# DisCoMBO: Steering Expert-in-the-Loop Black-Box Optimization via Distributional Conformance

**Jonas Seng***
Computer Science Department
TU Darmstadt
jonas.seng@tu-darmstadt.de

**Bennet Wittelsbach***
Computer Science Department
TU Darmstadt

**Kristian Kersting**
TU Darmstadt, hessian.AI, DFKI

## Abstract

Sequential Model-Based Optimization (SMBO) traditionally relies on Bayesian or ensembling surrogates for uncertainty quantification. While historically treated as fully data-driven, SMBO increasingly integrates external domain expertise to accelerate discovery. To overcome the opaque guidance and diminished integration fidelity of standard acquisition re-weighting, Probabilistic Circuits (PCs) have emerged as a generative surrogate alternative, enabling direct knowledge injection via conditional sampling. However, these generative routines lack the formal exploration-exploitation semantics required for rigorous optimization. We introduce the **Dis**tributional **Co**nformance Score (DisCo), a novel metric that unifies the flexibility and efficiency of PCs with a formal uncertainty framework. DisCo provides a bounded, $[0, 1]$-normalized measure of model "surprise" that (1) recovers properties comparable to kernel-based uncertainty known from, e.g., Gaussian Processes, while maintaining linear-time inference, and (2) enables accurate assessment of conformance of external knowledge w.r.t. model evidence. We then present DisCoMBO, a framework leveraging these properties for robust, knowledge-aware optimization. We prove that DisCoMBO is a zero-regret algorithm and demonstrate its effectiveness across diverse benchmarks from AutoML, material optimization, and wind park optimization.

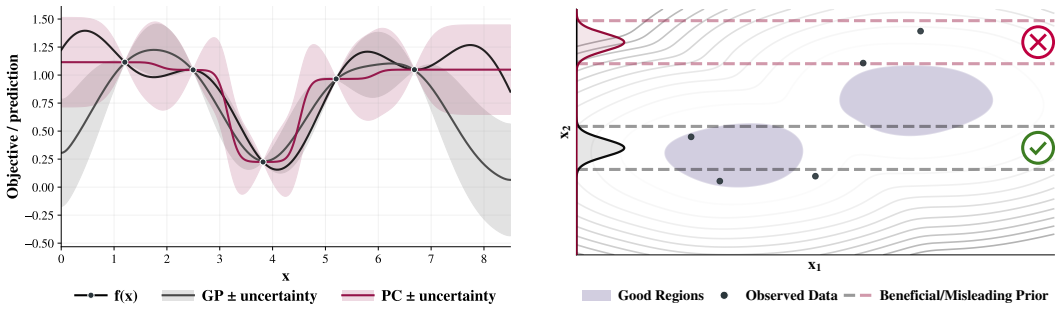
## 1 Introduction

Black-Box Optimization (BBO) is a cornerstone of modern scientific discovery and serves as an optimization engine throughout various disciplines and industries. Examples are material discovery [Kusne et al., 2020], wind park layout optimization [Ju et al., 2019], and hyperparameter tuning in machine learning [Schede et al., 2022]. Given the high cost of evaluating these objective functions, Sequential Model-Based Optimization (SMBO) has emerged as the dominant paradigm. By alternating between learning a surrogate model and acquiring the next candidate configuration, SMBO seeks to find the global optimum while navigating the search space with maximum sample efficiency.

Recently, different fields called for enabling the integration of external knowledge, e.g., from humans or Large Language Models (LLMs). Examples of such paradigms include human-centered AutoML [Wang et al., 2019, Park et al., 2021, Lindauer et al., 2024] and optimization in the physical sciences [Kusne et al., 2020] to increase sample efficiency, build user trust, and prevent optimizers from exploring implausible solutions. While traditional frameworks like $\pi$BO [Hvarfner et al.,

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).




(a) (b)

Figure 1: **DisCo provides uncertainty estimates for PC-based SMBO and allows for accurate conformance assessment of external knowledge – both with linear inference time.** (a) Given a learned PC from observations, DisCo (—) provides uncertainty estimates with **linear inference cost** based on how well a given point $\mathbf{x}$ conforms with the distribution represented by the PC, resulting in uncertainty quantification akin to Gaussian Processes (GPs; —). (b) DisCo can assess whether externally provided knowledge conforms to the PC's expectation for improving upon $y^*$, serving as a scale-independent basis for acceptance or rejection of given knowledge.

2022] and DynaBO [Fehring et al., 2026] integrate knowledge in the form of priors via acquisition re-weighting, this indirect mediation can result in "blurred“ guidance. Seng et al. [2025] introduced Probabilistic Circuits (PCs) as a novel surrogate in the SMBO landscape, enabling highly targeted knowledge integration by conditioning candidate generation on externally provided priors, thereby circumventing opaque reweighting logic. However, two fundamental challenges hinder the broader adoption of PCs in (expert-guided) SMBO: (1) a lack of native epistemic uncertainty estimates, which forces a reliance on pure likelihood sampling to obtain new candidates and precludes formal uncertainty-guided exploration; and (2) the absence of a robust and accurate metric for assessing whether external guidance contradicts model evidence, acting as a safeguard against misleading priors. Existing methods for assessing conformance of priors w.r.t. model evidence either rely solely on heuristics, such as relative acquisition changes [Fehring et al., 2026], or introduce structural complexity via auxiliary preference models [Xu et al., 2024]. To address the limitations of PC surrogates and existing safeguard mechanisms, we introduce the **Dis**tributional **Co**nformance Score (DisCo). While fundamentally a metric for evaluating the conformance of a sample against a learned circuit distribution, DisCo naturally serves as an uncertainty proxy under the premise that low conformance directly implies high epistemic uncertainty. We theoretically demonstrate that under specific PC parameterizations, this conformance-based proxy is rank-preserving w.r.t. kernel-like uncertainty estimates, maintaining the utility of traditional exploration-exploitation strategies while preserving the linear-time inference characteristic of PCs (see Fig. 1).

Building on this foundation, we present DisCoMBO (**DisCo**-informed **M**odel-**B**ased **O**ptimization), a framework that operationalizes DisCo to unify the tractability and mixed-variable flexibility of PCs with rigorous decision-theoretic semantics. By leveraging this conformance score, DisCoMBO simultaneously establishes the robust exploration mechanism required for BBO and provides a scale-independent safeguard to detect external priors that contradict empirical model evidence. The resulting procedure bridges the gap between flexible generative modeling and traditional optimization rigor, all without sacrificing the targeted integration capabilities of the generative paradigm.

**Contributions:** **(1)** The Distributional Conformance (DisCo) score, a novel and mathematically grounded score providing an uncertainty proxy for PCs at linear inference cost. **(2)** The DisCoMBO framework, a novel SMBO algorithm that utilizes DisCo to enable principled exploration-exploitation trade-offs. The framework facilitates targeted knowledge integration via generative sampling and provides a robust signal for assessing the conformance of external guidance with empirical data. Theoretical analysis establishes DisCoMBO as a zero-regret algorithm. **(3)** Extensive empirical evaluation across AutoML, material discovery, and wind park optimization benchmarks, demonstrating that DisCoMBO significantly outperforms fully generative PC-based SMBO and achieves performance competitive with state-of-the-art BO baselines in both unassisted and informed settings.

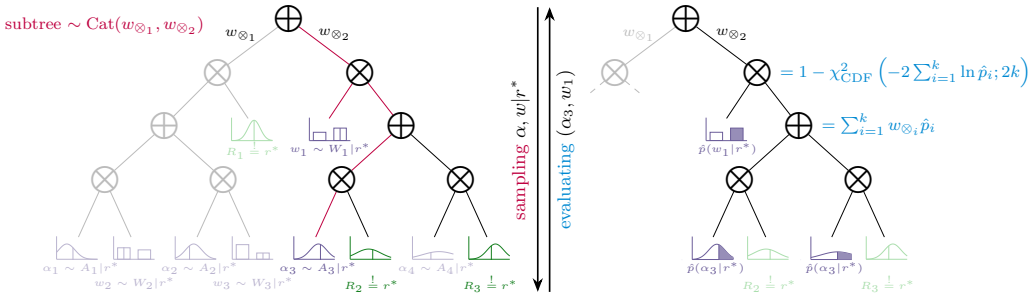



Figure 2: **DisCoMBO performs DisCo-guided candidate generation.** To generate new candidate configurations, DisCoMBO samples a large set of configurations from the conditional $p(\mathbf{x}|y^*)$ *(Left)* along with random candidates from the search space. These configurations are considered potentially promising and are scored using DisCo in a single bottom-up pass *(Right)*. Resulting DisCo scores are used to guide exploration and exploitation as well as to rate conformance of potentially given priors.

## 2 DisCoMBO: DisCo-guided Model-Based Optimization

To bridge the gap between purely generative Probabilistic Circuit (PC) driven SMBO and a rigorous uncertainty-guided exploration-exploitation trade-off, we first introduce the Distributional Conformance Score (DisCo) to measure model surprise in PCs. Building on this, we then present DisCoMBO, demonstrating how DisCo enables principled exploration-exploitation trade-offs and robust, bounded, and scale-invariant assessment of externally provided knowledge.

### 2.1 Probabilistic Circuits (PCs)

Since DisCo is defined based on PCs, let us first introduce PCs following Choi et al. [2020]. A PC is a computational graph encoding a distribution over a set of random variables $\boldsymbol{\mathcal{X}}$. It is defined as a tuple $(\mathcal{G}, \phi)$ where $\mathcal{G} = (V, E)$ is a rooted, directed acyclic graph and $\phi : V \to 2^{\boldsymbol{\mathcal{X}}}$ is the *scope* function assigning a subset of random variables to each node in $\mathcal{G}$. For each internal node $\mathsf{N}$ of $\mathcal{G}$, the scope is defined as the union of scopes of its children, i.e., $\phi(\mathsf{N}) = \cup_{\mathsf{N}' \in \mathrm{ch}(\mathsf{N})} \phi(\mathsf{N}')$. Each leaf node $\mathsf{L}$ computes a distribution/density over its scope $\phi(\mathsf{L})$. Internal nodes of $\mathcal{G}$ are either a (weighted) sum node $\mathsf{S}$ or a product node $\mathsf{P}$ where each sum node computes a convex combination of its children, i.e., $\mathsf{S} = \sum_{\mathsf{N} \in \mathrm{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{N}} \mathsf{N}$, and each product node computes a product of its children, i.e., $\mathsf{P} = \prod_{\mathsf{N} \in \mathrm{ch}(\mathsf{P})} \mathsf{N}$ [Choi et al., 2020]. In this work, we will consider *decomposable* and *smooth* PCs. Decomposability means that for product node $\mathsf{P}$, the scopes of its children are disjoint: $\forall \mathsf{N}_i, \mathsf{N}_j \in \mathrm{ch}(\mathsf{P}), i \neq j : \phi(\mathsf{N}_i) \cap \phi(\mathsf{N}_j) = \emptyset$ whereas smoothness means that for every sum node $\mathsf{S}$, its children share the same scope: $\forall \mathsf{N} \in \mathrm{ch}(\mathsf{S}) : \phi(\mathsf{N}) = \phi(\mathsf{S})$. Both are required to ensure tractable marginalization and conditioning in PCs [Choi et al., 2020]. See Fig. 2 (left) for an illustration and App. A for more details.

### 2.2 Distributional Conformance Score (DisCo)

In their vanilla form, PCs lack native epistemic uncertainty estimates, which are essential for well-defined exploration-exploitation trade-offs. A common heuristic is to treat the likelihood $p(\mathbf{x})$ as a proxy for confidence; however, $p(\mathbf{x})$ is an unreliable measure of model "surprise." Because the magnitude of $p(\mathbf{x})$ is determined by leaf parameterization and the dimensionality of $\boldsymbol{\mathcal{X}}$, a low density does not necessarily imply high uncertainty, but may simply reflect a broad, well-understood distribution. To address this, we introduce the Distributional Conformance Score (DisCo).

Akin to $p$-values, DisCo aggregates the cumulative probability of events that are "more extreme" than the current realization $\mathbf{x}$ under the model $p$, allowing to quantify the degree to which $\mathbf{x}$ is within the modeled distribution. Formally, DisCo provides a normalized metric in $[0, 1]$ where values near $0$ indicate high surprise and values near $1$ indicate that the model is well-conformed to the observation. A main advantage of DisCo is its computational efficiency; given a PC $p$ and a realization $\mathbf{x}$, the score is computed in a single bottom-up pass, guaranteeing inference time linear in the circuit size. We define the computation recursively across the graph structure:

**Leaf Nodes.** For a leaf node $\mathsf{L}$ with scope $s = \phi(\mathsf{L})$, the localized conformance $\rho_{\mathsf{L}}$ is defined by the two-tailed probability of the observation under the leaf's marginal distribution:

$$\rho_{\mathsf{L}} = 2 \cdot \min\big(F_{\mathsf{L}}(\mathbf{x}_s), 1 - F_{\mathsf{L}}(\mathbf{x}_s)\big) \tag{1}$$

where $F_{\mathsf{L}}$ denotes the cumulative distribution function (CDF) of the leaf. Note that this ensures that $\rho_{\mathsf{L}}$ bounded in $[0, 1]$.

**Sum Nodes.** Sum nodes maintain their latent variable interpretation as mixture models. To compute the mixture's conformance, we compute the convex combination of the scores of its children:

$$\rho_{\mathsf{S}} = \sum\nolimits_{\mathsf{N} \in \mathrm{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{N}} \cdot \rho_{\mathsf{N}} \tag{2}$$

Since $\sum_{\mathsf{N} \in \mathrm{ch}(\mathsf{S})} = 1$ by the definition of PCs, the score is kept bounded in an interval $[0, 1]$.

**Product Nodes.** For a product node $\mathsf{P}$ with $k$ children, decomposability ensures that each child represents an independent random variable. To aggregate evidence across these independent dimensions, we leverage Fisher's method [Fisher, 1932], which satisfies three key desiderata for our setting: (1) values are bounded in $[0, 1]$, (2) surprise in a single dimension reliably propagates to the aggregated score, and (3) it scales linearly in time (see App. D for further discussion). Although we do not perform formal hypothesis testing or compute statistically valid $p$-values, we exploit the property that the sum of log-transformed independent tail probabilities follows a $\chi^2$ distribution with $2k$ degrees of freedom to construct our proxy score for "model surprise":

$$\rho_{\mathsf{P}} = 1 - F_{\chi^2}\Big(-2 \cdot \sum\nolimits_{i=1}^{k} \ln(\rho_i); 2k\Big). \tag{3}$$

Here, $F_{\chi^2}(\cdot; 2k)$ is the $\chi^2$ CDF. Since any PC can be expanded into its induced tree representation [Zhao et al., 2016], the global DisCo score $\rho_p$ of a PC $p$ can be expressed in closed form:

$$\rho_p(\mathbf{x}) = \sum\nolimits_{t=1}^{\tau_s} \prod\nolimits_{(k,j) \in \mathcal{T}_t} w_{kj} \cdot \Big(1 - F_{\chi^2}\big(-2 \cdot \sum\nolimits_{i=1}^{n} \ln(\rho_i); 2n\big)\Big). \tag{4}$$

Here, $\tau_s$ is the number of induced trees, $\mathcal{T}_t$ is the $t$-th unique induced tree of $p$ and $\rho_i$ is the DisCo score of leaf $\mathsf{L}_i$ in $\mathcal{T}_t$. Each tree $\mathcal{T}_t$ contains the root of the PC, for each sum node it contains exactly one child and its corresponding edge, and for each product $\mathcal{T}_t$ contains all children. This admits a mixture representation of PCs with depth one (see App. A for more details).

In the following proposition, we use this representation to show that DisCo can be mapped to a behavior similar to kernel-based uncertainty estimates under some circumstances.

