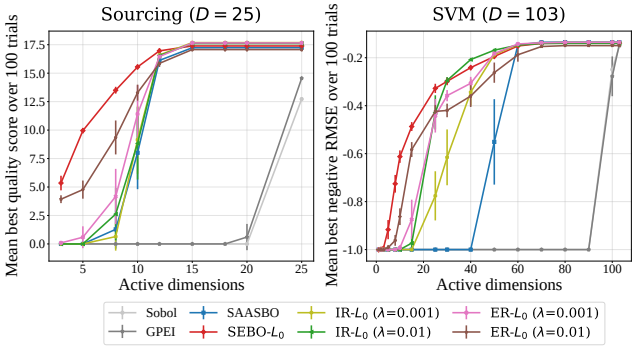


Figure 2: Objective-sparsity trade-offs after 100 trials for the two real-world problems. (Left) *Sourcing problem*: SEBO-$L_0$ regularization effectively explored all sparsity trade-offs. (Right) *SVM problem*: ER with $\lambda = 0.01$ and IR with $\lambda = 0.01$ were able to explore parts of the Pareto frontier, however were dominated by SEBO-$L_0$.

**Interpretation of sparse solutions:** We also examine what sources are selected in the recommender sourcing system problem to understand the obtained SEBO-$L_0$ solutions. Across 20 replications, we obtain the Pareto-optimal 25-dimensional retrieval policy and compute the average of retrievals per source at each sparsity level. For each source, we compute a source quality scores based on the simulation setup stated in S5.1. Fig. 3 visualizes the average number of items retrieved from each source at different sparsity levels. Each column corresponds to one source and is sorted based on source quality score in an ascending order (from left to right); each row represents the sparsity level (number of active dimensions). The color indicates the parameter values. It is observed that sources with low quality scores are turned off (zero query) and sources with higher scores have higher number of retrievals even with smaller active dimensions. This indicates that the SEBO sparse policy identifies the most effective sources and prioritizes to retrieve from the high-quality sources first.

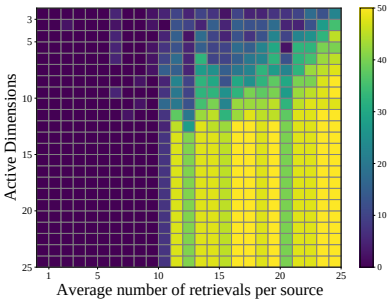


Figure 3: The heatmap of average retrieval policy values at different sparsity levels.

## References


[1] Eugene L Allgower and Kurt Georg. *Numerical continuation methods: an introduction*, volume 13. Springer Science & Business Media, 2012.

[2] 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*, NeurIPS, 2020.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[4] V. Joseph Bowman. On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In H. Thiriez and S. Zionts, editors, *Multiple Criteria Decision Making. Lecture Notes in Economics and Mathematical Systems (Operations Research), vol 130*, pages 76–86. Springer Berlin, Heidelberg, 1976.

[5] Roberto Calandra, André Seyfarth, Jan Peters, and Marc P. Deisenroth. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1): 5–23, 2015.

[6] Indraneel Das and John Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14:63–69, 1997.

[7] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems*, 33:9851–9864, 2020.

[8] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel Bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in Neural Information Processing Systems, 34*, 2021.

[9] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[10] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[11] Matthias Ehrgott. *Multicriteria Optimization*. Springer Berlin, Heidelberg, 2005.

[12] David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*, UAI, pages 493–503, 2021.

[13] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[14] Theodoros Evgeniou, Tomaso Poggio, Massimiliano Pontil, and Alessandro Verri. Regularization and statistical learning theory for data analysis. *Computational Statistics & Data*, 38(4): 421–432, 2002.

[15] Kobi Felton, Jan Rittig, and Alexei Lapkin. Summit: Benchmarking machine learning methods for reaction optimisation. *Chemistry Methods*, 1(2):116–122, 2021.

[16] Qing Feng, Benjamin Letham, Hongzi Mao, and Eytan Bakshy. High-dimensional contextual policy search with unknown context rewards using Bayesian optimization. In *Advances in Neural Information Processing Systems 33*, NeurIPS, pages 22032–22044, 2020.

[17] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

[18] Javier González, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch Bayesian optimization via local penalization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, AISTATS, pages 648–657, 2016.

[19] José Miguel Henrández-Lobato, Matthew W. Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems 27*, NIPS, pages 918–926, 2014.

[20] Xiyang Hu, Cynthia Rudin, and Margo Seltzer. Optimal sparse decision trees. In *Advances in Neural Information Processing Systems 32*, NeurIPS, pages 7267–7275, 2019.

[21] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*, LION, pages 507–523, 2011.

[22] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[23] Joshua Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

[24] Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.

[25] Daniel J. Lizotte, Tao Wang, Michael Bowling, and Dale Schuurmans. Automatic Gait Optimization with Gaussian Process Regression. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, IJCAI, pages 944–949, 2007.

[26] R. Timothy Marler and Jasbir S. Arora. The weighted sum method for multi-objective optimization: New insights. *Structural and Multidisciplinary Optimization*, 41:853–862, 2010.

[27] Changyong Oh, Jakub M. Tomczak, Efstratios Gavves, and Max Welling. Combinatorial Bayesian optimization using the graph Cartesian product. In *Advances in Neural Information Processing Systems 32*, NeurIPS, pages 2914–2924, 2019.

[28] Chiwoo Park, David J. Borth, Nicholas S. Wilson, and Chad N. Hunter. Variable selection for Gaussian process regression through a sparse projection. *IISE Transactions*, pages 1–14, 2021.

[29] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85, 2022.

[30] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems 28*, NIPS, pages 2503–2511, 2015.

[31] 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.

[32] Bobak Shahriari, Alexandre Bouchard-Côté, and Nando de Freitas. Unbounded Bayesian optimization via regularization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, AISTATS, pages 1168–1176, 2016.

[33] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, NIPS, pages 2951–2959, 2012.

[34] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, ICML, pages 1015–1022, 2010.

[35] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

[36] Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, pages 3–26, 2021.

[37] Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

[38] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient Bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning*, ICML, pages 3627–3635, 2017.

[39] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, Nando De Freitas, et al. Bayesian optimization in high dimensions via random embeddings. In *IJCAI*, pages 1778–1784. Citeseer, 2013.

[40] Mark Wilhelm, Ajith Ramanathan, Alexander Bonomo, Sagar Jain, Ed H. Chi, and Jennifer Gillenwater. Practical diversified recommendations on youtube with determinantal point processes. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 2165–2173, New York, NY, USA, 2018. Association for Computing Machinery. doi: 10.1145/3269206.3272018.

[41] James Wilson, Frank Hutter, and Marc Deisenroth. Maximizing acquisition functions for Bayesian optimization. In *Advances in Neural Information Processing Systems 31*, NeurIPS, 2018.

[42] Kaifeng Yang, Michael Emmerich, André Deutz, and Thomas Bäck. Multi-objective Bayesian global optimization using expected hypervolume improvement gradient. *Swarm and Evolutionary Computation*, 44:945–956, 2019.

[43] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.

[44] Tong Zhang. Multi-stage convex relaxation for learning with sparse regularization. *Advances in neural information processing systems*, 21, 2008.

# Appendices for Sparse Bayesian Optimization

## S1 Background and Related Work

**Bayesian Optimization:** Shahriari et al. [31] provide a thorough review of BO. In short, the goal is to maximize a black-box function $f : \mathbb{R}^D \to \mathbb{R}$ over a compact set $\mathcal{B} \subset \mathbb{R}^D$. We will assume that $f$ is continuous and bounded on this domain. For simplicity, we also assume that the domain is the unit hypercube $[0, 1]^D$. At each iteration of optimization, $f$ is modeled with a Gaussian process (GP) given the function evaluations observed so far, producing the normally distributed posterior $f(\mathbf{x}) \sim \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$. The location of the next function evaluation is selected by maximizing an acquisition function $\alpha(\mathbf{x}) := \mathbb{E}_f[u(\mathbf{x})]$ where $u$ is a utility function that defines the acquisition function. Typical acquisition functions include the expected improvement [EI, 22] and the upper confidence bound [UCB, 34]. EI is given by

$$\alpha_{\text{EI}}(\mathbf{x}) = \mathbb{E}_f\left[(f(\mathbf{x}) - f(\mathbf{x}^*))_+\right], \tag{S1}$$

where $\mathbf{x}^*$ is the best point observed so far. EI has a well-known analytic form in terms of the marginal posterior mean and variance. UCB is similarly computed directly from the marginal posterior,

$$\alpha_{\text{UCB}}(\mathbf{x}) = \mu(\mathbf{x}) + \sqrt{\beta}\sigma(\mathbf{x}), \tag{S2}$$

where $\beta$ is a hyperparameter that controls the exploration-exploitation trade-off. More recently, information-theoretic acquisition functions have been developed [19, 38].