**Proposition 1** (DisCo is order-isomorphic to symmetric kernels)**.** *Let $p$ be a PC defined over real-valued random variables $X_1, \ldots, X_n$, where each $X_i$ is modeled by $m$ leaves. Each leaf $\mathsf{L}_{ij}$ is characterized by a symmetric density $f$ with a location parameter $\theta_{ij}$ and a monotone cumulative distribution function (CDF) $\Phi$. Let $d : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}} \to \mathbb{R}_{\geq 0}$ be a $p$-norm and $k : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}} \to \mathbb{R}$ be a symmetric, strictly increasing bounded kernel. Assume that both the leaf densities and $k$ are strictly monotonic functions of $d$ up to constant factors, define $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_j\}_{j=1}^{m}$. Then, for any two points $\mathbf{x}_1, \mathbf{x}_2 \in \boldsymbol{\mathcal{X}}$ and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $\rho_p$ is order-isomorphic to the kernel $k(\cdot, \cdot)$, such that $d(\mathbf{x}_1, \boldsymbol{\theta}) > d(\mathbf{x}_2, \boldsymbol{\theta})$ implies both $(1 - \rho_p(\mathbf{x}_1; \boldsymbol{\theta})) > (1 - \rho_p(\mathbf{x}_2; \boldsymbol{\theta}))$ and $k(\mathbf{x}_1; \boldsymbol{\theta}) > k(\mathbf{x}_2; \boldsymbol{\theta})$.*

*Proof (Sketch).* To show that $k$ and $1 - \rho_p(\cdot)$ are order-isomorphic, we show that due to the coupling of $d$ to $k$ and $\rho_p$, the values of $k$ increase when $\rho_p$ decreases and vice versa. Intuitively, we make use of applying the $\min$-operator to find out whether a datapoint lies in the left or right half of the symmetric leaf density of the PC. We then proceed by showing that applying Fisher [1932] to obtain DisCo scores of product nodes exactly inverts the ranking, which is kept by the subsequent mixture aggregation of the root node in the induced tree representation. Since we use $(1 - \rho_p(\cdot))$ as an uncertainty proxy, the ranking is inverted once again, yielding the same ranking as $k$, which is assumed to only be a constant scale and shift of $d$. See App. B.1 for the full proof. □

Thus, DisCo provides an uncertainty estimate functionally similar to a symmetric kernel commonly used in BO algorithms when the circuit structure scales with the number of observations. Crucially, by exploiting the decomposability and smoothness of PCs, DisCoMBO enables exact inference in $O(n)$ relative to the circuit size, bypassing the $O(n^3)$ bottleneck of GPs while maintaining comparable decision-theoretic benefits.

### 2.3 DisCo-Guided SMBO and Knowledge Integration

With DisCo at hand, we are ready to propose DisCoMBO, the first PC-based SMBO framework guided by rigorous uncertainty quantification. DisCoMBO synthesizes the natural knowledge integration of generative PC-based SMBO with the exploration-exploitation semantics of traditional SMBO. This unification enables a formal detection and feedback mechanism for external expertise that contradicts empirical evidence. The resulting procedure (Algorithm 1) ensures robust optimization by treating expert-in-the-loop priors as informative yet fallible inputs, preserving convergence guarantees even under misleading priors.

**Algorithm 1 DisCoMBO**

1: **Input:** Search space $\boldsymbol{\mathcal{X}}$, objective $f : \boldsymbol{\mathcal{X}} \to \mathbb{R}$, user prior $q(\hat{\boldsymbol{\mathcal{X}}} \subset \boldsymbol{\mathcal{X}})$ (can be provided at any time), decay $\gamma$, number of bins $K$, binning schedule $s$
2: Sample $J$ configurations $\{\mathbf{x}\}_{i=1}^{J}$ randomly
3: $\boldsymbol{\mathcal{D}} \leftarrow \{(\mathbf{x}_i, f(\mathbf{x}_i)\}$ for $i \in \{1, ..., J\}$
4: **while** not converged **do**
5: Fit PC $p$ on $\boldsymbol{\mathcal{D}}$
6: Set $y^* \leftarrow \max Y \in \boldsymbol{\mathcal{D}}$
7: **if** $d \sim$ Bernoulli$(\gamma^t p_{\text{use_prior}}) = 1$ **then**
8: Sample $\hat{\mathbf{x}} \sim q(\hat{\boldsymbol{\mathcal{X}}})$ if $q$, else None
9: **if** $\rho_p(\hat{\mathbf{x}}) < \tau$ **then** $\hat{\mathbf{x}} \leftarrow$ None
10: Sample $\mathcal{C} = \{\mathbf{x} \sim p(\boldsymbol{\mathcal{X}}|\hat{\mathbf{x}}, y^*)\}_{i=1}^{N}$
11: **if** use random **then**
12: $\mathcal{C} = \mathcal{C} \cup \{\mathbf{x} \sim \mathcal{U}(\boldsymbol{\mathcal{X}})\}_{i=1}^{M}$
13: Compute DisCo scores $\mathbf{r} = [\rho_p(\mathbf{x}) \forall \mathbf{x} \in \mathcal{C}]$
14: Compute annealing $\beta_t = s(t)$
15: Compute bins $\mathbf{b} = [(\frac{b}{K})^{\beta_t}]_{b=0}^{K-1}$
16: Assign DisCo scores to corresponding bins
17: Sample $\rho$ from randomly chosen bin
18: Define $\mathbf{X} = \{\mathbf{x} : \mathbf{x} \in \mathcal{C} | \rho_p(\mathbf{x}) = \rho\}$
19: $\boldsymbol{\mathcal{D}} \leftarrow \boldsymbol{\mathcal{D}} \cup \{(\mathbf{x}', f(\mathbf{x}'))\}$ for random $\mathbf{x}' \sim \mathbf{X}$

Specifically, DisCoMBO operates in two phases: (1) *knowledge-informed candidate generation*, utilizing the tractability of PCs and DisCo-based prior quality assessment, and (2) the *final candidate selection*, leveraging DisCo to manage the exploration-exploitation trade-off.

**(1) Knowledge-informed Candidate Generation:** At each iteration $t$, we use LearnSPN [Gens and Domingos, 2013] to fit a PC $p(\boldsymbol{\mathcal{X}}, Y)$ to the observed data $\boldsymbol{\mathcal{D}}$ **(Line 5)**. To construct the candidate set $\mathcal{C}$, we leverage the tractability of PCs to perform *prior-conditioned sampling*. Given external knowledge as a prior $q(\hat{\boldsymbol{\mathcal{X}}})$ over a (possibly empty) subspace $\hat{\boldsymbol{\mathcal{X}}} \subset \boldsymbol{\mathcal{X}}$, we sample $\hat{\mathbf{x}} \sim q$ and condition the PC on both this prior and the best-observed performance $y^* = \max Y \in \boldsymbol{\mathcal{D}}$:

$$p(\boldsymbol{\mathcal{X}}'|\hat{\mathbf{x}}, y^*) = \frac{p(\boldsymbol{\mathcal{X}}', \hat{\boldsymbol{\mathcal{X}}} = \hat{\mathbf{x}}, Y = y^*)}{\int_{\boldsymbol{\mathcal{X}}'} p(\boldsymbol{\mathcal{X}}', \hat{\boldsymbol{\mathcal{X}}} = \hat{\mathbf{x}}, Y = y^*)} \tag{5}$$

where $\boldsymbol{\mathcal{X}}' = \boldsymbol{\mathcal{X}} \setminus \hat{\boldsymbol{\mathcal{X}}}$. We then sample $N$ configurations $\mathbf{x} = (\hat{\mathbf{x}}, \mathbf{x}')$ where $\mathbf{x}' \sim p(\boldsymbol{\mathcal{X}}'|\hat{\mathbf{x}}, y^*)$ (**Line 6-9**). This strategy introduces a proximity bias towards high-performing regions weighted by the expert prior, while $\mathcal{C}$ is optionally augmented with $M$ random samples from $\boldsymbol{\mathcal{X}}$ to maintain global coverage (**Line 10-11**). To safeguard against misleading priors, we employ a two-fold recovery mechanism. The first layer consists of a semantic prior filter that utilizes the DisCo score to quantify how well a sample $\hat{\mathbf{x}} \sim q$ conforms to the learned surrogate. By marginalizing the PC over the remaining search space $\boldsymbol{\mathcal{X}}'$ and conditioning the PC on $Y = y^*$, we obtain $p'(\hat{\boldsymbol{\mathcal{X}}}|y^*)$ and compute the conformance $\rho_p(\hat{\mathbf{x}})$ as described in Sec. 2.2. If $\rho_p(\hat{\mathbf{x}}) < \tau$ for a threshold $\tau \in [0, 1]$, the sample is rejected for failing to align with empirical evidence (**Line 9**). This is complemented by a fallback mechanism following Seng et al. [2025], which decays the probability of utilizing the external prior $q$ by a factor $\gamma$ at each iteration, ensuring the optimizer asymptotically reverts to the default SMBO regime without external priors (**Line 7-8**).

**(2) Final Candidate Selection:** From the generated set $\mathcal{C}$, we select the most promising configuration for evaluation by computing the DisCo score $\rho_p(\mathbf{x})$ for each $\mathbf{x} \in \mathcal{C}$ (**Line 13**). To calculate $\rho_p(\mathbf{x})$ in the absence of an observed $Y$, we apply a proxy strategy: setting $Y = y^*$ yields scores reflecting conformance with high-performing regions (useful to bias towards proximity to $\mathbf{x}^*$ during exploitation), while marginalizing $Y$ allows DisCo to quantify general "novelty" relative to all prior observations (useful to foster more aggressive exploration). Which strategy is applied can either be fixed (e.g., on use the $Y = y^*$ strategy) or can depend on the schedule $s(t)$ (e.g., alternating both strategies). Selection is governed by an annealing parameter $\beta_t$ derived from a schedule $s(t)$, which modulates the exploration-exploitation trade-off at iteration $t$ (**Line 14**). We partition candidates into bins within the unit interval $[0, 1]$ based on their $\rho_p(\mathbf{x})$ values. Under this scheme, $\beta_t \gg 1$ causes the selection probability to concentrate around bins near 0 (high uncertainty/exploration), while $\beta_t \to 0$ shifts focus toward bins near 1 (low uncertainty/exploitation) (**Line 15**). The final candidate $\mathbf{x}'$ is then sampled from a randomly selected bin according to this distribution (**Line 16-19**). This work

employs a sinusoidal schedule $s(t)$ ensuring smooth transitions between exploration ($\beta_t \gg 1$) and exploitation ($\beta_t \to 0$). However, the framework admits any functional form, including dynamic schedules based on surrogate "energy" or uncertainty decay. We leave the exploration of different scheduling alternatives for future work.

**DisCoMBO is Zero Regret and Efficient** After presenting DisCoMBO, we analyze its convergence behavior. We find that DisCoMBO is a zero regret algorithm as stated in the following proposition.

**Proposition 2** (DisCoMBO is a zero-regret algorithm)**.** *Assume some $f \in \mathcal{H}_k$ with $\mathcal{H}_k$ being a Reproducing Kernel Hilbert Space (RKHS) with squared exponential kernel $k = \phi \circ \rho$ such that $f : \boldsymbol{\mathcal{X}} \to \mathbb{R}$ and $\mathbf{x}^*$ being the optimum of $f$. Further, assume a periodic schedule $s(t) \in \mathbb{R}_{\geq 0}$ that determines whether focus is set on exploitation or exploration, s.t. exploitation periods get longer over time, i.e. $\pi_t = c \cdot \pi_{t-1}$ for some $c > 1$ and period length $\pi_t$. Let the number of bins $K \to \infty$ and $p(y|\mathbf{x})$ be unbiased where $p$ is a PC learned on observations $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. Then, DisCoMBO has sub-linear cumulative regret $R_T$, i.e. $\frac{R_T}{T} \to 0$ with $T \to \infty$.*

*Proof (Sketch).* To show that DisCoMBO is a zero regret algorithm, we exploit the fact that DisCo is order-isomorphic to symmetric kernels (see Prop. 1). Given that we assume a periodic exploration-exploitation schedule $s(t)$ which controls the exploration and exploitation, we show that DisCoMBO's cumulative regret grows sub-linear in exploration and exploitation. With $T \to \infty$, we switch infinitely often between exploration and exploitation, thus converging to zero regret eventually. See App. B.2 for the full proof. □

Furthermore, since DisCoMBO leverages PCs as surrogates, all inference and sampling operations (including computation of DisCo) are linear in inference time w.r.t. the circuit size. Therefore, DisCoMBO also scales linearly with increasing sample and search space sizes.

## 3 Experimental Evaluation

We provide an extensive empirical evaluation of DisCoMBO to investigate the following research questions: **(Q1) Competitive Performance:** Can DisCoMBO bridge the performance gap between purely generative PC-based methods and state-of-the-art Bayesian or ensembling surrogates in settings without external knowledge guidance? **(Q2) Effectiveness of Informed Optimization:** Does DisCoMBO effectively translate external domain expertise into increased sample efficiency, and does it maintain the effectiveness of generative knowledge integration? **(Q3) Robustness and Recovery:** Does the DisCo score provide a reliable, accurate signal for detecting misleading expertise, and does it enable principled recovery in the presence of misleading priors?

**Experimental Setup** To evaluate DisCoMBO as a general-purpose optimizer (**Q1**), we compared it against established SMBO frameworks with various surrogates, including BO with Random Forests (RF) [Head et al., 2018], BO with Tree-Parzen Estimators (TPE) [Akiba et al., 2019], SMAC [Hutter et al., 2011, Lindauer et al., 2022], HEBO (using a GP surrogate) [Cowen-Rivers et al., 2022], and CMA-ES [Hansen, 2023]. To assess its performance as a generative surrogate, we compared it against IBO-HPC [Seng et al., 2025]. For knowledge integration tasks (**Q2**, **Q3**), we evaluated against IBO-HPC, $\pi$BO [Hvarfner et al., 2022], and DynaBO [Fehring et al., 2026]. Our evaluation spans 9 tasks and 6 search spaces—including continuous, discrete, graph, and mixed domains—across multiple fields. For AutoML, we utilize NAS-Bench-201 [Dong and Yang, 2020], JAHS [Bansal et al., 2022], and HPO-B [Pineda-Arango et al., 2021] to cover NAS and HPO. In the Physical Sciences, we employ the PV crossed-barrel benchmark [Gongora et al., 2020] for material discovery and the Floris Wake Modeling suite for windpark optimization [Fleming et al., 2020]. We repeat each experiment with 50 different seeds to ensure statistical significance of our results. We report normalized regret against wall-clock time whenever wall-clock time measurements are provided by benchmarks (iterations if not). See App. C.1 for a detailed overview.

**Knowledge Integration Protocol** To evaluate the efficacy of knowledge integration, we followed the protocol in Seng et al. [2025] to define *beneficial* and *misleading* priors over search space subspaces. External knowledge is provided as priors over a subset of dimensions. Beneficial knowledge favors values near the global optima $\mathbf{x}^+$ for a small subset of dimensions. Conversely,

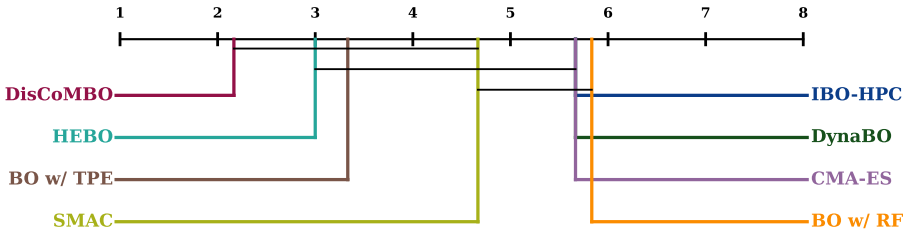



Figure 3: **DisCoMBO is competitive to top-tier SMBO methods.** DisCoMBO ranks **first** across 9 tasks when **no prior** is given, being statistically indistinguishable to strong baselines like HEBO. DisCoMBO significantly outperforms IBO-HPC, its PC-based counterpart, lacking a well-defined exploitation-exploration trade-off. Assessment is based on Wilcoxon signed-rank test ($p = 0.05$).