**Regularization in BO:** Regularization has been applied to acquisition function optimization, though not for the purpose of sparsity. Shahriari et al. [32] used regularization for unbounded BO, in which there are no bounds on the search space. They applied a form of $L_2$ regularization to the EI target value that penalized sampling points far from the initial center of the search space. González et al. [18] used regularization for batch BO, where the penalty discouraged points from being chosen close to points that had already been selected for the batch. This penalty is applied by multiplying the original acquisition function value by a penalty term.

**BO with Sparse Models:** Eriksson and Jankowiak [12] introduced the sparse axis-aligned subspaces (SAAS) function prior in which a structured sparse prior is induced over the inverse-squared kernel lengthscales $\{\rho_i\}_{i=1}^d$ to enable BO in high dimensions. The SAAS prior has the form $\tau \sim \mathcal{HC}(\alpha), \rho_i \sim \mathcal{HC}(\tau)$ where $\mathcal{HC}$ is the half-Cauchy distribution which concentrates at zero. The goal of the SAAS prior is to turn off unimportant parameters by shrinking $\rho_i$ to zero, which avoids overfitting in high-dimensional spaces, thus enabling sample-efficient high-dimensional BO. The global shrinkage parameter $\tau$ controls the overall model sparsity: as more observations are made, $\tau$ can be pushed to larger values, allowing the level of sparsity to adapt to the data as needed.

While sparsity in the GP model is different from the sparsity we seek here, we will show that combining the SAAS model with acquisition regularization is highly effective for sparse high-dimensional BO. By enforcing regularization in the acquisition function, the parameters identified as unimportant will be set to their baseline values, generating simpler and more interpretable policies. Other work has studied feature sparsity in GP regression but without considering sparsity in optimization [27, 28].

**Multi-Objective BO:** Multi-objective BO is used when there are several (often competing) objectives $f_1, \ldots, f_m$ and we wish to recover the Pareto frontier of non-dominated configurations. A popular method in this setting is ParEGO, which applies the standard single-objective EI acquisition function to a random scalarization of the objectives [23]. Many types of scalarizations have been developed for transforming multi-objective optimization (MOO) problems into single-objective problems [11]. Recent work on multi-objective BO has focused on developing acquisition functions that explicitly target increasing the hypervolume of the known Pareto frontier with respect to a pre-specified reference point. Acquisition functions in this class, such as Expected Hypervolume Improvement (EHVI), are considered state-of-the-art for multi-objective BO [42, 7, 8].

## S2 Regularization in Acquisition Functions

In this section, we analyze the properties and failure modes of both ER and IR approaches in details.

### S2.1 External Regularization

For our analysis of regularization in this paper, we will assume that $\mathbf{x}^s$ is the unique global minimum of $\xi(\mathbf{x})$. The regularization coefficient $\lambda$ must be set, just as in the classical regularized regression setting. This formulation separates the explore/exploit value of a point, encoded in $\alpha$, from its sparsity value, encoded in $\xi$. This can perform poorly, because there is necessarily interaction between these two notions of value. We now provide a negative result showing that external regularization cannot capture certain levels of sparsity.

**Proposition 1.** *Suppose $\alpha(\mathbf{x}) = 0$ for every $\mathbf{x}$ where $\xi(\mathbf{x}) \leq \theta$. Then, for any value of $\lambda > 0$, every maximizer of $\alpha_{ER}(\mathbf{x}; \lambda)$ will satisfy $\xi(\mathbf{x}) > \theta$, or will equal $\mathbf{x}^s$.*

This result is proved in Section S2.3 in the supplementary material. If the acquisition value is 0 whenever the sparsity penalty is below a certain level $\theta$, external regularization will not be able to recover any points with sparsity penalty below that level, other than the trivial point of maximum sparsity. Empirically, this manifests itself by the regularized acquisition choosing non-sparse points, or repeatedly sampling $\mathbf{x}^s$.

A setting where the acquisition value is 0 for all sparse points is easily encountered in practice when there is a trade-off between the objective function and sparsity, and we have sampled a point close to the (non-sparse) optimum. Consider the EI acquisition function with external regularization:

$$\alpha_{\text{EI-ER}}(\mathbf{x}; \lambda) = \mathbb{E}_f \left[(f(\mathbf{x}) - f(\mathbf{x}^*))_+\right] - \lambda\xi(\mathbf{x}). \tag{S3}$$

Once the GP is confident that sparse points have worse objective value than non-sparse points, sparse points will have acquisition value approximately 0, as their improvement is being evaluated with respect to a non-sparse incumbent best $\mathbf{x}^*$. By Proposition 1, sparse points will then not be selected by the regularized acquisition function, regardless of how $\lambda$ is tuned. Increasing $\lambda$ will change the maximum of the regularized acquisition function from a non-sparse point directly to the trivial solution of $\mathbf{x}^s$, skipping all levels of sparsity in between. There is nothing in (S3) to enable the acquisition function to select sparse points that improve over other points with a similar level of sparsity, which is necessary to fully explore the sparsity vs. objective trade-off.

### S2.2 Internal Regularization

The goal of the acquisition function in (2) is to maximize $g$, which can be made to have a sparse maximizer by appropriately setting $\lambda$. With internal regularization, EI becomes

$$\alpha_{\text{EI-IR}}(\mathbf{x}; \lambda) = \mathbb{E}_f \left[(g(\mathbf{x}) - g(\mathbf{x}^*))_+\right] = \mathbb{E}_f \left[(f(\mathbf{x}) - f(\mathbf{x}^*) - \lambda(\xi(\mathbf{x}) - \xi(\mathbf{x}^*)))_+\right] \tag{S4}$$

where $\mathbf{x}^*$ is now the incumbent-best of $g$, not of $f$. The difference between external and internal regularization depends on the acquisition function. It is easy to see that for the UCB acquisition of (S2), they are identical. For EI they are not, as seen by comparing (S3) and (S4). For EI, internal regularization avoids some of the issues of external regularization by incorporating sparsity directly into the assessment of improvement. In (S4), improvement is measured both in terms of increase of objective and increase in sparsity, and it is measured with respect to an incumbent best that has incorporated the sparsity penalty. However, internal regularization can also be incapable of recovering points at every level of sparsity, as we will show now. For this result, we are interested in the optimal objective value as a function of sparsity level:

$$h(\theta) = \max_{\mathbf{x} \in \mathcal{B}} f(\mathbf{x}) \text{ subject to } \xi(\mathbf{x}) = \theta. \tag{S5}$$

A trade-off between sparsity and objective would result in $h(\theta)$ increasing with $\theta$, though it need not be strictly increasing. We now give the negative result for internal regularization, which assumes that $\xi$ is continuous and bounded. See the supplementary material for more details.

**Proposition 2.** *For any $\theta$ in the interior of an interval where $h$ is strictly convex, there is no maximizer of (2) with $\xi(\mathbf{x}) = \theta$, for any $\lambda > 0$.*

This result shows that internal regularization can only hope to recover optimal points at all sparsity levels if $h$ is concave on its entire domain. This is a strong condition, one unlikely to hold for the types of functions typically of interest in BO, even with simple regularizers. Note that this result is independent of the choice of $\lambda$ and the acquisition function used. If the desired level of sparsity happens to lie within a region where $h$ is strictly convex, internal regularization can be expected to fail to find the optimum. Fig. S1 in the supplement shows an illustration of this result, in a problem where $h$ has a region of strict convexity.

We will see in the empirical results that internal regularization performs better than external regularization, though, consistent with Proposition 2, can fail to cover the entire objective vs. sparsity trade-off. For EI with internal regularization, the expectation in (S4) no longer has the closed form of regular EI, however this acquisition function can easily be computed and optimized using Monte Carlo methods [41, 2]. In this paper we focus on EI, but both forms of regularization can be applied to any acquisition function, including entropy search methods. In entropy search, the acquisition function evaluates points according to their information gain with respect to the current belief about the location or function value of the optimum. The information gain will thus depend on the level of sparsity in a similar way as with EI, and so external vs. internal regularization have similar considerations.