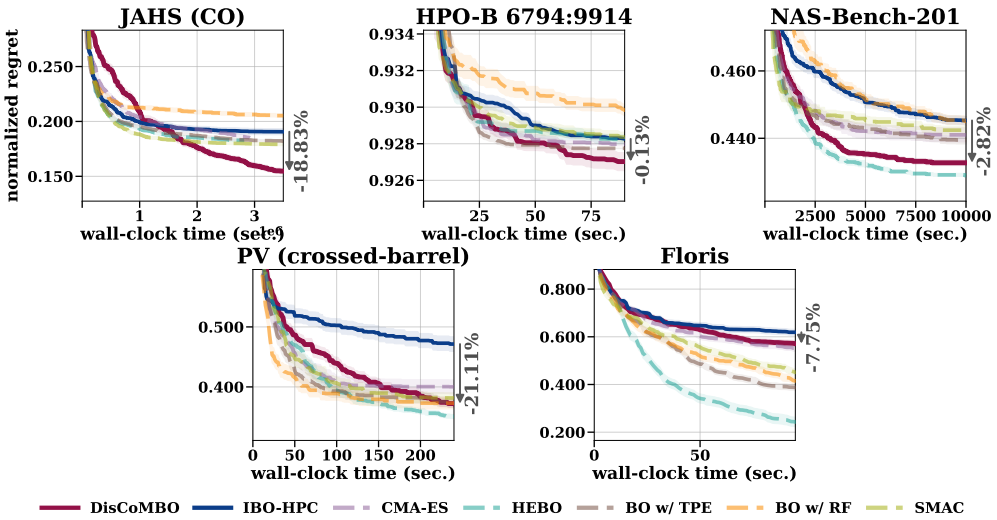



Figure 4: **DisCoMBO is effective on diverse tasks.** DisCoMBO (—) effectively optimizes black-box functions across tasks with diverse search spaces and optimization landscapes. In **8/9** tasks, DisCoMBO is competitive to top-tier SMBO algorithms (see App. C.4 for remaining tasks). We achieve significant improvement upon IBO-HPC (—) across **9/9** tasks.

misleading knowledge favors regions near poor-performing configurations $\mathbf{x}^-$ across a larger subset of dimensions to rigorously test the robustness of the DisCo rejection mechanism. This setup allows us to quantify the surrogate's ability to benefit from helpful hints while maintaining safety when faced with misleading domain expertise (see App. C.3 for exact definitions).

### 3.1 (Q1) DisCoMBO is Competitive to Traditional SMBO

To evaluate whether DisCoMBO successfully bridges the gap between generative modeling and decision-theoretic rigor, we compared it against six established baselines—including state-of-the-art SMBO methods and the generative IBO-HPC—in the unassisted setting. As shown in the critical difference diagram (Fig. 3), DisCoMBO significantly outperforms the generative baseline (IBO-HPC) and ranks first across the aggregated benchmark tasks. Statistically, DisCoMBO is indistinguishable from top-tier SMBO baselines, whereas IBO-HPC ranks significantly lower. These results, supported by the regret analysis in Fig. 4 and Fig. 7, confirm that DisCoMBO (—) achieves competitive performance to traditional SMBO while maintaining the unique flexibility of the PC architecture. DisCoMBO demonstrates multi-domain robustness, excelling across highly distinct optimization landscapes. It achieves top-tier performance on the materials discovery benchmark (PV Crossed-Barrel) and on structurally diverse AutoML tasks, natively navigating classical hyperparameter spaces (HPO-B), topological graphs for neural architectures (NAS-Bench-201), and complex joint spaces

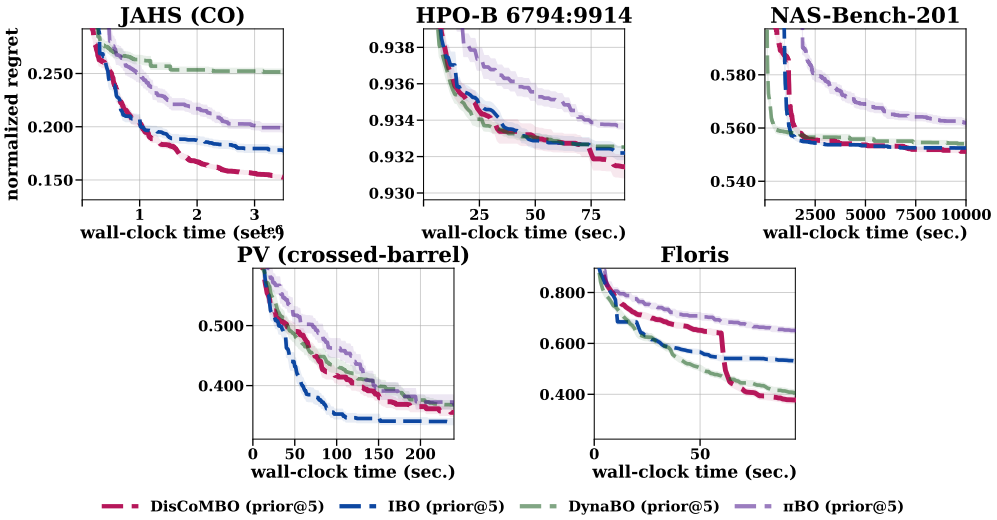



Figure 5: **DisCoMBO leverages beneficial priors effectively.** We find that DisCoMBO (- - ) retains IBO-HPC's (- - ) efficacy in incorporating priors. We outperform baselines or perform competitively on **7/9** tasks. See App. C.4 for additional results and discussion.

combining both (JAHS). This consistent performance highlights DisCoMBO's ability to capture diverse search space semantics without degradation. The only exception to this dominant profile occurs on Floris, a purely continuous landscape where DisCoMBO substantially outperforms IBO-HPC but falls slightly short of traditional SMBO. We attribute this to the topological properties of the Floris objective, which naturally favors discriminative surrogates explicitly optimized for continuous regression (see App. C.4 for an extended discussion). Nonetheless, DisCoMBO consistently outperforms prior PC-based optimization on **9/9** tasks and remains highly competitive with the state of the art on **8/9** tasks. We thus answer **(Q1)** affirmatively, establishing DisCoMBO as a robust, principled successor to purely generative PC-based methods.

### 3.2 (Q2, Q3) DisCoMBO Integrates Priors Effectively and is Resilient

We demonstrate that DisCoMBO retains and refines the capabilities for targeted and robust integration of external priors. We evaluate DisCoMBO under beneficial and adversarial expert guidance; detailed specifications for these prior distributions are provided in App. C.3. Lastly, we show that DisCo accurately discriminates between beneficial and misleading priors.

**Beneficial Priors** As illustrated in Fig. 5 and Fig. 9, DisCoMBO consistently retains or enhances the targeted knowledge integration characteristic of generative PC-based approaches when guided by beneficial priors. Across our evaluation, DisCoMBO matches or outperforms the purely generative IBO-HPC baseline. Furthermore, although optimizing assisted convergence was not the primary objective of this framework, DisCoMBO outperforms or matches baselines on **7/9** tasks. This robust behavior is replicated across the remaining JAHS, HPO-B, and Floris benchmarks (see App. C.4), confirming that our novel uncertainty semantics does not degrade the efficacy of generative prior guidance. Given these results, we answer **(Q2)** affirmatively.

**Recovery and Misleading Prior Detection** We evaluate DisCoMBO's resilience against misleading priors across three operational dimensions: the prior detection mechanism, the fallback recovery layer, and the capacity to handle sequences of contradicting priors. As shown in Fig. 6 (Left) and Fig. 11, when a severely misleading prior is initially accepted, DisCoMBO successfully recovers over time (-·-), demonstrating robust fallback behavior under worst-case assumptions. Furthermore, the framework effectively handles sequences of alternating beneficial and highly misleading priors introduced at different optimization phases (-·-), maintaining stability under these adversarial conditions (see App. C.3 for exact prior definitions). To isolate the prior detection mechanism from downstream optimization dynamics, we evaluate its accuracy against the heuristic-driven safeguard of DynaBO—the only baseline equipped with a comparable detection layer. Specifically, we measure

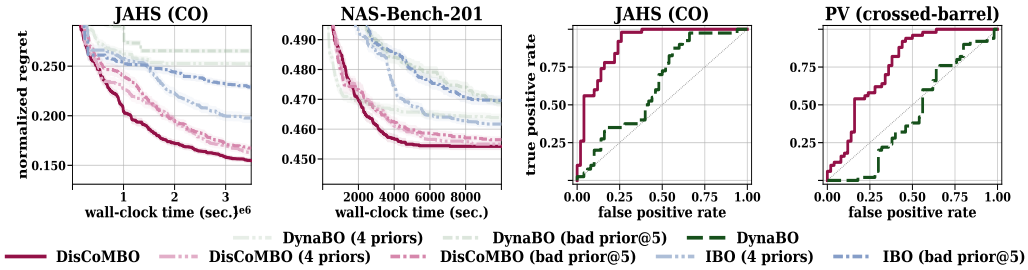



Figure 6: **DisCoMBO successfully recovers from and detects misleading priors.** (*Left*) DisCoMBO effectively recovers (- · -) from misleading priors, even in worst-case scenarios in which misleading priors are always accepted. Also, it effectively handles sequences of priors (- · -) with contradicting information. (*Right*) We find that our DisCo-based detection mechanism's accuracy (—) is significantly higher compared to DynaBO's heuristic-based detection mechanism (- - ), indicating DisCoMBO is able to detect misleading priors.

detection accuracy after 10%, 20%, and 30% of the allowed budget. Fig. 6 (Right) and Fig. 24 illustrate the ROC curve aggregated across iterations on JAHS (CO) and PV to ensure task diversity. DisCo-based detection yields a consistently higher AUC compared to DynaBO's heuristic approach, demonstrating that DisCo enables robust and accurate detection of misleading priors. We thus answer **(Q3)** affirmatively.

## 4 Related Work

In this section, we contextualize DisCoMBO within the broader landscape of sequential optimization, examining the evolution of external knowledge integration in SMBO and the emerging role of Probabilistic Circuits as a flexible alternative to traditional surrogates.

**Sequential Model Based Optimization (SMBO) with External Priors** SMBO seeks to optimize expensive black-box functions $f : \boldsymbol{\mathcal{X}} \to \mathbb{R}$ by alternating between a probabilistic surrogate and a selection policy determining the next query point $\mathbf{x}'$ [Shahriari et al., 2016]. Although effective, SMBO often significantly benefits from additional information available to simplify the problem at hand or provide additional guidance through the search process. To improve sample efficiency, especially in complex spaces, various strategies for incorporating domain expertise have emerged. Existing approaches either impose "hard" constraints to prune the search space *a priori* [Perrone et al., 2019, Ramachandran et al., 2020], which risks excluding global optima, or leverage "warm start" transfer learning from structured historical logs [Salinas et al., 2020, Bai et al., 2023]. More flexible "soft" integration methods, such as those proposed by Souza et al. [2021], Hvarfner et al. [2022], and Fehring et al. [2026], re-weight the acquisition function according to user-specified priors. However, this re-weighting mechanism can result in an opaque selection process where the prior's influence is mediated indirectly through the acquisition landscape. To tackle this, Seng et al. [2025] exploits the conditional sampling of Probabilistic Circuits (PCs) to allow for more targeted knowledge integration. Yet, existing generative routines lack the formal exploration–exploitation semantics intrinsic to traditional SMBO and provide no normalized, scale-invariant metric to evaluate the quality of the external guidance. Our work, DisCoMBO, bridges these gaps by introducing a formal conformance score that establishes a rigorous exploration-exploitation trade-off for PC-based surrogates while enabling the robust rejection of misleading external guidance.

**Probabilistic Circuits (PCs)** Probabilistic Circuits (PCs) are computational graphs that represent multivariate distributions while guaranteeing tractable exact inference for a broad range of probabilistic queries [Choi et al., 2020]. By imposing architectural constraints such as *decomposability* and *smoothness*, PCs reduce computationally expensive marginalization tasks to linear-time operations over tractable leaf distributions [Molina et al., 2018, Peharz et al., 2020]. Beyond their efficiency, PCs are uniquely suited for integrating external knowledge; unlike traditional regressive surrogates, their generative nature allows for the direct injection of domain expertise through conditional sampling, providing a more targeted alternative to heuristic re-weighting of acquisition functions or model

predictions [Seng et al., 2025]. While PC architectures range from GPU-optimized fixed structures [Peharz et al., 2020] to those learned directly from data [Gens and Domingos, 2013], the latter are particularly effective for the small-to-medium datasets typical of SMBO. These learned structures are derived by recursively partitioning data via clustering and independence tests, naturally capturing the structural dependencies of the search space. Given the hybrid nature of real-world optimization domains, DisCoMBO specifically utilizes Mixed Sum-Product Networks (MSPNs) [Molina et al., 2018], which extend the PC framework to mixed continuous and discrete variables using piecewise polynomial leaves. For aggregation of evidence, DisCo uses Fisher's method at product nodes as introduced in Membership Circuits [Wittelsbach et al., 2026]. For formal details, see App. A.

## 5 Conclusion

We introduced DisCo, a novel conformance score that quantifies model surprise for Probabilistic Circuits (PCs), and developed DisCoMBO, a principled PC-based SMBO framework. By leveraging DisCo as an uncertainty proxy, DisCoMBO simultaneously equips PC-based optimization with formal exploration-exploitation semantics and provides an accurate, scale-independent detection mechanism for misleading priors. Empirically, DisCoMBO successfully addresses the performance limitations of purely generative PC-based methods, operating on par with top-tier SMBO baselines across a diverse suite of real-world benchmarks. Crucially, it achieves this decision-theoretic rigor while fully retaining—and expanding upon—the targeted, robust external prior integration capabilities inherent to generative PC architectures.

**Limitations & Future Work** Despite these advancements, several promising avenues for future research remain. First, standard PC architectures are structurally constrained by a discrete latent variable formulation, which can introduce a discretization bias on smooth, fully continuous manifolds and limit regression accuracy. Advancing underlying PC parameterizations to natively support continuous topologies would directly enhance DisCoMBO's optimization efficiency on such landscapes. Furthermore, a deeper theoretical analysis of DisCo's kernel-like properties offers an exciting frontier. Refining this connection may ultimately allow PC-based optimization to bridge the gap between linear-time probabilistic inference and the elegant, mathematically rigorous properties characteristic of kernel methods like Gaussian Processes.

## Acknowledgements

This work was supported by the National High-Performance Computing Project for Computational Engineering Sciences (NHR4CES) and by the German Research Foundation (DFG) under Germany's Excellence Strategy (EXC-3057/1 "Reasonable Artificial Intelligence").

## References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, 2019.

Tianyi Bai, Yang Li, Yu Shen, Xinyi Zhang, Wentao Zhang, and Bin Cui. Transfer Learning for Bayesian Optimization: A Survey, 2023.

Archit Bansal, Danny Stoll, Maciej Janowski, Arber Zela, and Frank Hutter. Jahs-bench-201: A foundation for research on joint architecture and hyperparameter search. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. Technical report, UCLA, 2020.

Alexander I Cowen-Rivers, Wenlong Lyu, Rasul Tutunov, Zhi Wang, Antoine Grosnit, Ryan Rhys Griffiths, Alexandre Max Maraval, Hao Jianye, Jun Wang, Jan Peters, et al. Hebo: Pushing the limits of sample-efficient hyper-parameter optimisation. *Journal of Artificial Intelligence Research*, 74:1269–1349, 2022.

Xuanyi Dong and Yi Yang. NAS-Bench-201: Extending the scope of reproducible neural architecture search. In *International Conference on Learning Representations (ICLR)*, 2020.

Lukas Fehring, Marcel Wever, Maximilian Spliethöver, Leona Hennig, Henning Wachsmuth, and Marius Lindauer. Dynamic priors in Bayesian optimization for hyperparameter optimization. In *Proceedings of the Fifth International Conference on Automated Machine Learning*, 2026.

Ronald A Fisher. *Statistical methods for research workers*. Oliver and Boyd, 1932.

Paul Fleming, Jennifer King, Christopher J. Bay, Eric Simley, Rafael Mudafort, Nicholas Hamilton, Alayna Farrell, and Luis Martinez-Tossas. Overview of floris updates. *Journal of Physics: Conference Series*, 2020.

Robert Gens and Pedro Domingos. Learning the structure of sum-product networks. In *International Conference on Machine Learning*, pages 873–880. PMLR, 2013.

Aldair E Gongora, Bowen Xu, Wyatt Perry, Chika Okoye, Patrick Riley, Kristofer G Reyes, Elise F Morgan, and Keith A Brown. A Bayesian experimental autonomous researcher for mechanical design. *Science advances*, 6(15):eaaz1708, 2020.

Nikolaus Hansen. The cma evolution strategy: A tutorial, 2023.

Tim Head, MechCoder, Gilles Louppe, Iaroslav Shcherbatyi, fcharras, Zé Vinícius, cmmalone, Christopher Schröder, nel215, Nuno Campos, Todd Young, Stefano Cereda, Thomas Fan, rene rex, Kejia (KJ) Shi, Justus Schwabedal, carlosdanielcsantos, Hvass-Labs, Mikhail Pak, SoManyUsernamesTaken, Fred Callaway, Loïc Estève, Lilian Besson, Mehdi Cherti, Karlson Pfannschmidt, Fabian Linzberger, Christophe Cauet, Anna Gut, Andreas Mueller, and Alexander Fabisch. scikit-optimize/scikit-optimize, 2018. URL `https://doi.org/10.5281/zenodo.1207017`.