### S2.3 Proofs

Here we provide the proofs of Propositions 1 and 2, as well as an illustration of the result of Proposition 2.

*Proof of Proposition 1.* Suppose $\mathbf{x}^\dagger \in \arg\max \alpha^{\text{ext}}(\mathbf{x}; \lambda)$ and $\xi(\mathbf{x}^\dagger) \leq \theta$. Then, $\alpha(\mathbf{x}^\dagger) = 0$, so $\alpha^{\text{ext}}(\mathbf{x}^\dagger; \lambda) = -\lambda\xi(\mathbf{x}^\dagger)$.

By $\mathbf{x}^\dagger$ being a maximizer of $\alpha^{\text{ext}}$ we must have

$$-\lambda\xi(\mathbf{x}^\dagger) = \alpha^{\text{ext}}(\mathbf{x}^\dagger; \lambda) \geq \alpha^{\text{ext}}(\mathbf{x}^s; \lambda) = -\lambda\xi(\mathbf{x}^s).$$

Thus $\xi(\mathbf{x}^\dagger) \leq \xi(\mathbf{x}^s)$. Because $\mathbf{x}^s$ is a strict global minimum, we have then that $\mathbf{x}^\dagger = \mathbf{x}^s$. □

We assume $\xi$ is continuous and bounded, which implies $h$ is continuous and bounded:

**Assumption 1.** *$\xi$ is continuous on $\mathcal{B}$, and has minimum value $\xi(\mathbf{x}^s) = s_l$ and maximum value $s_u$.*

**Proposition 3.** *$h$ is continuous and bounded on the domain $[s_l, s_u]$.*

*Sketch of Proof.* This result falls from the continuity and boundedness of $f$, and by applying the intermediate value theorem to $\xi$. □

*Proof of Proposition 2.* Suppose $h$ is strictly convex over the interval $[\theta_l, \theta_u]$. For the sake of contradiction, assume that there exists a $\theta_\dagger \in (\theta_l, \theta_u)$ and an $\mathbf{x}^\dagger$ such that $\mathbf{x}^\dagger \in \arg\max g(\mathbf{x}; \lambda)$ and $\xi(\mathbf{x}^\dagger) = \theta_\dagger$.

It is clear that $\mathbf{x}^\dagger \in \arg\max f(\mathbf{x})$ subject to $\xi(\mathbf{x}) = \theta_\dagger$, otherwise the point with strictly larger $f$ and equal $\xi$ value would have a higher value for $g$, and $\mathbf{x}^\dagger$ could not be optimal for $g$. Thus, $f(\mathbf{x}^\dagger) = h(\theta_\dagger)$.

We can express $\theta_\dagger = t\theta_l + (1-t)\theta_u$ for some $t \in (0, 1)$. By strict convexity of $h$ on this interval, we have that

$$h(\theta_\dagger) < th(\theta_l) + (1-t)h(\theta_u). \tag{S6}$$

Take $\mathbf{x}^u \in \arg\max f(\mathbf{x})$ subject to $\xi(\mathbf{x}) = \theta_u$, and $\mathbf{x}^l \in \arg\max f(\mathbf{x})$ subject to $\xi(\mathbf{x}) = \theta_l$. These are the points in $\mathcal{B}$ corresponding to $h(\theta_l)$ and $h(\theta_u)$. The optimality of $\mathbf{x}^\dagger$ implies that $g(\mathbf{x}^\dagger; \lambda) \geq g(\mathbf{x}^u; \lambda)$ and $g(\mathbf{x}^\dagger; \lambda) \geq g(\mathbf{x}^l; \lambda)$. Thus,

$$\begin{aligned} g(\mathbf{x}^\dagger; \lambda) &\geq tg(\mathbf{x}^l; \lambda) + (1-t)g(\mathbf{x}^u; \lambda) \\ h(\theta_\dagger) - \lambda\theta_\dagger &\geq th(\theta_l) - t\lambda\theta_l + (1-t)h(\theta_u) - (1-t)\lambda\theta_u \\ h(\theta_\dagger) &\geq th(\theta_l) + (1-t)h(\theta_u), \end{aligned} \tag{S7}$$

using $\theta_\dagger = t\theta_l + (1-t)\theta_u$. The result in (S7) contradicts the convexity in (S6), and so $\mathbf{x}^\dagger$ cannot be optimal for $g$. □

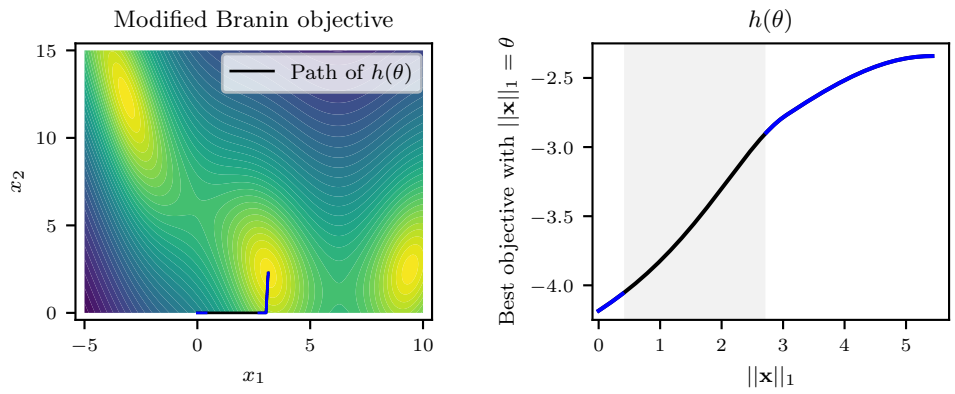


Figure S1: An illustration of the internal regularization result in Proposition 2. (Left) The objective $f$ is a modified Branin function. The sparsity penalty $\xi$ is the $L_1$ norm. (Right) The optimal objective vs. sparsity trade-off, $h(\theta)$, shows the best-achievable objective value for any specified value of $L_1$ norm. The shaded region is an interval where $h$ is strictly convex. By Proposition 2, the regularized function in (2) has no maximizers with $L_1$ norm in that range, for any value of $\lambda$.

Fig. S1 shows an illustration of the result of Proposition 2 on a log-transformed version of the classic Branin problem, where $f(x_1, x_2) = -\log(10 + \text{Branin}(x_1, x_2))$, and we are using a traditional $L_1$ regularization penalty, $\xi(x_1, x_2) = |x_1| + |x_2|$. The right panel shows $h(\theta)$, from (S5), as it traces the trade-off from the minimum of $\xi$ to the maximum of $f$. There is a wide interval of $L_1$-norm values in the middle, $0.4$ to $2.7$, where $h(\theta)$ is strictly convex. By Proposition 2, there is no value of $\lambda$ under which the maximizer of (2) has $L_1$ norm in that range. That range of sparsity levels thus cannot be reached by maximizing the regularized function $g$.

## S3 Relationship between ParEGO and Internal Regularization

**Remark 1.** *Internal regularization can be viewed as a linear scalarization of the two objectives $f$ and $-\xi$, with $\lambda$ the weight. Linear scalarizations are commonly used in MOO [26]. In this section, we discuss the connection between internal regularization and the ParEGO method for multi-objective BO.*

As described in Section S1, ParEGO applies the EI acquisition function to a random scalarization of multiple objectives. With internal regularization, random sampling of $\lambda$ for each acquisition optimization produces a ParEGO-style strategy for sparse BO, that differs only in the form of the scalarization.

The inability of linear scalarizations to capture the entire Pareto front, seen in Proposition 2, is a well-known failure mode for MOO. This result has inspired a large number of alternative scalarizations [6]. ParEGO avoids the issue by replacing the linear scalarization with an augmented Chebyshev scalarization [4]. When applied to the context of sparse regularization, this means maximizing

$$T(\mathbf{x}; \lambda) = C(f(\mathbf{x}) - \lambda\xi(\mathbf{x})) - \max(f^* - f(\mathbf{x}), \lambda(\xi(\mathbf{x}) - \xi(\mathbf{x}^s)),$$

where $f^*$ is an estimate for the maximum of $f$ and $C$ is a constant, usually set to $0.05$. Unlike $g$ in (2), maximizers of $T$ span the entire objective vs. sparsity trade-off [23]. Using EI to optimize this regularized function with randomly sampled values of $\lambda$ is equivalent to applying ParEGO to the objective and the (negative) sparsity penalty.

## S4 Optimization with $L_0$ Sparsity

### S4.1 Homotopy continuation

In this section we provide some additional details for the homotopy continuation described in Sec. 2. For computational reasons, we use a sequence of 30 $a$'s starting from $a_{\text{start}} = 10^{-0.5}$ and ending at $10^{-3}$ that is linearly spaced on a log-scale. First, we optimize the acquisition function using

L-BFGS-B from 20 different starting points to obtain 20 local optima of $H(x, a_{\text{start}})$. We then increment the value of $a$ and use L-BFGS-B to re-optimize the homotopy starting from each of the previously found 20 local optima. This process is continued until we reach $a = 0$ which is the acquisition function corresponding to the true $L_0$ norm. Note that this procedure traces 20 curves $c(a) \in \arg\min_x H(x, a)$ from $a = a_{\text{start}}$ to $a = 0$ and that this curve is of finite length under the assumption that the domain is compact. These curves are potentially different as the acquisition function may be non-convex and have multiple local optima. Finally, we choose the candidate as the point that achieves the best acquisition function value.

We use $a_{\text{start}} = 10^{-0.5}$ as it strikes a balance between being large enough to find initial points with non-zero acquisition function values, and being small enough to discover points that are almost sparse. To better understand this choice note that $\max_{x,z\in[0,1]} \left|\varphi'_{10^{-0.5}}(\mathbf{x} - \mathbf{z})\right| \approx 0.067$ while, e.g., $\max_{x,z\in[0,1]} |\varphi'_{0.1}(\mathbf{x} - \mathbf{z})| \approx 2 \times 10^{-20}$ which shows that $0.1$ may be too small to serve as $a_{\text{start}}$. We also investigate this choice in an ablation study in Appendix S7.2 and find that the performance of SEBO-$L_0$ is not sensitive to the choice of $a_{\text{start}}$ as long as the value is not too small.

### S4.2 SEBO algorithm

The SEBO-$L_0$ method is described in Algorithm 1. We start with an initial space-filling experiment design. In each iteration step, we fit a SAAS GP model and optimize the acquisition function to find the next point to evaluate, as shown at line 1. When optimizing the acquisition function, homotopy continuation is used to handle the discontinuous $L_0$ norm. This part is shown on line 11.

**Algorithm 1** Sparsity Exploring Bayesian Optimization with $L_0$ norm (SEBO-$L_0$)

1: **procedure** SEBO-$L_0$ ▷ Outer loop of BO
2: Place a Gaussian Process prior on $f$
3: Observe $f$ at $n_0$ quasi-random initial points and get the initial dataset $\mathcal{D}_{n_0}$
4: **for** $n \leftarrow n_0 + 1$ to $N$ **do**
5: Update the posterior probability distribution on $f$ using observed data $\mathcal{D}_{n-1}$
6: Select the next point $\mathbf{x}_n \leftarrow$ OPTIMIZE-HOMOTOPY($\hat{f}_n$)
7: Evaluate $\mathbf{x}_n$: $\mathcal{D}_n \leftarrow \{\mathcal{D}_{n-1}, (\mathbf{x}_n, f(\mathbf{x}_n))\}$
8: **end for**
9: **return** The best point
10: **end procedure**

11: **procedure** OPTIMIZE-HOMOTOPY($\hat{f}$) ▷ Optimize SEBO-$L_0$ acquisition function
12: Define a homotopy $H(\mathbf{x}, a)$ using the posterior on $f$
13: Initialize a candidate pool $\mathcal{X}_a \leftarrow \{\}$
14: **for** $a \leftarrow a_{\text{start}}$ to $a_{\text{end}}$ **do**
15: $\mathbf{x_a} \leftarrow$ maximize $H(\mathbf{x}, a)$ based on the best points in $\mathcal{X}_a$
16: $\mathcal{X}_a \leftarrow \{\mathcal{X}_a, \mathbf{x}_a\}$
17: **end for**
18: **return** $\mathbf{x_a}$
19: **end procedure**

## S5 Additional Experimental Studies

### S5.1 Ranking sourcing system simulation

In the sourcing simulation experiment in Section 3, the recommender sourcing system has 25 content sources and 1000 possible pieces of content (i.e., *items*) for retrieval. We consider a 25-dimensional retrieval policy $\mathbf{x}$ over the integer domain $[0, 50]^{25}$. We take inspiration from the Latent Dirichlet Allocation (LDA) model [3] in defining a generative probabilistic model of items recommended by each source. We assume there are 8 latent topics and that each item can be represented as a mixture over topics. Each source contains a mixture over a set of topics, and particular items will be more likely to be recommended by topically related sources. Such topical overlaps can create redundancy

of recommendations across sources. Retrieving more items from additional sources comes at a cost making sparse retrieval policies preferred.

Before describing the simulation in pseudo-code, we need the following definitions:

- $T$ is the number of latent topics.
- $K$ is the number of distinct items.
- $S$ is the number of content sources.
- $\theta_s \in \Delta^T$ is the topic distribution for source $s$, where $\Delta^T$ denotes the $T$-dimensional simplex. $\{\theta_s\}_{s=1}^S$ follow a Dirichlet distribution, i.e., $\theta_s \sim \text{Dir}(\alpha)$ where $\alpha = 0.2$.
- $\phi_i \in \Delta^K$ is the item distribution for each topic $i$, where $\Delta^K$ denotes the $K$-dimensional simplex. $\{\phi_i\}_{i=1}^T$ also follow a Dirichlet distribution, i.e., $\phi_i \sim \text{Dir}(\beta)$ where $\beta = 0.5$.
- $z_{s,k}$ is the topic assignment for item $k$ in source $s$ and follows multinomial distribution: $z_{s,k} \sim \text{Multi}(\theta_s)$
- $w_{s,k}$ is the indicator of item $k$ is retrieved from source $s$ and follows multinomial distribution: $w_{s,k} \sim \text{Multi}(\phi_{z_{s,k}})$.
- $Q_i$ is the relevance score of each topic $i$ and is sampled from a log-Normal distribution with mean $0.25$ and standard deviation $1.5$.
- $m_k$ is the relevance score of each item $k$, which is derived as the weighted average across topic scores based on the item distribution over $8$ latent topics, i.e., $m_k = \sum_{i=1}^T \phi_{i,k} Q_i$.
- $c_s$ is the infrastructure cost per fetched item for source $s$. The cost $c_s$ is assumed to be positively correlated with source relevance score $q_s = \sum_{i=1}^T \theta_{s,i} Q_i$ and follows a Gaussian distribution with mean $\frac{q_s}{2\sum_{s=1}^S q_s}$ and standard deviation of $0.1$.

To simulate the retrieval of one item from the source $s$, we sample a topic for an item $k$ from the multinomial $\text{Multi}(\theta_s)$, i.e., $z_{s,k} \sim \text{Multi}(\theta_s)$, and sample an item $w_{s,k} \sim \text{Multi}(\phi_{z_{s,k}})$ where $w_{s,k}$ indicates item $k$ being retrieved from source $s$. Given the sourcing policy $\mathbf{x} \in \mathbb{R}^S$, we execute the above sampling $\mathbf{x_s}$ times for each source $s$ as described at lines 1 in Algorithm 2, and then compute the quality score given a list of retrieved items.

The overall content relevance score is the sum of the content relevance scores after de-duplicating the retrieved content. The infrastructure load is a sum of products of a number of retrievals and the cost per fetched item $c_s$ for each source, in which $c_s$ varies across sources and positively correlates with the source relevance score. This setup is based on the real-world observation that sources providing higher relevance content are generally more computationally expensive. The objective in the benchmark experiments is a weighted sum of overall content relevance and negative infrastructure load. In the experiment, we repeat this simulation (at line 10) $1000$ times for a given policy and compute the mean and standard error of the objective values, which we refer to as the *quality score* in the main text.