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*, pages 507–523. Springer, 2011.

Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. $\pi$bo: Augmenting acquisition functions with user beliefs for Bayesian optimization. In *International Conference on Learning Representations (ICLR)*, 2022.

Xinglong Ju, Feng Liu, Li Wang, and Wei Jen Lee. Wind farm layout optimization based on support vector regression guided genetic algorithm with consideration of participation among landowners. *Energy Conversion and Management*, 181:1104–1114, 2019.

Aaron Gilad Kusne, Heshan Yu, Changming Wu, Huairuo Zhang, Jason Hattrick-Simpers, Brian DeCost, Shubhadeep Sarker, Corey Oses, Cormac Toher, Stefano Curtarolo, Albert V Davydov, Ritesh Agarwal, Leonid A Bendersky, Manyuan Li, Apurva Mehta, and Ichiro Takeuchi. On-the-fly closed-loop materials discovery via Bayesian active learning. *Nature Communications*, 11(1):5966, 2020.

Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. Smac3: A versatile Bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54):1–9, 2022.

Marius Lindauer, Florian Karl, Anne Klier, Julia Moosbauer, Alexander Tornede, Andreas Müller, Frank Hutter, Matthias Feurer, and Bernd Bischl. Position: A call to action for a human-centered automl paradigm. In *International Conference on Machine Learning (ICML)*, 2024.

Alejandro Molina, Antonio Vergari, Nicola Di Mauro, Sriraam Natarajan, Floriana Esposito, and Kristian Kersting. Mixed sum-product networks: A deep architecture for hybrid domains. In *AAAI Conference on Artificial Intelligence*, 2018.

Heungseok Park, Yoonsoo Nam, Ji-Hoon Kim, and Jaegul Choo. Hypertendril: Visual analytics for user-driven hyperparameter optimization of deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 27, 2021.

Robert Peharz, Sebastian Tschiatschek, Franz Pernkopf, and Pedro Domingos. On theoretical properties of sum-product networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Robert Peharz, Steven Lang, Antonio Vergari, Karl Stelzner, Alejandro Molina, Martin Trapp, Guy Van den Broeck, Kristian Kersting, and Zoubin Ghahramani. Einsum networks: Fast and scalable learning of tractable probabilistic circuits. In *International Conference on Machine Learning (ICML)*, 2020.

Valerio Perrone, Huibin Shen, Matthias W Seeger, Cedric Archambeau, and Rodolphe Jenatton. Learning search spaces for Bayesian optimization: Another view of hyperparameter transfer learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Sebastian Pineda-Arango, Hadi S. Jomaa, Martin Wistuba, and Josif Grabocka. HPO-B: A large-scale reproducible benchmark for black-box HPO based on openml. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Anil Ramachandran, Sunil Gupta, Santu Rana, Cheng Li, and Svetha Venkatesh. Incorporating expert prior in Bayesian optimisation via space warping. *Knowledge-Based Systems*, 195, 2020.

David Salinas, Huibin Shen, and Valerio Perrone. A quantile-based approach for hyperparameter transfer learning. In *International Conference on Machine Learning (ICML)*, pages 8438–8448, 2020.

Elias Schede, Jasmin Brandt, Alexander Tornede, Marcel Wever, Viktor Bengs, Eyke Hüllermeier, and Kevin Tierney. A survey of methods for automated algorithm configuration. *Journal of Artificial Intelligence Research*, 75:425–487, 2022.

Jonas Seng, Fabrizio Ventola, Zhongjie Yu, and Kristian Kersting. Hyperparameter optimization via interacting with probabilistic circuits. In *Proceedings of the Fourth International Conference on Automated Machine Learning*, 2025.

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, 2016.

Artur L. F. Souza, Luigi Nardi, Leonardo B. Oliveira, Kunle Olukotun, Marius Lindauer, and Frank Hutter. Bayesian optimization with a prior for the optimum. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, 2021.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5), 2012.

Qianwen Wang, Yao Ming, Zhihua Jin, Qiaomu Shen, Dongyu Liu, Micah J. Smith, Kalyan Veeramachaneni, and Huamin Qu. Atmseer: Increasing transparency and controllability in automated machine learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

Bennet Wittelsbach, Arseny Skryagin, Jonas Seng, Moritz Willig, and Kristian Kersting. Membership circuits: Tractable membership testing via probabilistic circuits. In *The ICML 2026 Workshop on Hypothesis Testing*, 2026.

Wenjie Xu, Masaki Adachi, Colin Jones, and Michael A Osborne. Principled Bayesian optimization in collaboration with human experts. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Han Zhao, Tameem Adel, Geoff Gordon, and Brandon Amos. Collapsed variational inference for sum-product networks. In *International Conference on Machine Learning (ICML)*, volume 48, pages 1310–1318, 2016.

# A Probabilistic Circuits

Since probabilistic circuits (PCs) are a key component of our method, we provide more details on these models in the following. Let us first start with a rigorous definition of PCs.

**Definition 1.** *A probabilistic circuit (PC) is a computational graph encoding a distribution over a set of random variables* $\mathbf{X}$*. It is defined as a tuple* $(\mathcal{G}, \phi)$ *where* $\mathcal{G} = (V, E)$ *is a rooted, directed acyclic graph and* $\phi : V \to 2^{\mathbf{X}}$ *is the scope function assigning a subset of random variables to each node in* $\mathcal{G}$*. For each internal node* $\mathsf{N}$ *of* $\mathcal{G}$*, the scope is defined as the union of scopes of its children, i.e.* $\phi(\mathsf{N}) = \cup_{\mathsf{N}' \in \mathrm{ch}(\mathsf{N})} \phi(\mathsf{N}')$*. Each leaf node* $\mathsf{L}$ *computes a distribution/density over its scope* $\phi(\mathsf{L})$*. All internal nodes of* $\mathcal{G}$ *are either a sum node* $\mathsf{S}$ *or a product node* $\mathsf{P}$ *where each sum node computes a convex combination of its children, i.e.,* $\mathsf{S} = \sum_{\mathsf{N} \in \mathrm{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{N}} \mathsf{N}$*, and each product computes a product of its children, i.e.,* $\mathsf{P} = \prod_{\mathsf{N} \in \mathrm{ch}(\mathsf{P})} \mathsf{N}$.

With this definition at hand, we describe the tractable key operations of PCs relevant to our method in more detail.

**Inference** Inference in PCs is a bottom-up procedure. To compute the probability of given evidence $\mathbf{X} = \mathbf{x}$, the densities of the leaf nodes are evaluated first. This yields a density value for each leaf. The leaf densities are then propagated bottom-up by computing all product/sum nodes. Eventually, the root node holds the probability/density of $\mathbf{x}$. Note that typically, multiple leaf nodes correspond to the same random variable. Thus, if the children of a sum node have the same scope, we can interpret sum nodes as mixture models. Conversely, if the children of a product node have *non*-overlapping scopes, a product node can be interpreted as a product distribution of two (independent) random variables. We call these two properties smoothness and decomposability. More formally, *smoothness* means that for each sum node $\mathsf{S} \in V$ it holds that $\phi(\mathsf{N}) = \phi(\mathsf{N}')$ for $\mathsf{N}, \mathsf{N}' \in \mathrm{ch}(\mathsf{S})$. *Decomposability* means that for each product node $\mathsf{P} \in V$ it holds that $\phi(\mathsf{N}) \cap \phi(\mathsf{N}') = \emptyset$ for $\mathsf{N}, \mathsf{N}' \in \mathrm{ch}(\mathsf{P})$, $\mathsf{N} \neq \mathsf{N}'$. Hence, PCs can be interpreted as hierarchical mixture models.

**Marginalization** Decomposability implies that marginalization is tractable in PCs and can be done in linear time of the circuit size. This is because integrals that can be rewritten by nesting single-dimensional integrals can be computed only in terms of leaf integrals, which are assumed to be tractable as they follow certain distributions (e.g., Gaussian). Computing such nested integrals only in terms of leaf integrals is possible because single-dimensional integrals commute with the sum operation and affect only a single child of product nodes. For more details on the computational implications of decomposability, refer to [Peharz et al., 2015].

Practically, there are two ways to marginalize certain variables from the scope of a PC. One approach is structure-preserving, and marginalization is achieved by setting all leaves corresponding to the set of random variables that are supposed to be marginalized to 1. The second approach constructs a new PC representing the marginal distribution, i.e. the structure of the PC is changed. The second approach is beneficial if samples should be drawn from the marginalized PC because the sampling procedure remains the same, i.e. the PC is adapted to obtain the marginal distribution, not vice versa.

**Conditioning** Computing a conditional distribution $p(\mathbf{X}_1|\mathbf{X}_2) = \frac{p(\mathbf{X})}{\int_{\mathbf{X}_1} p(\mathbf{X})}$ where $\mathbf{X}_1 \cup \mathbf{X}_2 = \mathbf{X}$ and $\mathbf{X}_1 \cap \mathbf{X}_2 = \emptyset$ is achieved by combining marginalization (denominator) and inference (numerator). Since inference is tractable for PCs in general and marginalization is tractable for decomposable PCs, conditioning is also tractable.

**Sampling** Sampling in PCs is a top-down procedure and recursively samples a sub-tree, starting at the root. Each sum node $\mathsf{S}$ holds a parameter vector $\mathbf{w}$ s.t. $\sum_{i=0}^{|\mathrm{ch}(\mathsf{S})|} \mathbf{w}_i = 1$. Based on the distribution induced by $\mathbf{w}$, one of the children of $\mathsf{S}$ is sampled as a sub-tree. By decomposability, the scope of the children of a product node is non-overlapping; thus, sampling from a product node corresponds to sampling from all its child nodes. If a leaf node is reached, a sample is obtained from the distribution at that leaf.

**Learning** Learning PCs consists of two steps: Identifying the structure of the PC and learning the parameters of the PC. A common approach to learning both the structure and parameters is LearnSPN [Gens and Domingos, 2013]. We employ LearnSPN to learn the PC after obtaining new

data. The basic idea of LearnSPN is to split the data by alternating clustering (i.e., split the data along the sample dimension) and independence tests (i.e., split the data along the features dimension). In other words, the data matrix is split by rows (samples) and columns (features). Usually, rows are clustered when the independence test fails in splitting the features. Clusters correspond to sum nodes in the learned PC, while product nodes correspond to successfully passed independence tests (assessing that two subsets of features are statistically independent). The parameters (i.e., weights of sum nodes) are set proportional to the cluster sizes of clusters represented by the child nodes of a sum node. Leaf parameters are commonly defined via maximum likelihood estimation.

LearnSPN is efficient on small-to-medium datasets (e.g., $50$ variables and $1000$ samples), as it is based on K-Means and RDC for independence testing, resulting in a quadratic runtime complexity. Since we are interested in high sample efficiency in SMBO, we operate in the regime of small to medium-sized datasets.

**Induced Tree Representation** Here, we briefly describe the induced tree representation from Zhao et al. [2016]. We use this representation in our proofs.

**Definition 2.** *Induced Trees [Zhao et al., 2016]. Given a complete and decomposable PC $s$ over $\boldsymbol{\mathcal{H}} = \{H_1, \ldots, H_n\}$, $\mathcal{T} = (\mathcal{T}_V, \mathcal{T}_E)$ is called an induced tree PC from $s$ if*

1. $\mathsf{N} \in \mathcal{T}_V$ *where* $\mathsf{N}$ *is the root of* $s$.
2. *for all sum nodes* $\mathsf{S} \in \mathcal{T}_V$*, exactly one child of* $\mathsf{S}$ *in* $s$ *is in* $\mathcal{T}_V$*, and the corresponding edge is in* $\mathcal{T}_E$.
3. *for all product node* $\mathsf{P} \in \mathcal{T}_V$*, all children of* $\mathsf{P}$ *in* $s$ *are in* $\mathcal{T}_V$*, and the corresponding edges in* $\mathcal{T}_E$.

We can use Def. 2 to represent decomposable and complete PCs as mixtures [Zhao et al., 2016].

**Proposition 3** (Induced Tree Representation)**.** *Let $\tau_s$ be the total number of induced trees in $s$. Then the output at the root of $s$ can be written as $\sum_{t=1}^{\tau_s} \prod_{(k,j)\in\mathcal{T}_{tE}} w_{kj} \prod_{i=1}^{n} p_t(H_i = \boldsymbol{\theta}_i)$, where $\mathcal{T}_t$ is the $t$-th unique induced tree of $s$ and $p_t(H_i)$ is a univariate distribution over $H_i$ in $\mathcal{T}_t$ as a leaf node.*

# B Proofs

This section provides proofs for all propositions in the main paper.

## B.1 Order-Isomorphism between Kernels and DisCo

Here, we provide the proof that DisCo is order-isomorphic to symmetric kernels under certain parameterizations of PCs.

**Proposition 1** (DisCo is order-isomorphic to symmetric kernels)**.** *Let $p$ be a PC defined over real-valued random variables $X_1, \ldots, X_n$, where each $X_i$ is modeled by $m$ leaves. Each leaf $\mathsf{L}_{ij}$ is characterized by a symmetric density $f$ with a location parameter $\theta_{ij}$ and a monotone cumulative distribution function (CDF) $\Phi$. Let $d : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}} \to \mathbb{R}_{\geq 0}$ be a $p$-norm and $k : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}} \to \mathbb{R}$ be a symmetric, strictly increasing bounded kernel. Assume that both the leaf densities and $k$ are strictly monotonic functions of $d$ up to constant factors, define $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_j\}_{j=1}^{m}$. Then, for any two points $\mathbf{x}_1, \mathbf{x}_2 \in \boldsymbol{\mathcal{X}}$ and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $\rho_p$ is order-isomorphic to the kernel $k(\cdot,\cdot)$, such that $d(\mathbf{x}_1, \boldsymbol{\theta}) > d(\mathbf{x}_2, \boldsymbol{\theta})$ implies both $(1 - \rho_p(\mathbf{x}_1; \boldsymbol{\theta})) > (1 - \rho_p(\mathbf{x}_2; \boldsymbol{\theta}))$ and $k(\mathbf{x}_1; \boldsymbol{\theta}) > k(\mathbf{x}_2; \boldsymbol{\theta})$.*

*Proof.* To show order-isomorphism between $k(\cdot)$ and $1 - \rho_p(\cdot)$, we exploit the induced tree representation from Eq. 4. Further, we use that - by definition - $k$ is an affine transformation of $d$, thus $k$ preserves the order induced by $d$.

We now have to show three things:

1. DisCo preserves the order of $d$ in all leaf nodes, i.e., $\rho_{\mathsf{L}}$ has the same order as $d$.
2. Since product nodes combine DisCo scores of children using Fisher's method [Fisher, 1932], we have to show that Fisher is order-preserving w.r.t $d$, i.e., all $\rho_{\mathsf{P}}$ inherit the order induced by $d$.

3. The final weighted sum in the root node of the induced tree representation preserves the order of incoming child DisCo scores, i.e., $\rho_p$ inherits the order of $d$.

**$\rho_{\mathsf{L}}$ is order-preserving.** We start by showing that $\rho_{\mathsf{L}}$ preserves the order of $d$. Remember that each leaf $\mathsf{L}_{ij}$ has a symmetric density $f_{ij}$ that is an affine transformation of $d$, i.e. $f_{ij}(\cdot) = a_{ij} \cdot d(\cdot)$ with $a_{ij} > 0$ and s.t. $\int f_{ij}(x; \theta_{ij})dx = 1$ where $\theta_{ij}$ parameterizes the density. Remember

$$\rho_{\mathsf{L}_{ij}}(x) = 2 \cdot \min\Big( \int_{-\infty}^{x} f_{ij}(x; \theta_{ij}), \int_{x}^{\infty} f_{ij}(x; \theta_{ij}) \Big).$$

Since $f_{ij}$ is symmetric, where the symmetry point is given by its parametrization $\theta_{ij}$, $\rho_{\mathsf{L}_{ij}}$ is also symmetric at $\theta_{ij}$ because the probability mass is exactly split in two halves at the symmetry point and $\min$ only selects which side we consider. Thus, for $x_1, x_2 \in \text{supp}(f_{ij})$ we have to consider two cases:

1. $x_1 < \theta_{ij}$ and $x_2 < \theta_{ij}$
2. $x_1 < \theta_{ij}$ and $x_2 > \theta_{ij}$

Starting with 1., since $d(x_1, \theta_{ij}) < d(x_2, \theta_{ij})$ is assumed, we find that

$$\int_{-\infty}^{x_1} f_{ij}(x; \theta_{ij})dx > \int_{-\infty}^{x_2} f_{ij}(x; \theta_{ij})dx$$

and thus $\rho_{\mathsf{L}_{ij}}(x_1) > \rho_{\mathsf{L}_{ij}}(x_2)$, therefore $1 - \rho_{\mathsf{L}_{ij}}(x_1) < 1 - \rho_{\mathsf{L}_{ij}}(x_2)$.