### S5.2 Additional Experiments

We evaluate EI-IR, EI-ER and SEBO on two synthetic and three real-world problems. We show the results using $L_0$ regularization for most problems except for the last problem, where the group lasso is used to demonstrate that the methods can be applied to recover different forms of sparsity, such as group sparsity. In addition, we provide an ablation study that demonstrates the importance of using $L_0$ regularization by comparing it to $L_1$ regularization. We show in an ablation study that the homotopy continuation approach from Section 2 is crucial for effective $L_0$ regularization.

**Experimental setup:** Our experiments all have high-dimensional parameter spaces, so we use the SAAS model when optimizing with ER, IR, and SEBO. We compare performance to quasi-random search (Sobol), BO with a standard ARD Matérn-$5/2$ kernel and the EI acquisition function (GPEI), and SAASBO. For the SAAS model, we use the same hyperparameters as suggested by Eriksson and Jankowiak [12] and use the No-U-Turn (NUTS) sampler for model inference. The acquisition function is computed by averaging over the MCMC samples. We always scale the domain to be the unit hypercube $[0, 1]^D$ and standardize the objective to have mean $0$ and variance $1$ before fitting

**Algorithm 2** Recsys Simulation

**procedure** ITEM-RETRIEVAL($x_s$)
  $\vec{n}_s \leftarrow \overrightarrow{0} \in \mathbb{R}^K$ ▷ number of retrievals for $K$ distinct items
  **for** $n \leftarrow 1$ to $x_s$ **do** ▷ retrieve $x_s$ items
    Sample a topic for an item $k$ in source $s$ i.e. $z_{s,k} \sim$ Multi($\theta_s$)
    Sample an item $w_{s,k} \sim$ Multi($\phi_{z_{s,k}}$)
    $\vec{n}_s \leftarrow \vec{n}_s + \vec{w}_s$
  **end for**
  **return** $\vec{n}_s$
**end procedure**

**procedure** SOURCING($\mathbf{x}$)
  $\vec{n} \leftarrow \overrightarrow{0} \in \mathbb{R}^K$ ▷ number of retrievals for $K$ distinct items
  **for** $s \leftarrow 1$ to $S$ **do** ▷ retrieve items for each source $s$
    $\vec{n}_s \leftarrow$ ITEM-RETRIEVAL($x_s$)
    $\vec{n} \leftarrow \{\vec{n} + \vec{n}_s\}$
  **end for**
  Compute relevance score RS $= \sum_{k=1}^{K} \mathbb{1}(n_k > 0) m_k$ and infrastructure cost $C = \sum_{s=1}^{S} c_s \times x_s$
  **return** quality score $Q =$ RS $- 0.6 \times C$
**end procedure**

the GP model. For the homotopy continuation approach described in Sec. 2, we discretize the range of $a$ to use 30 values of starting from $a_{\text{start}} = 10^{-0.5}$, see the appendix for more details. Fig. S11 shows that SEBO is not sensitive to the choice of $a_{\text{start}}$. We use a deterministic model for sparsity when using it as an objective. The figures show the mean results across replications (10 replications for the adaptive bitrate simulation (ABR) problem and 20 for all other experiments), and the error bars correspond to 2 standard errors. All experiments were run on a Tesla V100 SXM2 GPU (16GB RAM). Methods, benchmarks, code for replicating this work will be available upon publication.

**Evaluation plots:** We evaluate optimization performance in terms of the trade-off between the objective and sparsity. To compare the trade-offs, we show the resulting Pareto frontier by treating sparsity as a separate objective, e.g., Fig. S2 (right) and Fig. S3. In particular, for each level of sparsity (active dimensions), we will plot the best value found using *at most* that number of non-sparse components. We also show hypervolume traces in the Appendix S5.3. In cases where a method is unable to find at least one configuration for a given level of sparsity we assign replications an imputed function value corresponding to the worst label shown on the y-axis. In addition, for the synthetic problems where the true active dimensions and the optima are known, we plot simple regret for a fixed level of sparsity, e.g., in Fig. S2 (left, middle).

**Synthetic functions:** We first consider two synthetic problems where the level of sparsity is known. We use the Branin and Hartmann6 functions embedded into a 50D space where 0 is considered sparse. We used 50 trials (evaluations) with 8 quasi-random initial points for Branin and 100 trials with 20 quasi-random initial points for Hartmann6. The results are shown in Fig. S2. The two leftmost plots show the optimization results by evaluating the objective only on observed points whose number of active (i.e., non-zero) parameters was less than or equal to the true effective dimension (2 for Branin and 6 for Hartmann6). We observe that SEBO-$L_0$ performed the best, followed by IR with $\lambda = 0.001$. This suggests IR may perform competitively if the regularization coefficient is chosen optimally. On the other hand, ER performed worse than SEBO and IR. Finally, methods with non-regularized acquisition functions (Sobol, GPEI, and SAASBO) failed to identify sparse configurations. Fig. S2 (right) visualizes the trade-off between the objective and sparsity, in which SEBO-$L_0$ yielded the best sparsity trade-offs.

**Ranking sourcing system simulation:** The sourcing component of a recommendation system is responsible for retrieving a collection of items that are sent to the ranking algorithm for scoring. Items are retrieved from multiple sources, for instance that may represent different aspects of the user interest taxonomy [40]. Querying for more items can potentially improve the quality of the

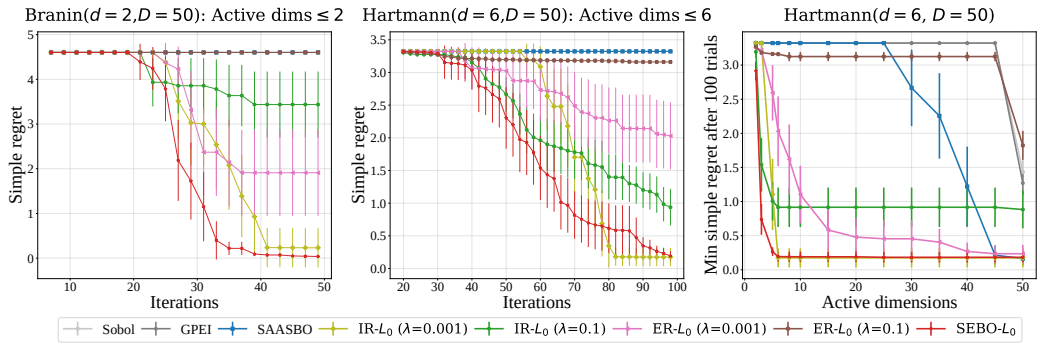


Figure S2: (Left) Simple regret for Branin embedded into a 50D space, considering only observations with at most 2 active (non-sparse) parameters. SEBO-$L_0$ performed the best followed by IR with $\lambda = 0.001$. (Middle) SEBO-$L_0$ and IR with $\lambda = 0.001$ performed the best for the Hartmann6 function embedded into a 50D space when considering only observations with at most 6 active parameters. (Right) The objective-sparsity trade-off after all 100 iterations on the Hartmann6 problem. Shown is the Pareto frontier between sparsity and simple regret after the evaluation budget has been exhausted. SEBO-$L_0$ is able to explore the trade-offs and is able to discover sparse configurations with fewer than 6 active parameters that are not found by the other methods.

recommendation system, but comes at the cost of increasing the infrastructure load. In addition, each source may require individual maintenance; thus, deprecating poor sources could reduce technical debt and maintenance costs of an entire recommendation system [30]. Our goal is thus to identify a retrieval policy that uses a minimal number of sources while still maximizing the ranking quality score, measured by a function of content relevance and infrastructure load.

We developed a simulation of a recommender sourcing system that simulates the quality and infrastructure load of recommendations produced by a particular sourcing policy. The sourcing system is modeled as a topic model, where each source has a different distribution over topics, and topics have different levels of relevance to the user. When two sources are (topically) similar to one another, they may obtain duplicate items, which will not improve recommendation quality.

We consider a 25-dimensional retrieval policy in which each parameter specifies the number of items retrieved from a particular source. Our desired sparsity is to set parameters to 0, i.e., turning off the source. See Sec. S5.1 for more details. We used 8 initial points and ran 100 trials for all the methods. Fig. S3 (Left) shows that SEBO-$L_0$ performed the best in optimizing the ranking quality score under different sparsity levels. Sobol and GPEI could not find sparse policies and obtained worse quality scores even with 25 active parameters. IR and SAASBO performed similarly, and ER with the larger regularization parameter $\lambda = 0.01$ achieved higher quality score with less than 10 active dimensions.