Proceeding with 2., consider

$$\rho_{\mathsf{L}_{ij}}(x_1) = \int_{-\infty}^{x_1} f_{ij}(x; \theta_{ij})dx$$
$$\rho_{\mathsf{L}_{ij}}(x_2) = \int_{x_2}^{\infty} f_{ij}(x; \theta_{ij})dx.$$

This is because the $\min$ operator always selects the minimum area under the density curve which spans from $-\infty$ to $x_1$ for $x_1$ and spans from $x_2$ to $\infty$ for $x_2$. Since $f_{ij}$ is symmetric and $d(x_1, \theta_{ij}) < d(x_2, \theta_{ij})$ is assumed, $\rho_{\mathsf{L}_{ij}}(x_1)$ "collects" more mass than $\rho_{\mathsf{L}ij}(x_2)$, therefore $\rho_{\mathsf{L}_{ij}}(x_1) > \rho_{\mathsf{L}_{ij}}(x_2)$, and thus $1 - \rho_{\mathsf{L}_{ij}}(x_1) < 1 - \rho_{\mathsf{L}_{ij}}(x_2)$.

**Fisher preserves order.** Next, we show that applying Fisher to incoming DisCo values at product nodes preserves the order between points $\mathbf{x}_1, \mathbf{x}_2 \in \text{supp}(\mathsf{P})$ induced by $d$. Note that we now consider vectors instead of scalar values since product nodes model product distributions over multiple independent random variables. Remember that for a product $\mathsf{P}$, DisCo is computed as

$$\rho_{\mathsf{P}}(\mathbf{x}) = \Big(1 - F_{\chi^2}\big( -2 \cdot \sum_{i=1}^{j} \ln(\rho_i(\mathbf{x}_i)); 2j\big)\Big)$$

where $j$ is the number of children of $\mathsf{P}$. Again, we assume $d(\mathbf{x}_1, \boldsymbol{\theta}_{\mathsf{P}}) < d(\mathbf{x}_2, \boldsymbol{\theta}_{\mathsf{P}})$ where $\boldsymbol{\theta}_{\mathsf{P}}$ is the parameter vector of all $j$ leaf nodes connected to $\mathsf{P}$. Note that we can safely assume that $\mathsf{P}$ is only connected to exactly these $j$ children due to the induced tree representation. Since $d$ is a $p$-norm, we have

$$\Big( \sum_{i=1}^{j} |\mathbf{x}_{1,i} - \boldsymbol{\theta}_{\mathsf{P}_i}|^p \Big)^{1/p} < \Big( \sum_{i=1}^{j} |\mathbf{x}_{2,i} - \boldsymbol{\theta}_{\mathsf{P}_i}|^p \Big)^{1/p}$$

which implies

$$-2 \cdot \sum_{i=1}^{j} \ln(\rho_i(\mathbf{x}_{1,i})) < -2 \cdot \sum_{i=1}^{j} \ln(\rho_i(\mathbf{x}_{2,i}))$$

by 1. and the fact that ln is a monotonic function. Since $F_{\chi^2}$ is a monotonically increasing function as well and bound in $[0, 1]$, applying $1 - F_{\chi^2}(\cdot)$ yields $\rho_{\mathsf{P}}(\mathbf{x}_1) > \rho_{\mathsf{P}}(\mathbf{x}_2)$ and thus $1 - \rho_{\mathsf{P}}(\mathbf{x}_1) < 1 - \rho_{\mathsf{P}}(\mathbf{x}_2)$.

**Root node preserves order.** Finally, let us proceed with showing that under our assumptions, the weighted sum of the root node preserves the order induced by its children (and thus $d$). Since we assume that $\mathbf{x}_1$ and $\mathbf{x}_2$ share the same infimum $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, that is the same neighbor, the sum reduces to an affine transformation

$$\prod_{(j,k)\in\mathcal{T}_t} w_{j,k} \cdot \rho_{\mathsf{P}_{j,k}}(\mathbf{x}).$$

Therefore, the order is preserved and $1 - \rho_p(\mathbf{x}_1) < 1 - \rho_p(\mathbf{x}_2)$ holds. □

**Discussion of Assumptions** While we acknowledge that some of our assumptions might be violated in practice, especially in mixed domains, this result shows an interesting and promising bridge between uncertainty estimation in PCs and kernel-based methods.

From a modeling perspective, pairing symmetric leaf distributions in the PC with a symmetric kernel is a natural and well-justified choice. In classical Bayesian optimization, symmetric kernels (such as Gaussian or Matérn) encode the fundamental stationary prior that information propagates uniformly in all directions from an observed point. By utilizing symmetric leaf distributions (e.g., isotropic Gaussians or symmetric categorical priors), the PC surrogate mirrors this isotropic inductive bias locally within its tractable density estimation architecture. This alignment ensures that "model surprise" (DisCo) scales monotonically with the distance from existing data clusters, providing a coherent uncertainty proxy. While out of the scope of this paper, different PC parameterizations might allow for relaxed assumptions regarding the form of the kernel.

Beyond our theoretical analysis, we provide empirical evidence demonstrating that DisCo remains an effective uncertainty proxy even when the assumptions of Prop. 1 are relaxed—specifically within mixed or discrete search spaces. First, the regret analysis in App. C.4 shows that DisCo-based guidance consistently identifies high-performing regions across diverse landscapes. Furthermore, Fig. 24 demonstrates that DisCo reliably discriminates between beneficial and misleading priors across continuous, mixed, and discrete domains. This indicates that the score provides a robust signal for distinguishing conformant from non-conformant samples, regardless of the underlying domain structure.

### B.2 Regret Analysis

In this section, we provide the regret analysis of DisCoMBO.

**Lemma 1** (Coverage)**.** *Let $f$ be a function from a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k$ such that kernel $k = \phi \circ \rho$ where $\phi$ is a mapping such that $k$ is a valid symmetric kernel. Assume a PC $p$ such that $p(y|\mathbf{x})$ is unbiased, $f(\mathbf{x}) - \mathbb{E}[p(y|\mathbf{x})]$ is sub-Gaussian and $y = f(\mathbf{x}) + \epsilon$ with $\epsilon \sim$ SubGauss$(\sigma)$. Then, $\mathbb{E}[p(y|\mathbf{x})] + \beta_t \cdot \phi(\rho(\mathbf{x}))$ entails $f$ with probability $1 - \delta_t$, for a given $\beta_t \geq \sqrt{2ln(\frac{2}{\delta_t})}$.*

*Proof.* For an RKHS we have that $|f(\mathbf{x})| = \langle f, k(\mathbf{x}, \cdot)\rangle \leq ||f||_k \cdot \sqrt{k(\mathbf{x}, \mathbf{x})}$. Given the existence of $\phi$, we have

$$|f(\mathbf{x}) - \mathbb{E}[p(y|\mathbf{x})]| \leq ||f||_k \cdot \phi(\rho(\mathbf{x}))$$

Now, for $\beta_t$, we want that

$$P(|f(\mathbf{x}) - \mathbb{E}[p(y|\mathbf{x})]| \leq \beta_t \cdot \phi(\rho(\mathbf{x}))) \leq \delta_t$$

holds. Since the error is assumed to be sub-Gaussian, we have

$$P(|f(\mathbf{x}) - \mathbb{E}[p(y|\mathbf{x})]| \leq \beta_t \cdot \phi(\rho(\mathbf{x}))) \leq 2 \cdot \exp\Big(-\frac{\beta_t \cdot \phi(\rho(\mathbf{x}))^2}{2\phi(\rho(\mathbf{x}))^2}\Big).$$

We solve for $\beta_t$:

$$\begin{aligned}\delta_t &= 2\text{exp}(-\frac{\beta_t^2}{2}) \\ \frac{\delta_t}{2} &= \text{exp}(-\frac{\beta_t^2}{2}) \\ \ln(\frac{\delta_t}{2}) &= -\frac{\beta_t^2}{2} \\ \beta_t &= \sqrt{2\ln(\frac{2}{\delta_t})}\end{aligned}$$

□

**Proposition 2** (DisCoMBO is a zero-regret algorithm)**.** *Assume some $f \in \mathcal{H}_k$ with $\mathcal{H}_k$ being a Reproducing Kernel Hilbert Space (RKHS) with squared exponential kernel $k = \phi \circ \rho$ such that $f : \boldsymbol{\mathcal{X}} \to \mathbb{R}$ and $\mathbf{x}^*$ being the optimum of $f$. Further, assume a periodic schedule $s(t) \in \mathbb{R}_{\geq 0}$ that determines whether focus is set on exploitation or exploration, s.t. exploitation periods get longer over time, i.e. $\pi_t = c \cdot \pi_{t-1}$ for some $c > 1$ and period length $\pi_t$. Let the number of bins $K \to \infty$ and $p(y|\mathbf{x})$ be unbiased where $p$ is a PC learned on observations $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. Then, DisCoMBO has sub-linear cumulative regret $R_T$, i.e. $\frac{R_T}{T} \to 0$ with $T \to \infty$.*

*Proof.* In order to prove zero-regret, we have to prove that DisCoMBO (1) has sub-linear regret during exploration, (2) has sub-linear regret during exploitation, and (3) spends almost all iterations in exploitation with $T \to \infty$. We assume that the candidate pool is constructed by sampling from $p(\mathbf{x}|y^*)$ at each iteration.

**DisCoMBO has sub-linear regret during exploration.** We start by showing that $R_T$ is sub-linear during exploration. With $K \to \infty$ and when $s(t) \to 0$, our bucketing mechanism defines a density over DisCo scores that follows a Beta-distribution Beta$(s(t), 1)$. Since $s(t) \to 0$, we concentrate all probability mass at 0, i.e., minimum conformance scores (corresponding to maximizing uncertainty). By Prop. 1, $\rho$ is order-isomorphic to any symmetric kernel, allowing us to assume that there exists $\phi$ such that $k = \phi \circ \rho$ is a squared exponential kernel. We know that for all $\mathbf{x} \in \boldsymbol{\mathcal{X}}$ that

$$f(\mathbf{x}) \leq \mathbb{E}[p(y|\mathbf{x})] + \beta_t \cdot \phi(\rho(\mathbf{x}))$$

holds for all $\mathbf{x}$ with probability $1 - \delta_t$ (see Lemma 1) as well as

$$f(\mathbf{x}^*) \leq \mathbb{E}[p(y|\mathbf{x}^*)] + \beta_t \cdot \phi(\rho(\mathbf{x}^*))$$

holds. Now, let $\boldsymbol{\mathcal{X}}' = \{\mathbf{x} \in \boldsymbol{\mathcal{X}} | \int_{\mathbf{x}} p(\mathbf{x}|y^*) > \epsilon_t\}$ the set of candidates with probability of at least $\epsilon_t$ of being sampled into the candidate pool. Intuitively, this defines an $\epsilon_t$-ball around the best observations. We know that $f$ is $L$-Lipschitz smooth (implied by RKHS assumption). Let $\mathbf{x}_t = \arg\max_{\mathbf{x}' \in \boldsymbol{\mathcal{X}}'} \phi(\rho(\mathbf{x}'))$ be the candidate maximizing DisCo. Then,

$$\begin{aligned}&\mathbb{E}[p(y|\mathbf{x}_t)] + \beta_t \cdot \phi(\rho(\mathbf{x}_t)) \geq \mathbb{E}[p(y|\mathbf{x}^*)] + \beta_t \cdot \phi(\rho(\mathbf{x}^*)) \\ \equiv &\mathbb{E}[p(y|\mathbf{x}_t^*) - L\epsilon_t] + \beta_t \cdot \phi(\rho(\mathbf{x}_t)) \geq \mathbb{E}[p(y|\mathbf{x}^*)] + \beta_t \cdot \phi(\rho(\mathbf{x}^*))\end{aligned}$$

holds with probability $1 - \delta_t$ (Lemma 1). Following Lemma 5.2 from Srinivas et al. [2012], we can conclude that the regret $r_t$ at iteration $t$ is bounded by $L\epsilon_t + 2\beta_t\phi(\rho(\mathbf{x}))$. To obtain the cumulative regret bound, we note that $k$ can be assumed to be a squared exponential kernel; we know that the information gain $\gamma_T$ grows at a rate of $O(\log^{d+1}(T))$ with $d$ being the search space dimension. Also, with growing coverage, we can shrink $\epsilon_t \to 0$ at the same rate. Therefore, we can again follow Srinivas et al. [2012] and conclude that for all $T \geq 1$ we have

$$\sum_{t=1}^{T} r_t^2 = (f(\mathbf{x}^*) - f(\mathbf{x}_t^*) + 2L\epsilon_t) \leq C\beta_T\gamma_T$$

by the Cauchy-Schwarz inequality in the exploration phase.

**DisCoMBO has sub-linear regret during exploitation.** Next, we need to show that the sub-linear regret property also holds for the exploitation phase, i.e., when $s(t) \to \max(s(t))$. Assuming we can choose a sufficiently high amplitude for $s(t)$, our Beta-distribution Beta$(s(t), 1)$ concentrates

all of the probability mass at 1, i.e., maximum conformance scores (corresponding to minimum uncertainty). Since we then only sample $\mathbf{x} \in \boldsymbol{\mathcal{X}}$ with $\rho(\mathbf{x}) \to 1$ from $p(\mathbf{x}|y^*)$, this amounts to minimizing uncertainty near the current incumbent. Since $f$ is L-Lipschitz smooth (a direct consequence from the RKHS), with long enough exploration we cover enough space s.t.

$$f(\mathbf{x}^*) \leq f(\mathbf{x}_t^*) + L\epsilon$$

holds for some incumbent $\mathbf{x}_t^*$ at $t$ and some $\epsilon > 0$. Note that we can assume that we explore "enough" for this condition to hold since we switch infinitely often between exploration and exploitation. Since we know that

$$\mathbb{E}[p_t(y|\mathbf{x}_t^*)] \geq \mathbb{E}[p_t(y|\mathbf{x}^*)]$$

holds at $t$ (i.e., the model belief is that $\mathbf{x}_t^*$ is the optimum) and since $\beta_t \cdot \phi(\rho(\mathbf{x})) \to 0$ for all $\mathbf{x}$ as $t \to \infty$, we can conclude that

$$\mathbb{E}[p_t(y|\mathbf{x}_t^*)] + \beta_t \cdot \phi(\rho(\mathbf{x}_t^*)) \geq \mathbb{E}[p_t(y|\mathbf{x}^*)]\beta_t \cdot \phi(\rho(\mathbf{x}^*)).$$

As before, following Srinivas et al. [2012] implies that

$$\sum_{t=1}^{T} r_t^2 = (f(\mathbf{x}^*) - f(\mathbf{x}_t^*) + 2L\epsilon_t) \leq C\beta_T\gamma_T$$

as $T \to \infty$ by Cauchy-Schwarz inequality.

**DisCoMBO spends most time exploiting.** Finally, we show that DisCoMBO spends most of its time in exploitation. Since the period length is increased every iteration, $\pi_t \to \infty$ as $T \to \infty$. This shows that the exploitation period gets infinitely long.

Combining these three results, we see that

$$\frac{\sqrt{\frac{1}{2} \cdot 2 \cdot TC\beta_T\gamma_T}}{T} \to 0 \quad \text{with} \quad T \to \infty.$$

□

**Interpretation of DisCoMBO's Zero-Regret Property.** The zero-regret property of DisCoMBO shows that our algorithm ultimately finds the best solution in the limit of infinite compute budget and does not waste exploration budget in unpromising regions. However, note that it does not provide any finite-time convergence properties. Also, note that the exploration-exploitation trade-off in our current implementation is schedule-driven. While our proof considers periodic schedules, other ways to implement the exploration-exploitation trade-off are of course possible. For example, DisCoMBO can easily be adapted to fit into the classical acquisition function-based BO loop by dropping the schedule and using the DisCo score as an uncertainty signal, guiding the acquisition function's selection. However, our schedule-driven trade-off works similarly to a stochastic UCB version: The fact that we construct the candidate pool based on $p(\mathbf{x}|y^*)$ acts similarly to the mean-term in UCB, while maximizing/minimizing DisCo acts similarly to the exploration-exploitation trade-off in UCB. Empirically, we show in 3 that this strategy achieves performances on par with state-of-the-art BO algorithms, demonstrating that DisCo provides a useful uncertainty estimate for BO tasks.