**SVM Machine learning hyperparameter tuning:** We consider the problem of doing joint feature selection and hyperparameter tuning for a support vector machine (SVM). We tuned the $C$, $\varepsilon$, and $\gamma$ hyperparameters of the SVM, jointly with separate scale factors in the continuous range $[0, 1]$ for each feature. We used 100 features from the CT slice UCI dataset [10] and the goal was to minimize the RMSE on the test set. This produces a 103D optimization problem where we shrink towards a scale factor of 0, as it effectively removes the feature from the dataset. We took $C \in [0.01, 1.0]$, $\varepsilon \in [0.01, 1.0]$, and $\gamma \in [0.001, 0.1]$, where the center of each interval was considered sparse as this is the default value in Sklearn. We optimized $C$, $\varepsilon$, $\gamma$ on a log-scale, and initialized all methods with 20 points and ran 100 evaluations. Fig. S3 (Middle) shows that SEBO-$L_0$ was best able to explore the trade-offs between sparsity and (negative) RMSE.

**Adaptive bitrate simulation:** Video streaming and real-time conferencing systems use adaptive bitrate (ABR) algorithms to balance video quality and uninterrupted playback. The goal is to maximize the quality of experience (QoE). The optimal policy for a particular ABR controller may depend on the network, for instance a stream with large fluctuations in bandwidth will benefit from different ABR parameters than a stream with stable bandwidth. This motivates the use of a contextual policy where ABR parameters are personalized by context variables such as country or network

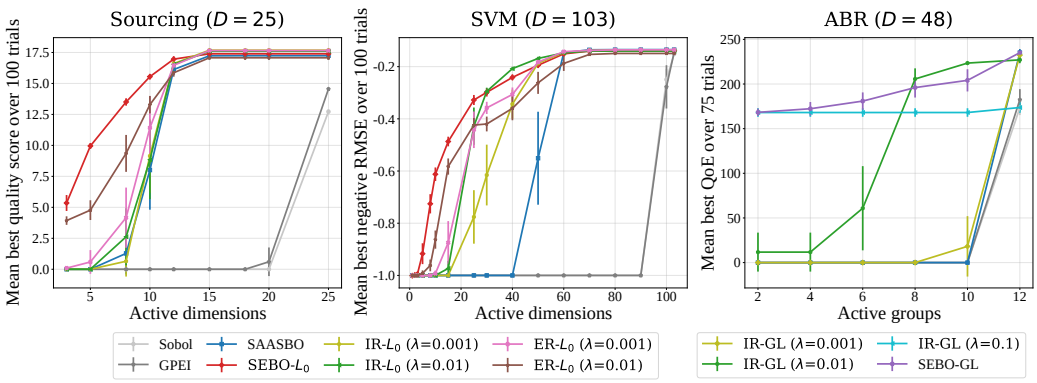


Figure S3: Objective-sparsity trade-offs after 100 (75 for ABR) trials for the three real-world problems. (Left) *Sourcing problem*: SEBO-$L_0$ regularization effectively explored all sparsity trade-offs. (Middle) *SVM problem*: ER with $\lambda = 0.01$ and IR with $\lambda = 0.01$ were able to explore parts of the Pareto frontier, however were dominated by SEBO-$L_0$. (Right) *ABR problem*: Similar behavior as in the SVM problem was seen here with a group lasso penalty.

type [16]. Various other systems and infrastructure applications commonly rely on tunable parameters which can benefit from contextualization.

We suppose that the system has already been optimized with a global non-contextual policy, $\pi_{\text{global}}$, that is used for all contexts. Our goal here is to use sparse BO to find the contextualized residuals $\Delta\pi_i$ for each individual context $i$, i.e., $\pi_i = \pi_{\text{global}} + \Delta\pi_i$. By regularizing the contextualized residuals $\Delta\pi_i$'s using the group lasso (GL) norm [43], we hope to find policies that require minimum alteration to the global policy $\pi_{\text{global}}$, in which the minimum number of contexts have parameters that deviate from the global optimum. This adds both simplicity and interpretability to the contextual policy, since we can interpret the policy by looking at the contextual residuals $\Delta\pi_i$.

Fig. S3 (Right) shows the results of applying our methods to the contextual ABR optimization problem from Feng et al. [16]. For this problem, we have 12 contexts and 4 parameters for each context resulting in a 48D optimization problem. We used 75 trials with 8 quasi-random initial points for all the methods. The group lasso penalty is defined by assigning parameters for each individual context to be within the same group. We observe that IR with a fixed $\lambda$ was able to explore trade-offs at certain sparsity levels and that stronger regularization (larger $\lambda$) resulted in finding configurations that were more sparse. SEBO-GL, on the other hand, automatically and efficiently explored the trade-off between sparsity and reward at all sparsity levels. All other baselines (Sobol, GPEI, SAASBO) failed to find any sparse configurations that achieve non-zero reward.

### S5.3 Hypervolume trace plots

We evaluate optimization performance by showing the average best obtained hypervolume across 20 replicates, with 95% confidence interval over 100 trials. The results are shown for the sourcing problem (left), the SVM problem (middle) and the Hartmann6 function embedded into a 50D (right) in Figure S4. It can be seen that SEBO-$L_0$ (red traces) outperforms all the other methods and achieved the best hypervolume value over 100 iterations. The IR and ER methods with well selected regularization parameter values can sometimes achieve competitive results and usually outperform the methods with non-regularized acquisition functions, e.g. SAASBO.

### S5.4 Benchmark with additional HDBO methods

We conduct evaluations of additional high-dimensional BO methods for the Hartmann6 function embedded in a 50D space, including trust region BO (TuRBO) by [13] and Random Embedding BO (REMBO) by [39]. The left plot in Figure S5 shows the trade-off between the objective and sparsity after all 100 iterations. Although SAASBO and TuRBO achieve good non-sparse solutions, they fail to obtain sparse solutions. REMBO does not obtain better sparse solution than SAASBO. In the right plot, we show the simple regret considering only observations with at most 35 active (non-sparse)

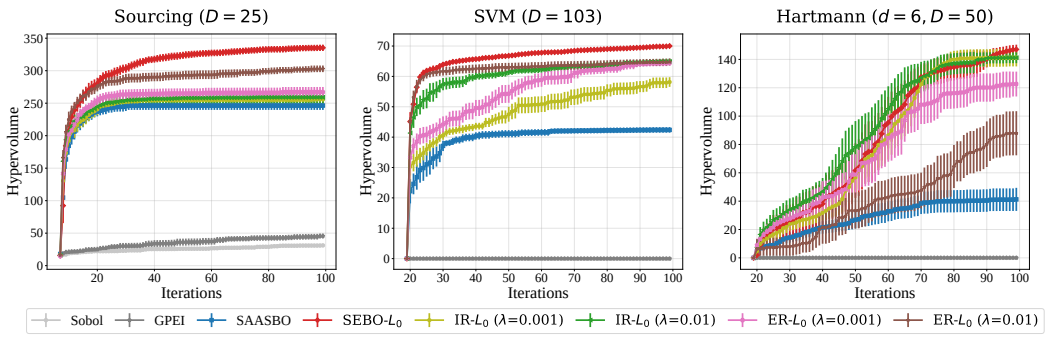


Figure S4: Hypervolume benchmark traces. (Left) Sourcing problem.(Middle) SVM problem. (Right) Hartmann6 function embedded into a 50D. The results are the average best hypervolume (with 95% confidence interval) obtained over 100 iterations across 20 replications. SEBO-$L_0$, shown in red, performs the best in all three problems.

parameters. SEBO-$L_0$ outperforms these high-dimensional BO since these methods do not encourage sparse solutions.