# C Experimental Details

Here, we present additional details of our empirical evaluation. The raw logs from our experiments are available at `https://doi.org/10.6084/m9.figshare.32183937`, and our code is available at `https://github.com/bewit/DisCoMBO`.

## C.1 Benchmarks

Benchmarks serve as a critical component for the systematic evaluation of SMBO algorithms, providing a reproducible and computationally efficient framework for assessing performance in high-stakes or resource-intensive domains. A task within a benchmark is defined by a search space $\boldsymbol{\mathcal{X}}$ and an objective function $f : \boldsymbol{\mathcal{X}} \to \mathbb{R}$, such as maximizing annual energy production in wind farm layout design or minimizing validation error in hyperparameter optimization. To bypass the

prohibitive costs and latency of real-world evaluations, benchmarks typically utilize three primary representations: tabular benchmarks, where a discrete subset of the search space is pre-evaluated and stored; surrogate benchmarks, which employ regression models to interpolate between observed points and enable optimization over continuous parameters; and simulation benchmarks, which leverage low-fidelity numerical proxies to capture complex system dynamics. By abstracting away the execution costs of physical experiments or large-scale model training, these benchmarks facilitate standardized, large-scale comparisons across diverse SMBO strategies under consistent budget and noise constraints.

In our experimental evaluation, we used the following benchmarks: JAHS [Bansal et al., 2022], NAS-Bench-201 [Dong and Yang, 2020], HPO-B [Pineda-Arango et al., 2021], crossed barrel [Gongora et al., 2020], and Floris [Fleming et al., 2020]. We now briefly describe the characteristics of these benchmarks.

**JAHS.** JAHS aims to provide a reproducible benchmark for joint optimization of neural architectures and other hyperparameters on real-world tasks. It consists of a 14-dimensional search space (of which some dimensions, like the number of epochs, can also be treated as fidelities). The search space contains both discrete and continuous hyperparameters. Regarding tasks, JAHS provides training and evaluation statistics of models trained on CIFAR-10, Fashion-MNIST and Colorectal Histology. It further provides runtime statistics for each trained configuration (i.e., architecture and hyperparameter settings) in terms of the wall clock time. Since all configurations were trained on the same hardware, wall clock time serves as a proxy for how expensive a given configuration is. For a fair evaluation, we follow standard practice from the AutoML community and report regrets against wall clock time for each algorithm. All tasks are image classification tasks. See [Bansal et al., 2022] for more information.

**NAS-Bench-201.** NAS-Bench-201 provides a reproducible benchmark for neural architecture search. It comes with a dense encoding of architectures, leading to a 6-dimensional search space. NAS-Bench-201 provides validation accuracy, test accuracy and training time statistics for each architecture on the CIFAR-10, CIFAR-100, and Imagenet datasets. Like for JAHS, we follow standard practices from the AutoML community and report regret against wall clock time for a fair comparison of methods. In our experiments, we only used CIFAR-10. See [Dong and Yang, 2020] for more information.

**HPO-B.** HPO-B is a large-scale HPO benchmark based on a diverse set of OpenML tasks. It comes with 176 search spaces and 196 datasets. Since some search spaces have been evaluated on multiple datasets, many tasks are available. Most of the tasks are classification and regression tasks on different data modalities, such as tabular data or image data. In our evaluation, we used the credit-g and vehicle datasets (dataset IDs 31 and 9914), paired with search spaces over hyperparameters of a random search model (search space ID 6794) and a gradient boosting model (search space ID 6767). Both datasets are tabular datasets. We chose these since both provide many evaluations reported at OpenML (506k for credit-g and 31k for vehicle), allowing rigorous comparability. The search spaces we used contained discrete and continuous hyperparameters. The gradient boosting search space is defined over 17 hyperparameters, while the random forest search space is defined over 9 hyperparameters. For more information, refer to [Pineda-Arango et al., 2021].

**PV.** The crossed barrel benchmark is a mechanical design task where the objective is to maximize the toughness (energy absorption per unit volume) of a 3D-printed architected material. The structure consists of two parallel platforms connected by $n$ hollow, thin-walled columns. The geometry of these columns is defined by their outer radius ($r$), thickness ($t$), and a twist angle ($\theta$) between the platforms, which creates a non-linear, reentrant geometry. Performance is evaluated through physical mechanical testing, where the structure is compressed to measure its force-displacement response. The task is characterized by a complex, non-convex objective landscape where the highest toughness results from a balance between high strength and high ductility. See [Gongora et al., 2020] for more details.

**Floris.** The Floris benchmark evaluates optimization performance in the design and control of wind power plants. The primary objective is to maximize the Annual Energy Production (AEP) or net revenue of a wind farm by mitigating wake-induced power losses. The design space includes the spatial coordinates of $N$ turbines (layout optimization) and their operational setpoints, such as yaw misalignment angles and induction factors (wake steering and control). The benchmark employs steady-state engineering wake models to simulate the complex, non-linear aerodynamic interactions where upstream turbines reduce the wind speed and increase turbulence for downstream

units. This task is characterized by high-dimensional search spaces, significant multimodality due to the nonlinear nature of wake deficits, and the requirement to integrate over diverse wind roses (stochastic wind speeds and directions). See [Fleming et al., 2020] for more information.

### C.2 Search Space Extension of JAHS

To make the HPO problem on JAHS more challenging, we decided to extend the search space slightly as JAHS – as a surrogate benchmark – allows us to query hyperparameter values which were not tested explicitly in the benchmark. We defined three search spaces for JAHS which are presented in the following table.

Table 1: **JAHS Search Space.** We define three versions of the JAHS search space, ranging from simpler to harder spaces.

| | S1 | S2 | S3 |
|---|---|---|---|
| Activation | [Mish, ReLU, Hardswish] | [Mish, ReLU, Hardswish] | [Mish, ReLU, Hardswish] |
| Learning Rate | [1e-3, 1e0] | [1e-3, 1e0] | [1e-3, 1e0] |
| Weight Decay | [1e-5, 1e-2] | [1e-5, 1e-2] | [1e-5, 1e-2] |
| Trivial Augment | [True, False] | [True, False] | [True, False] |
| Op1 | 0-6 | 0-6 | 0-6 |
| Op2 | 0-6 | 0-6 | 0-6 |
| Op3 | 0-6 | 0-6 | 0-6 |
| Op4 | 0-6 | 0-6 | 0-6 |
| Op5 | 0-6 | 0-6 | 0-6 |
| Op6 | 0-6 | 0-6 | 0-6 |
| N | 1-15 | 1-11 | 1-5 |
| W | 1-31 | 1-23 | 1-16 |
| Epoch | 1-200 | 1-200 | 1-200 |
| Resolution | 0-1 | 0-1 | 0-1 |

### C.3 External Knowledge

To evaluate the robustness and adaptability of DisCoMBO, we define a structured suite of external knowledge profiles designed to simulate various expert-AI interaction scenarios. These profiles consist of both beneficial and misleading priors, characterized as probability distributions over the search space that favor specific configurations. A beneficial prior is constructed by identifying the best-performing configuration $\mathbf{x}^+$ from a random sample of $10^4$ evaluations and assigning it a sampling probability up to $1000$ times higher than the baseline for a subset of dimensions. Conversely, a misleading prior is centered on the worst-performing configuration $\mathbf{x}^-$. We emphasize strong priors to rigorously test the recovery mechanisms of DisCoMBO against baseline methods like $\pi$BO and DynaBO, which inherently favor high-strength priors in their selection policies. Our experimental design systematically explores four distinct interaction regimes: (1) sparse beneficial knowledge provided in the initial optimization stages to assess early-stage acceleration; (2) sparse beneficial knowledge introduced in later iterations to evaluate late-stage integration; (3) dense misleading knowledge covering a large majority of dimensions to challenge the algorithm's ability to recover from deceptive guidance; and (4) a temporal sequence of contradictory priors that alternates between beneficial and misleading information. In the sparse scenarios, the prior influences fewer than 25% of the total parameters, testing whether the model can effectively leverage localized information. In the misleading and sequential scenarios, we increase the dimensionality of the prior to demonstrate that DisCoMBO can reliably filter high-strength, low-accuracy information even when it dominates the input signal, thereby approximating the noisy and potentially inconsistent feedback loops found in real-world expert-in-the-loop systems.

**JAHS.** The following JSON code shows the priors given in our JAHS experiments.


```
[
    {
        "type": "bad",
        "intervention": {"Activation": 1, "LearningRate": 0.8201676371308472, "N": 15,
        "Op1": 3, "Op2": 4, "Op3": 1, "Op4": 2, "Resolution": 0.5096959403985494,
        "TrivialAugment": 0, "W": 14,
         "WeightDecay": 0.002697686639935806, "epoch": 10},
        "iteration": 5
    },
    {
        "type": "good",
        "intervention": {"N": 3, "W": 16, "Resolution": 1},
        "iteration": 15
    },
    {
        "type": "good",
        "intervention": null,
        "iteration": 20
    },
    {
        "type": "good",
        "kind": "dist",
        "intervention": {"N": {"dist": "cat", "parameters":
        [1, 1, 1, 1e4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]},
        "W": {"dist": "cat", "parameters":
        [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1e4]},
        "Resolution": {"dist": "uniform", "parameters": [0.98, 1.02]}},
        "iteration": 5
    }
]
```


**NAS-Bench-201.** The following JSON code shows the priors provided in our experiments on NAS-Bench-201.


```
[
    {
```

```
        "type": "good",
        "kind": "point",
        "intervention": {"Op_0": 2, "Op_1": 2, "Op_2": 0},
        "iteration": 5
    },
    {
        "type": "bad",
        "kind": "point",
        "intervention": {"Op_0": 1, "Op_1": 2, "Op_2": 1},
        "iteration": 5
    },
    {

        "type": "good",
        "kind": "point",
        "intervention": null,
        "iteration": 20
    },
    {
        "type": "good",
        "kind": "dist",
        "intervention": {"Op_0": {"dist": "cat", "parameters": [1, 1, 1e4, 1, 1]},
                         "Op_1": {"dist": "cat", "parameters": [1, 1, 1e4, 1, 1]},
                         "Op_2": {"dist": "cat", "parameters": [1e4, 1, 1, 1, 1]}},
        "iteration": 5
    }
]
```

**HPO-B.** The following shows the priors defined for our HPO-B experiments. Each of the three priors corresponds to one of the three tasks we tested from HPO-B. The order is 6767:31, 6794:31, 6794:9914. Note that we only applied beneficial priors in the case of HPO-B.

```
    {
        "type": "good",
        "kind": "dist",
        "intervention": {
            "eta": {"dist": "uniform", "parameters": [0.45, 0.55]},
            "subsample": {"dist": "uniform", "parameters": [0.55, 0.65]},
            "lambda": {"dist": "uniform", "parameters": [-150, 0]},
            "min_child_weight": {"dist": "uniform", "parameters": [-15, -10]}},
        "iteration": 1
    },
    {
        "type": "good",
        "kind": "dist",
        "intervention": {
            "num_trees": {"dist": "int_uniform", "parameters": [1690, 1720]},
            "mtry": {"dist": "int_uniform", "parameters": [30, 34]},
            "min_node_size": {"dist": "int_uniform", "parameters": [780, 800]}
        },
        "iteration": 1
    },
    {
        "type": "good",
        "kind": "dist",
        "intervention": {
            "num_trees": {"dist": "int_uniform", "parameters": [1690, 1720]},
            "mtry": {"dist": "int_uniform", "parameters": [280, 320]},
            "sample_fraction": {"dist": "uniform", "parameters": [0.5, 0.53]}
        },
```

```
    "iteration": 1
}
```


**PV.** The following shows the interactions defined for the task considered on crossed barrel.


```
    {
    "type": "good",
    "kind": "point",
    "intervention": {"theta": 142.46},
    "iteration": 5,
    "task": "crossed_barrel"
},
{
    "type": "bad",
    "kind": "point",
    "intervention": {"theta": 32.11, "n": 11.91, "r": 2.48},
    "iteration": 5,
    "task": "crossed_barrel"
},
{
    "type": "good",
    "kind": "dist",
    "intervention": {"theta": {"dist": "uniform", "parameters": [141, 144]}},
    "iteration": 5,
    "task": "crossed_barrel"
},


{
    "type": "good",
    "kind": "point",
    "intervention": {"theta": 142.46},
    "iteration": 15,
    "task": "crossed_barrel"
},
{
    "type": "bad",
    "kind": "point",
    "intervention": {"theta": 33.11, "n": 12.01, "r": 2.58},
    "iteration": 25,
    "task": "crossed_barrel"
},
{
    "type": "good",
    "kind": "point",
    "intervention": {"theta": 141.96},
    "iteration": 35,
    "task": "crossed_barrel"
}
```


**Floris.** The following shows the interactions defined for the task considered on Floris.


```
{
    "type": "good",
    "kind": "point",
    "intervention": {
        "yaw_t000_deg": -10.83748690531058,
        "yaw_t001_deg": 11.472652376984005,
        "yaw_t002_deg": 13.89467307805834
    },
```

```
    "iteration": 5
},
{
    "type": "bad",
    "kind": "point",
    "intervention": {
        "yaw_t000_deg": -27.275959662325022,
        "yaw_t001_deg": 1.4292151641649902,
        "yaw_t002_deg": 26.63570488559511,
        "yaw_t003_deg": -25.844124879037324,
        "yaw_t004_deg": -18.74817789106919,
        "yaw_t005_deg": -29.885699970885156,
        "yaw_t006_deg": 9.70821606188883,
        "yaw_t007_deg": -27.2264816726402,
        "yaw_t008_deg": -21.44977989395875,
        "yaw_t009_deg": 23.53069799422112,
        "yaw_t010_deg": -8.328123875812214,
        "yaw_t011_deg": -29.022544007776677,
        "yaw_t012_deg": -15.702255207610008,
        "yaw_t013_deg": -19.815213890373023,
        "yaw_t014_deg": 23.418886185782853,
        "yaw_t015_deg": -1.0822319543582601,
        "yaw_t016_deg": 3.496271047466287
    },
    "iteration": 5
},
{
    "type": "good",
    "kind": "dist",
    "intervention": {
                     "yaw_t000_deg":
                     {"dist": "uniform", "parameters": [-11, -10]},
                     "yaw_t001_deg":
                     {"dist": "uniform", "parameters": [10, 12]},
                     "yaw_t002_deg":
                     {"dist": "uniform", "parameters": [12, 14]}
                    },
    "iteration": 5
},

{
    "type": "good",
    "kind": "point",
    "intervention": {
        "yaw_t000_deg": -10.83748690531058,
        "yaw_t001_deg": 11.472652376984005,
        "yaw_t002_deg": 13.89467307805834
    },
    "iteration": 15
},
{
    "type": "bad",
    "kind": "point",
    "intervention": {
        "yaw_t000_deg": -28.275959662325022,
        "yaw_t001_deg": 1.6292151641649902,
        "yaw_t002_deg": 27.53570488559511,
        "yaw_t003_deg": -25.844124879037324,
        "yaw_t004_deg": -18.74817789106919,
```

```
            "yaw_t005_deg": -29.885699970885156,
            "yaw_t006_deg": 9.70821606188883,
            "yaw_t007_deg": -27.2264816726402,
            "yaw_t008_deg": -21.44977989395875,
            "yaw_t009_deg": 23.53069799422112,
            "yaw_t010_deg": -8.328123875812214,
            "yaw_t011_deg": -29.022544007776677,
            "yaw_t012_deg": -15.702255207610008,
            "yaw_t013_deg": -19.815213890373023,
            "yaw_t014_deg": 23.418886185782853,
            "yaw_t015_deg": -1.0822319543582601,
            "yaw_t016_deg": 3.496271047466287
        },
        "iteration": 25
    },
    {
        "type": "good",
        "kind": "point",
        "intervention": {
            "yaw_t000_deg": -11.23748690531058,
            "yaw_t001_deg": 12.072652376984005,
            "yaw_t002_deg": 14.09467307805834
        },
        "iteration": 35
    }
```


### C.4 Further Results & Ablations

In this section, we describe further results we obtained during our experimental evaluation.