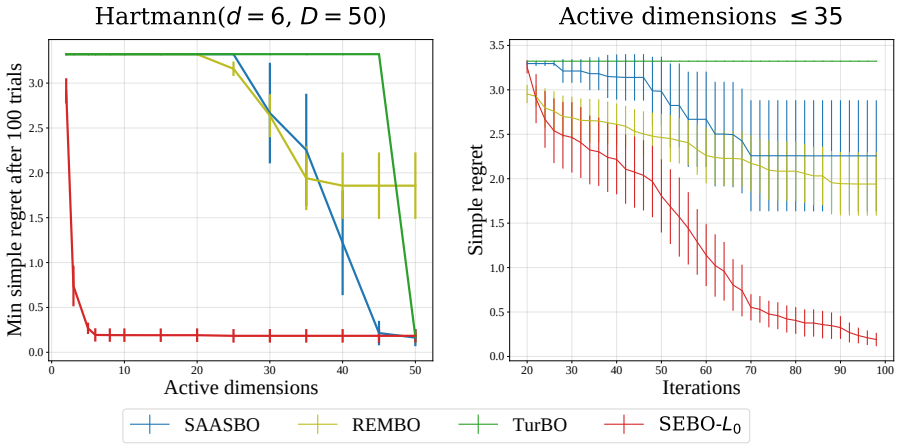


Figure S5: Results of additional high-dimensional BO methods for the Hartmann6 function embedded in a 50D space. (Left) The objective-sparsity trade-off after all 100 iterations. SAASBO and TuRBO, although obtaining competitive objective values with 50 active parameters, do not encourage sparse solutions. (Right) The simple regret for Hartmann6 function considering only observations with at most 35 active (non-sparse) parameters.

## S6 Ablation Studies

### S6.1 Benchmarks with $L_1$ regularization

Our proposed method can work together with different forms of sparsity. Here we show the results of EI-ER, EI-IR and SEBO using $L_0$ regularization for the Hartmann6 function embedded in a 50D space. As can been seen in Fig. S6, using $L_0$ leads to significant improvement over $L_1$ for all three methods.

### S6.2 Alation study on using homotopy continuation

We show in Fig. S7 by means of an ablation study the importance of using the homotopy continuation approach from Section 2 to target $L_0$ sparsity. We focus on SEBO as it consistently outperformed IR and ER. Using a fixed value of $a$ for the $L_0$ approximation performs poorly, particularly when $a$ is small, which is due to the acquisition function being zero almost everywhere and thus difficult

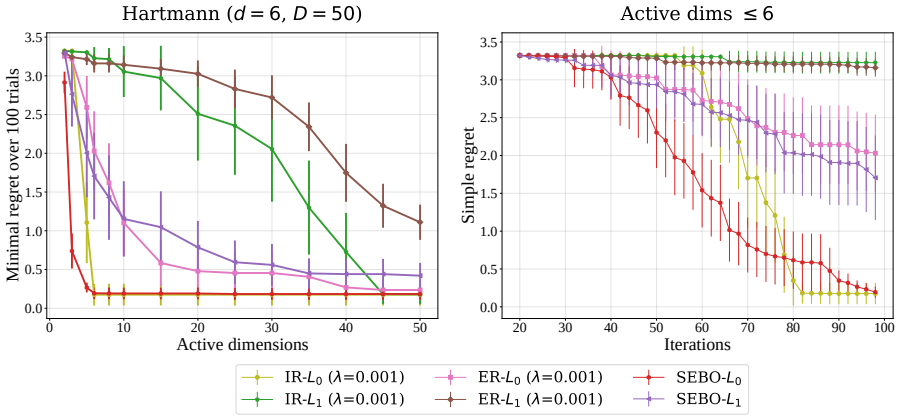


Figure S6: Results for the Hartmann6 function embedded in a 50D space. (Left) $L_0$ regularization outperforms $L_1$ regularization in exploring the objective-sparsity trade-offs for IR, ER and SEBO. (Right) $L_0$ regularization obtains better optimization performances considering only observations with at most 6 active (non-sparse) parameters.

to optimize. On the other hand, $a = 1$ results in a failure to discover sparse configurations and the resulting method performs similar to SAASBO (see Fig. S2).

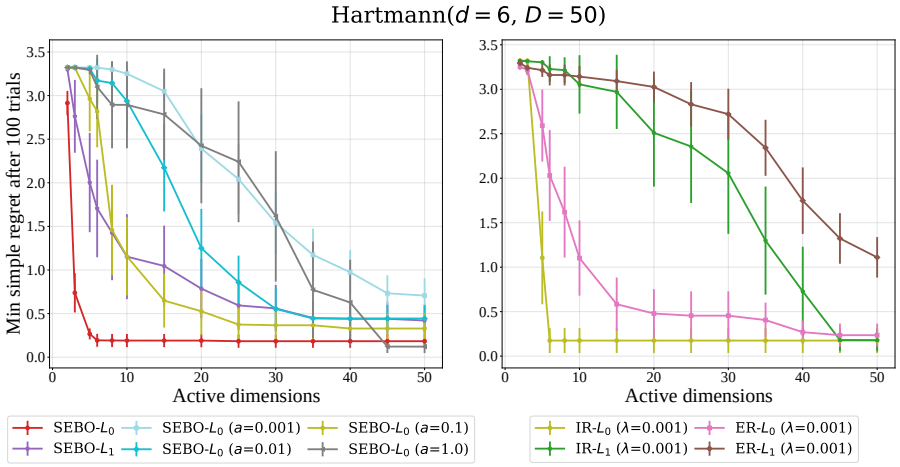


Figure S7: Ablation study on the Hartmann6 function embedded in a 50D space. (Upper) SEBO-$L_0$ works much better than SEBO-$L_1$ as it directly targets sparsity. Using a fixed value of $a$ performs poorly, confirming the importance of our homotopy continuation approach. (Lower) Working directly with $L_0$ regularization works drastically better than $L_1$ regularization for both IR and ER.

### S6.3 Ablation study on using SAAS

To illustrate the importance of using the SAAS model, we compare to using EI-IR-$L_1$ with a standard GP in Fig. S8. We observe that EI-IR-$L_1$ with a standard GP fails to discover non-trivial sparse configurations for all values of $\lambda$. This confirms that sparsity in the GP model is crucial for finding sparse configurations. This can also be observed by comparing performances of SAASBO and GPEI in Fig. S2 where there is a huge gap in terms of the best function value optimized even when looking at dense points (active dimensions $= 50$).

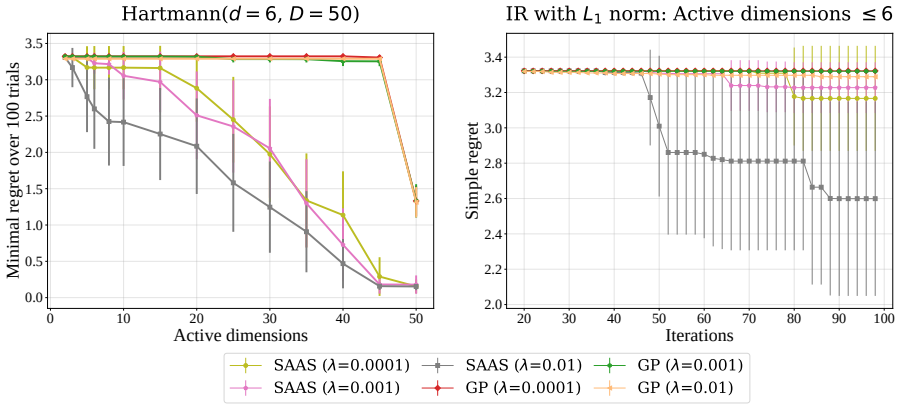


Figure S8: Results for the Hartmann6 function embedded in a 50D space. EI-IR-$L_1$ using the SAAS model significantly outperforms EI-IR-$L_1$ using a standard GP.

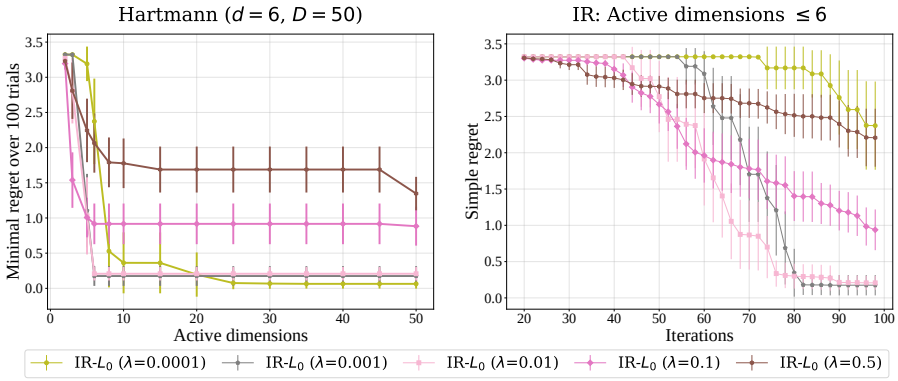


Figure S9: Results of EI-IR with different $\lambda$ values for Hartmann6 function embedded into a 50D space. (Left) The objective-sparsity trade-off after all 100 iterations. (Right) The simple regret considering only observations with at most 6 active (non-sparse) parameters.

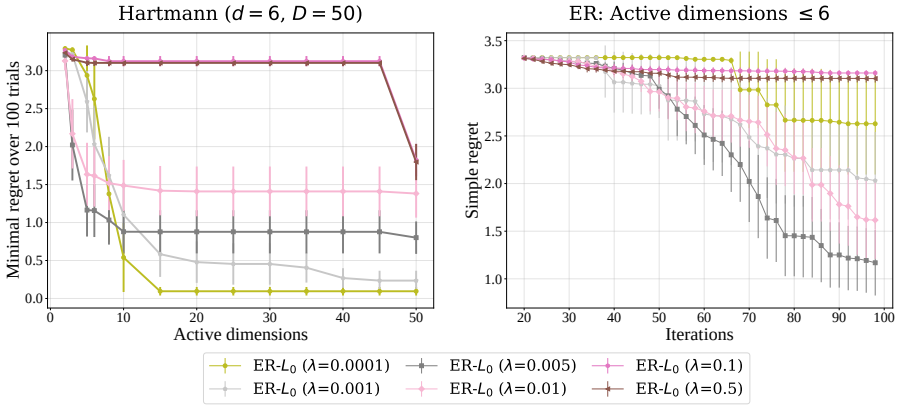


Figure S10: Results of EI-ER with different $\lambda$ values for Hartmann6 function embedded into a 50D space. (Left) The objective-sparsity trade-off after all 100 iterations. (Right) The simple regret considering only observations with at most 6 active (non-sparse) parameters.

# S7 Sensitivity Analysis

## S7.1 Sensitivity analysis of regularization parameter $\lambda$

We conduct a sensitivity analysis of regularization parameter $\lambda$ used by IR and ER by sweeping different values of $\lambda$ on the 50D Hartmann6 benchmark. The results are given in Fig. S9 and Fig. S10. We observe that we are able to control the sparsity level by appropriately choosing $\lambda$. In general, larger $\lambda$ implies stronger regularization and results in finding configurations with a higher level of sparsity. When $\lambda$ increases above a certain point, the regularization becomes too strong and fails to help find high-quality sparse points.

By comparing results of IR and ER for different $\lambda$ values, we note that IR is able to achieve effective optimization performance over a wider range of $\lambda$'s while ER is more sensitive to the value of $\lambda$. This validates the discussion about ER in Section S2.1 that ER is not as effective as IR due to ER's inability to select a new sparse point that improves over sparse points from previous iterates if the new sparse point does not improve on the dense points that are already observed.

## S7.2 Sensitivity analysis of $a_{\text{start}}$ in SEBO-$L_0$ optimization

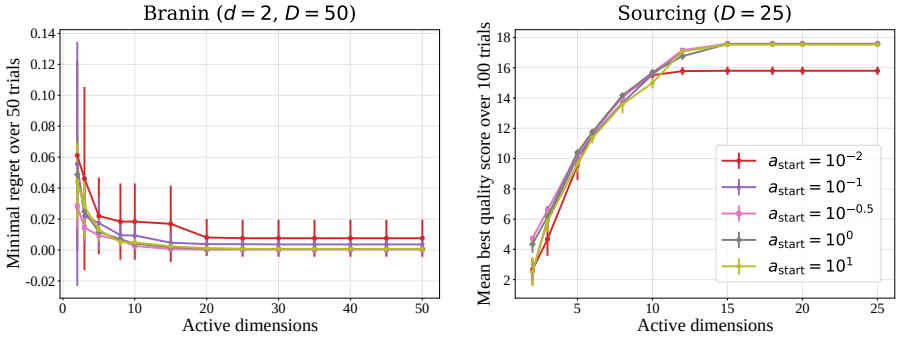


Figure S11: Ablation study of $a_{\text{start}}$ in SEBO-$L_0$. (Left). Results of Branin $(d = 2, D = 50)$. (Right). Results of Sourcing $(D = 25)$. There is no statistically significant difference between using different $a_{\text{start}}$ except for the extremely small $a_{\text{start}}$ $(= 10^{-2})$. This shows the robustness of having a default $a_{\text{start}}$ for optimizing SEBO-$L_0$ acquisition function.

The value of $a_{\text{start}}$ is set to be $10^{-0.5}$ for all the experiments. To better understand the robustness of this choice we conducted an ablation study on the Branin$(d = 2, D = 50)$ and Sourcing $(D = 25)$ problems considered in Section 3. The results in Figure S11 show that there is no statistically significant difference between using $10^{-1}$, $10^{-0.5}$, $10^{0}$ and $10^{1}$ as the value of $a_{\text{start}}$. However, using a value of $10^{-2}$ leads to a clear drop in performance as this starting value is too small to optimize the acquisition function.

# S8 Interpretation of Sparse Solutions

We examine what active dimensions are selected in the recommender sourcing system problem to understand the obtained sparse solutions. For SEBO-$L_0$ results across 20 replications, we obtain the optimal 25-dimensional retrieval policy and also compute the average of retrievals per source at each sparsity level. For each source, we compute a source quality scores based on the simulation setup stated in S5.1. Each source contains a mixture over a set of topics with source relevance score being $q_s$ and the infrastructure cost per fetched item being $c_s$. With this, we define and compute the source quality score as $q_s - 4 \times c_s$. Note the score is computed for each source in order to interpret the obtained solutions and differ from the quality score used in the optimization.

In Figure S12, the left heatmap visualizes the optimal policy at different sparsity levels across 20 replications and the middle one visualizes the average retrieval policy values. Each column corresponds to one source and is sorted based on source quality score in an ascending order (from left to right); each row represents the sparsity level (number of active dimensions). The color indicates

the parameter values. As it can be seen, sources with low quality scores are turned off (zero query) and sources with higher scores have higher number of retrievals even with smaller active dimensions. This indicates that the sparse policy obtained from SEBO identifies the most effective sources at each sparsity level. The right plot in Figure S12 shows the relationship between number of items retrieved from each source and source quality score with 5 active parameters. Each dot represents a source. The curve is a fitted spline to visualize the relationship. From both plots we can see that more items are retrieved from higher quality sources, while the number of items from lower quality sources are driven to zero.

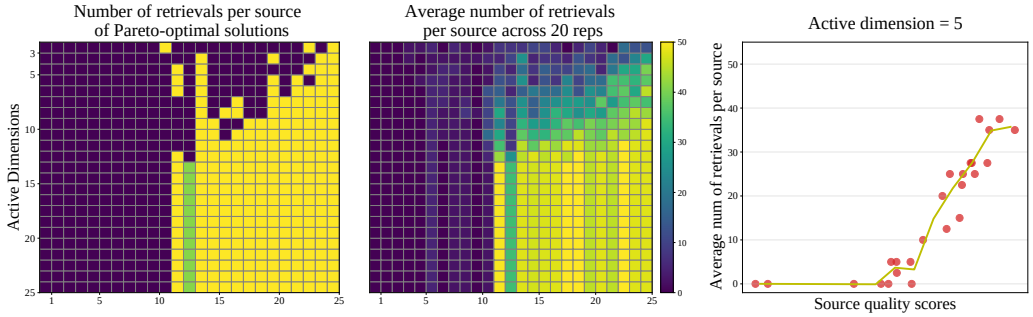


Figure S12: (Left). The heatmap of optimal retrieval policy at different sparsity levels. (Mid) The heatmap of average retrieval policy values at different sparsity levels. (Right) The scatter plot between average retrieval policy values with 5 active parameters and source quality score. We can see that more items are retrieved from higher quality sources, while the number of items from lower quality sources are driven to zero to achieve sparsity.

## S9 Code and Implementations

The GPEI, SAASBO and EHVI used in SEBO were implemented using BoTorch, a framework for BO in PyTorch [2] and are available in Ax `https://github.com/facebook/Ax`. The code is licensed under the MIT License. The SVM hyperparameter tuning experiment uses the SVM implementation in Sklearn and the CT slice dataset in the UCI machine learning repository [10]. The Adaptive bitrate simulation experiment is available in the Contextual BO open source code `https://github.com/facebookresearch/ContextualBO`, licensed under the MIT License.