**Regret Analysis** To examine the performance of DisCoMBO, we provide plots showing the normalized regret over time (either cumulative wall-clock time if provided by benchmarks or iterations).

Fig. 7 compares DisCoMBO (—) with all our baselines in the default SMBO setting, that is, without any external priors to guide optimization. We find that DisCoMBO ranks first on **5/9** tasks, is competitive to top-tier baselines on **3/9** tasks, and only falls slightly short on one task (Floris). Overall, we find that DisCoMBO effectively identifies high-performing solutions. Notably, we achieve our goal of improving upon purely generative PC-based SMBO (IBO-HPC; —) on **9/9** tasks.

Fig. 9 compares the regret over time between DisCoMBO (- · -) and our baselines $\pi$Bo, IBO-HPC, and DynaBO when a beneficial prior is provided after 5 iterations. We find that DisCoMBO outperforms or matches all baselines on **7/9** tasks, demonstrating that it successfully retains or even improves upon the efficacy of incorporating priors of our baselines, especially IBO-HPC.

Fig. 11 demonstrates that DisCoMBO (- -) successfully recovers from misleading priors on all tasks. In **4/6** tasks, we recover faster than our baselines, demonstrating robustness. We evaluate recovery on JAHS, NAS-Bench-201, Floris, and PV as a representative subset of tasks.

Fig. 13 demonstrates that DisCoMBO (- -) effectively handles sequences of priors, even if they contain contradicting information. We first provide a misleading prior, followed by a beneficial prior, followed by a misleading prior, and finally a beneficial prior again. It can be observed that DisCoMBO leverages beneficial priors effectively while not suffering from misleading ones. Priors are provided at iterations 5, 15, 25, and 35. Note that $\pi$BO is not considered here as it assumes that priors are available before optimization.

**Ranking Analysis** Another key metric when analyzing the performance of an SMBO algorithm is to consider the average rank over time. This shows at which phase algorithms excel and in which phases they flatten out.

Fig. 15 shows the average rank of each algorithm over time across 50 seeds when no prior is provided. We find that in terms of final rank, DisCoMBO (—) wins on **6/9** tasks while being competitive to

top-tier baselines on the remaining tasks except Floris (see App. C.4 for more information). On most tasks, DisCoMBO seems to find good solutions quickly as it ranks high early on all AutoML tasks. For PV (crossed-barrel), DisCoMBO seems to benefit from exploration, indicating that our exploration samples diverse candidates, helping DisCoMBO to exploit this knowledge effectively in later iterations. This confirms the effectiveness of DisCoMBO across our diverse task suite.

Fig. 16 shows the average rank of each algorithm over time across 50 seeds when a beneficial prior is provided after 5 iterations. We find that in terms of final rank, DisCoMBO (—) wins on **7/9** tasks. On all tasks except Floris, DisCoMBO seems to effectively leverage provided priors. This confirms that DisCoMBO retains, or even improves, the efficacy of IBO-HPC in incorporating provided priors across our diverse task suite.

Fig. 17 shows that DisCoMBO (—) achieves the top final rank on **4/6** tasks. Across all benchmarks except Floris and PV (crossed-barrel), the framework successfully recovers from a misleading prior given after 5 iterations, demonstrating that DisCoMBO retains or enhances the targeted knowledge integration capabilities of purely generative approaches like IBO-HPC. Similar results can be obtained in Fig. 18 when a sequence of priors is provided.

**Relative Improvement** Since one of the key goals of DisCoMBO is to improve upon existing PC-based SMBO to close the gap between PC-based and traditional SMBO, we show the relative improvement of DisCoMBO and our baselines against IBO-HPC. Fig. 19, 20, 21, and 22 show that DisCoMBO achieves significant improvement across all tasks when no prior is provided. If priors are provided, regardless of whether they are beneficial, misleading, or both, DisCoMBO improves upon IBO-HPC and most baselines in the majority of tasks. This demonstrates that DisCo effectively serves as an uncertainty measure, effectively guiding our exploration and exploitation, and it shows that DisCo handles priors well in various settings.

**Critical Difference Analysis** Our critical difference diagrams (see Fig. 8, 10, 12, and 14 reveal that DisCoMBO ranks first throughout all settings, namely: Standard SMBO (without priors), SMBO with beneficial priors, SMBO with misleading priors, and SMBO with multiple, contradicting priors. Especially noticeable is that DisCoMBO is statistically indistinguishable from established state-of-the-art BO baselines like HEBO. This demonstrates the efficacy of our uncertainty-guided exploration-exploitation strategy as well as the robustness and flexibility of DisCoMBO in diverse settings. We used the Wilcoxon signed-rank test at $p = 0.05$ with a multiple-testing correction.

**Ablation and Sensitivity Analysis of Hyperparameters** Since DisCoMBO has a few hyperparameters as well, we provide an ablation on the three most critical ones ($\beta$, $n$, and $c$). $\beta$ is a tuple and controls the level of exploration and exploitation. High $\beta$ values mean high exploration, while $\beta \to 0$ means full exploitation. $c$ controls the cycle length of each exploration and exploitation phase. The parameter $n$ controls the number of candidate samples generated, which are then rated by DisCo and from which one is selected for evaluation.

Fig. 23 shows 9 different settings of these parameters. It can be seen that DisCoMBO is robust to hyperparameter choices, demonstrating that it achieves high performance even with non-optimal hyperparameter settings.

**Misleading Prior Detection** As illustrated by the ROC AUC curves in Fig. 24, we evaluate the isolating performance of DisCoMBO's (—) prior detection layer against DynaBO's (- - ) heuristic safeguard after 5, 10, and 15 optimization iterations. Empirically, DisCoMBO consistently outperforms DynaBO across all **5/5** benchmark tasks, demonstrating a robust capability to reliably discriminate between beneficial and misleading external guidance in **4/5** cases. The singular exception to this strong discriminative performance occurs on the Floris task. Because the underlying PC density estimator occasionally struggles to model the unique and highly specialized topological properties of the Floris objective landscape, the resulting conformance scores become less distinctive, consequently degrading detection accuracy in this specific domain (see App. C.4 for an extended discussion).

**DisCoMBO on JAHS** DisCoMBO exhibits exceptionally strong performance on the JAHS benchmark (see Fig. 7). We attribute this efficacy to the inductive biases of Mixed Sum-Product Networks (MSPNs), the specific Probabilistic Circuit parameterization utilized in our evaluation. MSPNs are

structurally engineered to handle heterogeneous domains by natively integrating continuous and discrete random variables without requiring artificial encodings or continuous relaxations.

This architectural alignment allows DisCoMBO to accurately capture the complex joint distribution and search space semantics inherent to mixed-variable domains. Consequently, the learned surrogate maintains high fidelity across the structural and numerical boundaries of the JAHS search space, facilitating a highly efficient traversal of the optimization landscape.

**DisCoMBO on Floris** On the Floris benchmark, DisCoMBO yields significant performance gains over the purely generative IBO-HPC baseline but does not fully close the gap to state-of-the-art discriminative baselines (see Fig. 7). This behavior is fundamentally rooted in the architectural contrast between joint density estimators and specialized regression models, as well as in the properties of the search space. Floris' search space is a fully continuous search space governed by smooth target topologies. While models explicitly designed for regression naturally exploit surface continuity, Probabilistic Circuits natively function as discrete latent variable models. Consequently, standard PC parameterizations tend to partition continuous manifolds into coarse, localized densities, introducing a discretization bias that risks omitting high-performing regions during generative candidate sampling.

Despite this architectural constraint, DisCoMBO consistently outpaces IBO-HPC. We argue that this resilience stems directly from the formal uncertainty semantics introduced by the DisCo score. Because DisCo behaves analogously to symmetric kernels on continuous domains, it effectively injects a smoothness bias into the framework's uncertainty estimates. Even when the underlying PC density representation lacks continuous regression accuracy, the kernel-like properties of DisCo provide a well-calibrated exploration-exploitation signal. This robust mechanism mitigates the coarse discretization of the surrogate, allowing DisCoMBO to navigate smooth continuous topologies far more effectively than heuristics driven by pure likelihood sampling.

**DynaBO on JAHS** During our experiments, we observed that DynaBO [Fehring et al., 2026] occasionally struggled to converge on specific landscapes, most notably on JAHS (CIFAR-10) (see Fig. 9, 11, and 13). Although we expected DynaBO's performance to closely parallel that of $\pi$BO, this discrepancy prompted us to consult directly with the authors of the original framework. Despite a collaborative review of the implementation, the precise root cause underlying this localized behavior could not be definitively isolated or resolved. Given that DynaBO performs as expected across the other evaluated benchmarks and given that we used the implementation of Fehring et al. [2026], we suspect this anomaly stems either from an undiscovered misconfiguration unique to the JAHS environment or from a subtle interplay between DynaBO's safeguard mechanisms and the specific topology of the JAHS search space.

**DisCoMBO with broader priors** To provide a fair and comparable analysis, the prior definitions in our main results reused the prior definitions of Seng et al. [2025]. Since these priors are quite narrow, we also provide a speed-up analysis for broader priors. It can be seen that also under broader priors, DisCoMBO clearly benefits from the provided priors (see Tab. 2). The broader priors were defined by doubling the variance of the prior for continuous variables compared to the priors in App. C.3. For discrete parameters, the configurations were upweighted by a factor of 3 instead of $1e^4$.

**DisCoMBO vs. GP+EI** To strengthen our results further, we add the final regret of GP+EI and compare it against DisCoMBO. Tab. 3 shows that DisCoMBO outperforms GP+EI in **6/8** cases, demonstrating its effectiveness on various tasks. NASBench201 was excluded due to time constraints (GP+EI runs did not finish within a reasonable time).

**DisCo in UCB acquisition function** We can use DisCo in a UCB acquisition function by setting $\mu(x)$ as the standardized expected conditional performance std($\mathbb{E}[y|x]$) for a configuration $x$ and $\sigma(x)$ as the $\lambda$-scaled inverse DisCo score, i.e. $\mathrm{UCB}_{\mathrm{DisCo}}(x) = \mu(x) + \lambda(1 - \rho(x))$. We used UCB to score $N$ samples from $p(x|y^*)$ and $M$ samples randomly drawn from the search space, and selected the $K$ highest values (with $K$ = $L$ = 5 as reported in App. C.5). Tab. 4 shows that the UCB-version is on-par with our originally reported results, which is expected given that the acquisition function is directly derived from our underlying model and the DisCo score. We suspect that this perspective may ease the integration with existing BO frameworks. However, please note that careful tuning of $\lambda$ and extensive analysis of the acquisition function's properties is left for future work.

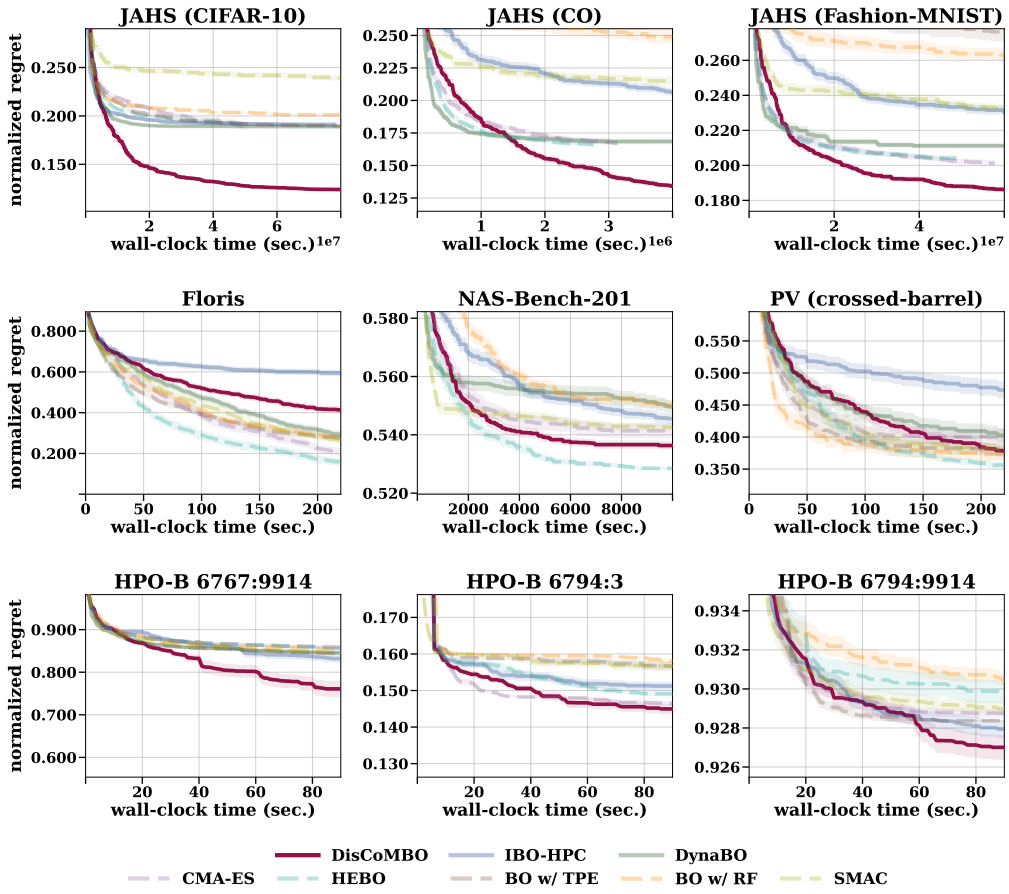



Figure 7: **DisCoMBO achieves top-tier performance without external priors.** Without guidance with external priors, DisCoMBO (—) achieves top-tier performance on **8/9** tasks. On Floris, we still significantly improve upon IBO-HPC (—), however, DisCoMBO cannot fully catch up with top-tier baselines (see App. C.4) for further discussion.

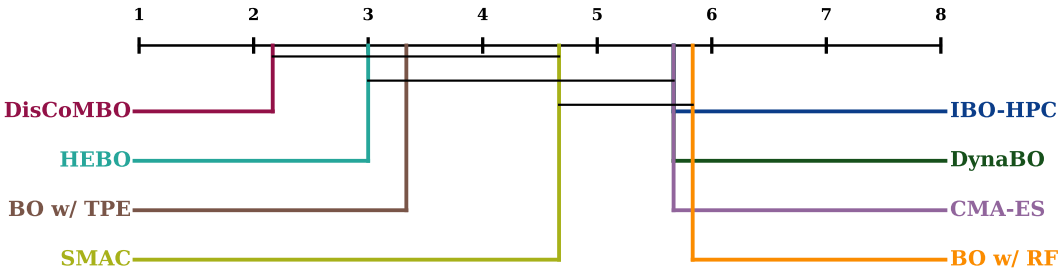



Figure 8: **DisCoMBO is competitive to top-tier SMBO methods without priors.** DisCoMBO ranks **first** across 9 tasks when **no prior** is given, being statistically indistinguishable from strong baselines like HEBO. DisCoMBO significantly outperforms IBO-HPC, its PC-based counterpart, lacking a principled exploitation-exploration trade-off. Assessment is based on Wilcoxon signed-rank test ($p = 0.05$), corrected for multiple tests.

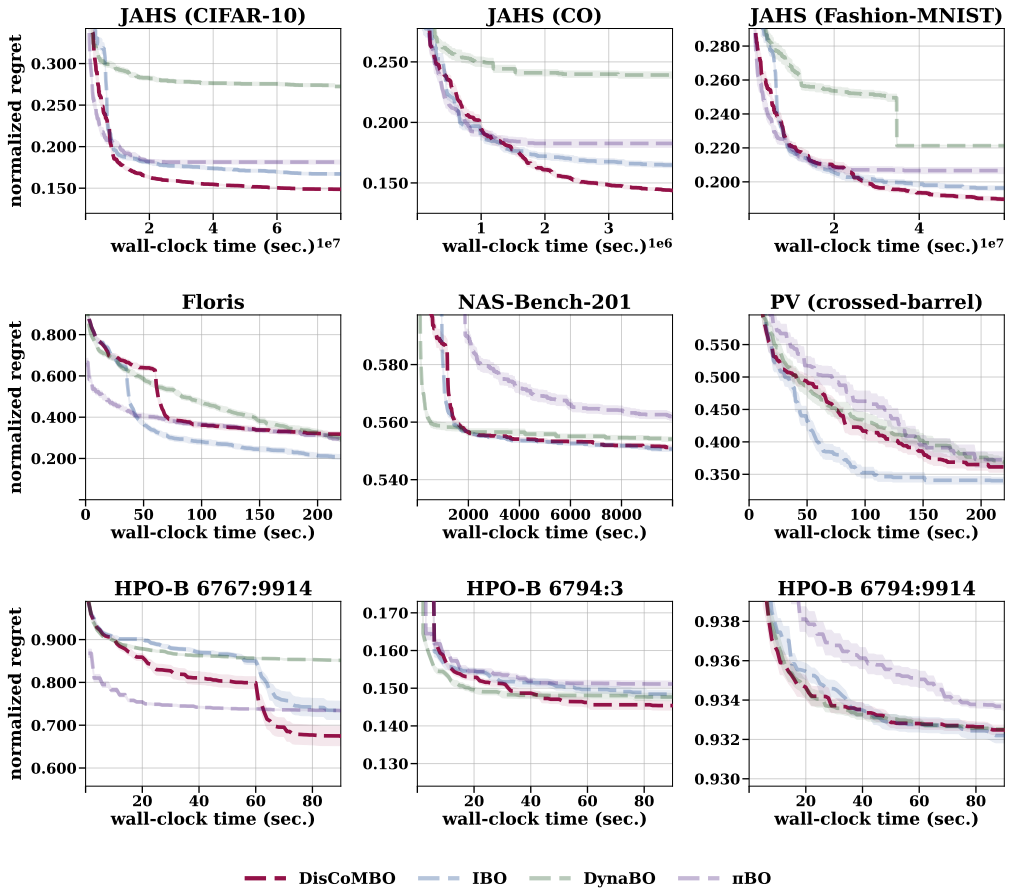



Figure 9: **DisCoMBO retains knowledge integration efficacy of generative PC-based SMBO.** When provided with external priors early in optimization, DisCoMBO (- -) retains the efficacy of IBO-HPC (-·-) in leveraging that provided knowledge to converge faster and to better solutions. We outperform baselines on **7/9** tasks.

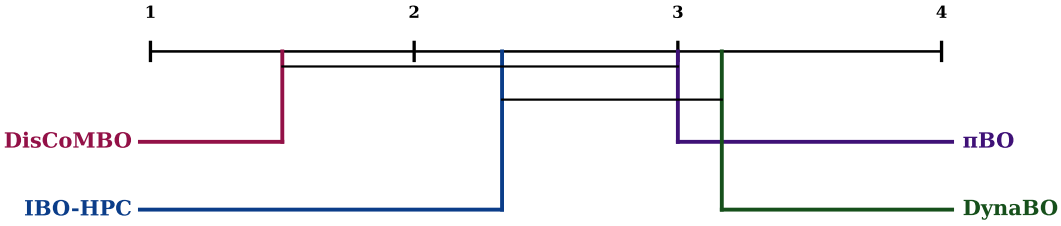



Figure 10: **DisCoMBO effectively incorporates priors into optimization.** DisCoMBO ranks **first** across 9 tasks when a prior is given, being statistically indistinguishable to IBO-HPC and $\pi$BO. This also demonstrates that DisCoMBO retains the efficacy of IBO-HPC in integrating external priors into the optimization process. Assessment is based on Wilcoxon signed-rank test ($p = 0.05$) and corrected for multiple tests.

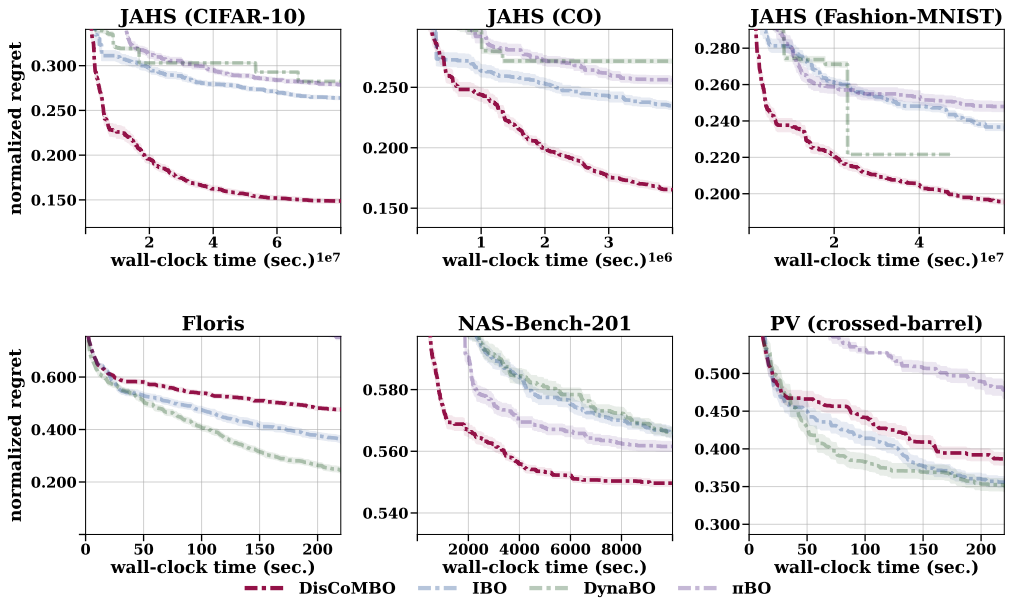



Figure 11: **DisCoMBO effectively recovers from misleading priors.** DisCoMBO (- -) successfully recovers from misleading priors on all tasks. In **4/6** tasks, we recover faster than our baselines, demonstrating robustness. We evaluate recovery on JAHS, NAS-Bench-201, Floris, and PV as a representative subset of tasks. On Floris, $\pi$BO was not able to recover from the misleading prior.

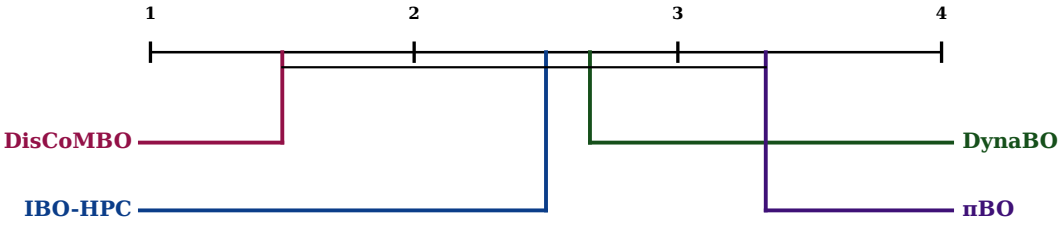



Figure 12: **DisCoMBO effectively recovers from misleading priors.** DisCoMBO ranks **first** across 9 tasks when **a misleading** prior is given. While DisCoMBO is statistically indistinguishable to IBO-HPC and DynaBO, the results show that DisCoMBO retains or even improves upon the robustness of all competing methods. Assessment is based on Wilcoxon signed-rank test ($p = 0.05$) and corrected for multiple tests.

Table 2: **DisCoMBO is effective with broader priors.** When priors are not as specific as in our main results, DisCoMBO still clearly benefits from provided priors in terms of average speed-up. A value $> 1.0$ means the algorithm was $(x - 1.0) \cdot 100\%$ faster compared to DisCoMBO with no prior provided.

| Benchmark/task | DisCoMBO (no interaction) | DisCoMBO (broad prior) |
|---|---|---|
| JAHS/cifar10 | 1.0000 | 2.1603 |
| JAHS/colorectal_histology | 1.0000 | 1.9342 |
| JAHS/fashion_mnist | 1.0000 | 1.7872 |
| HPO-B/6767:31 | 1.0000 | 2.7492 |
| HPO-B/6767:9914 | 1.0000 | 2.1132 |
| HPO-B/6794:3 | 1.0000 | 1.7865 |
| HPO-B/6794:31 | 1.0000 | 1.8305 |
| HPO-B/6794:9914 | 1.0000 | 1.1544 |
| HPO-B/6794:10101 | 1.0000 | 1.5094 |

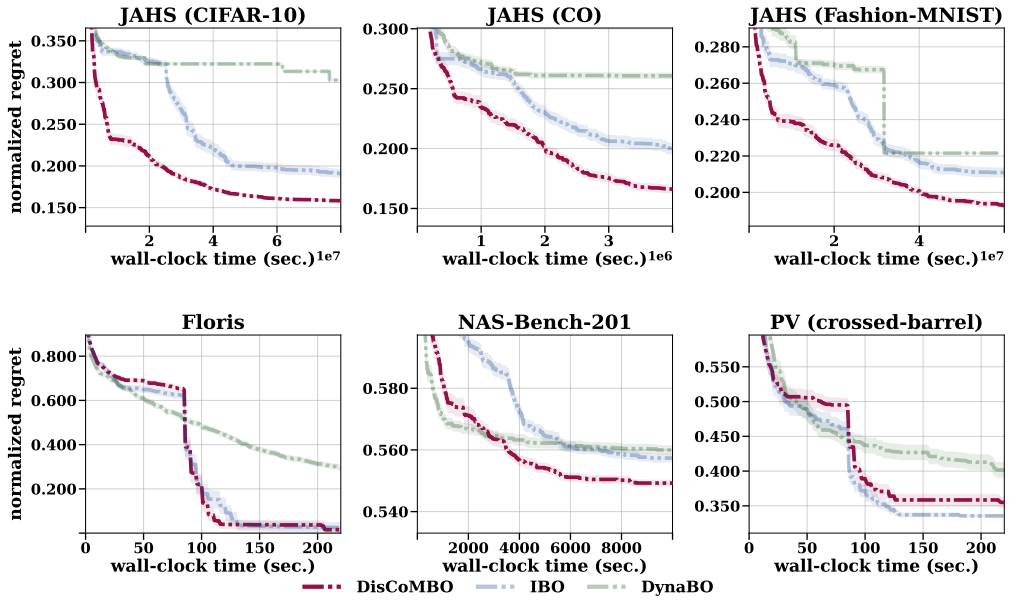



Figure 13: **DisCoMBO effectively handles sequences of priors.** We demonstrate that DisCoMBO (- - ) effectively handles sequences of priors, even if they contain contradicting information. We first provide a misleading prior, followed by a beneficial prior, followed by a misleading prior, and finally a beneficial prior again. It can be observed that DisCoMBO leverages beneficial priors effectively while not suffering from misleading ones. $\pi$BO does not allow for multiple priors.

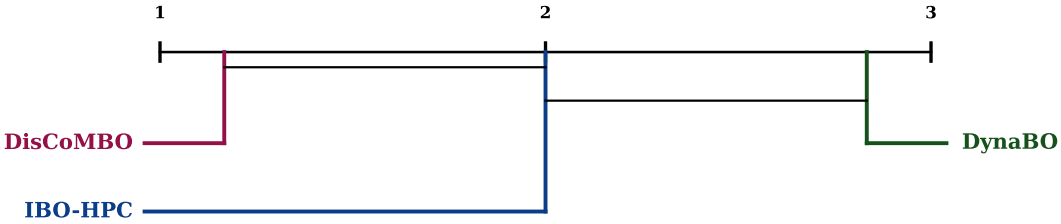



Figure 14: **DisCoMBO effectively handles sequences of priors.** DisCoMBO ranks **first** across 9 tasks when **a sequence** of priors is given. This demonstrates that DisCoMBO successfully handles multiple, even contradicting priors. Assessment is based on Wilcoxon signed-rank tests ($p = 0.05$) and corrected for multiple tests.

Table 3: **DisCoMBO outperforms GP+EI in 6/8 tasks.** Besides HEBO, we also compare the final regret of GP+EI against DisCoMBO. It can be seen that DisCoMBO outperforms GP+EI in 6/8 cases, demonstrating its effectiveness on various tasks. NASBench201 was excluded due to time constraints.

| Task | DisCoMBO | GP+EI |
|---|---|---|
| JAHS/cifar10 | **0.0845** $\pm$ 0.0040 | 0.1341 $\pm$ 0.0228 |
| JAHS/colorectal_histology | **0.0511** $\pm$ 0.0050 | 0.0633 $\pm$ 0.0088 |
| JAHS/fashion_mnist | 0.0526 $\pm$ 0.0021 | **0.0523** $\pm$ 0.0017 |
| HPO-B/6767:9914 | **0.0364** $\pm$ 0.0101 | 0.0464 $\pm$ 0.0044 |
| HPO-B/6794:3 | 0.0490 $\pm$ 0.0031 | **0.0436** $\pm$ 0.0000 |
| HPO-B/6794:9914 | **0.0148** $\pm$ 0.0001 | 0.0149 $\pm$ 0.0000 |
| PV/Crossed Barrel | **1.4159** $\pm$ 0.1622 | 1.4173 $\pm$ 0.1283 |
| Floris | **2.0565** $\pm$ 0.5485 | 2.5907 $\pm$ 0.3100 |

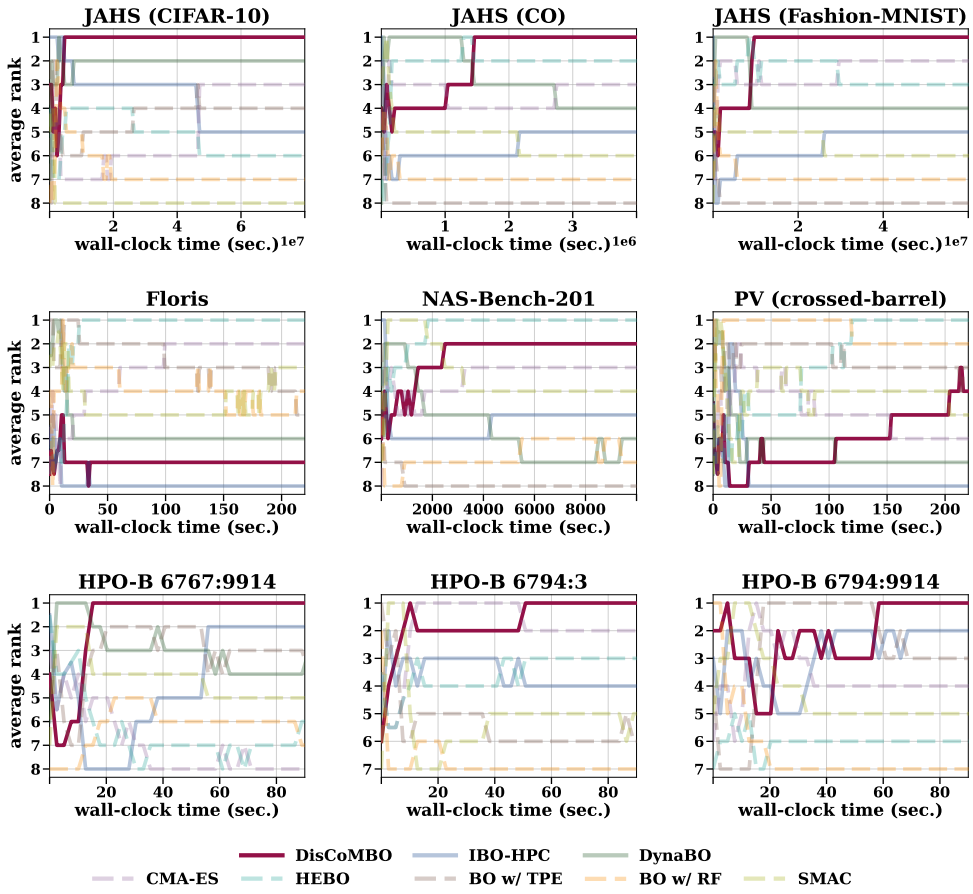



Figure 15: **Average algorithmic rank over time across 50 independent seeds.** DisCoMBO (—) achieves the top final rank on 6/9 tasks while remaining highly competitive on the remaining landscapes, with the exception of Floris (see App. C.4). On the AutoML suite, DisCoMBO quickly identifies high-quality solutions early in the optimization trajectory. On PV (crossed-barrel), it benefits from structured exploration to effectively exploit the search space in later iterations.

Table 4: **DisCoMBO scheduling performs similar to UCB-style acquisition.** When incorporating DisCo scores as uncertainty estimate in a UCB-style construction of an acquisition function, we observed marginal differences to our proposed scheduling mechanism.

| | JAHS/CO | NAS201 | HPO-B (6794:9914) | PV |
|---|---|---|---|---|
| norm. regret (UCB vs. Scheduling) | +0.27% | +1.05% | –0.21% | –2.01% |

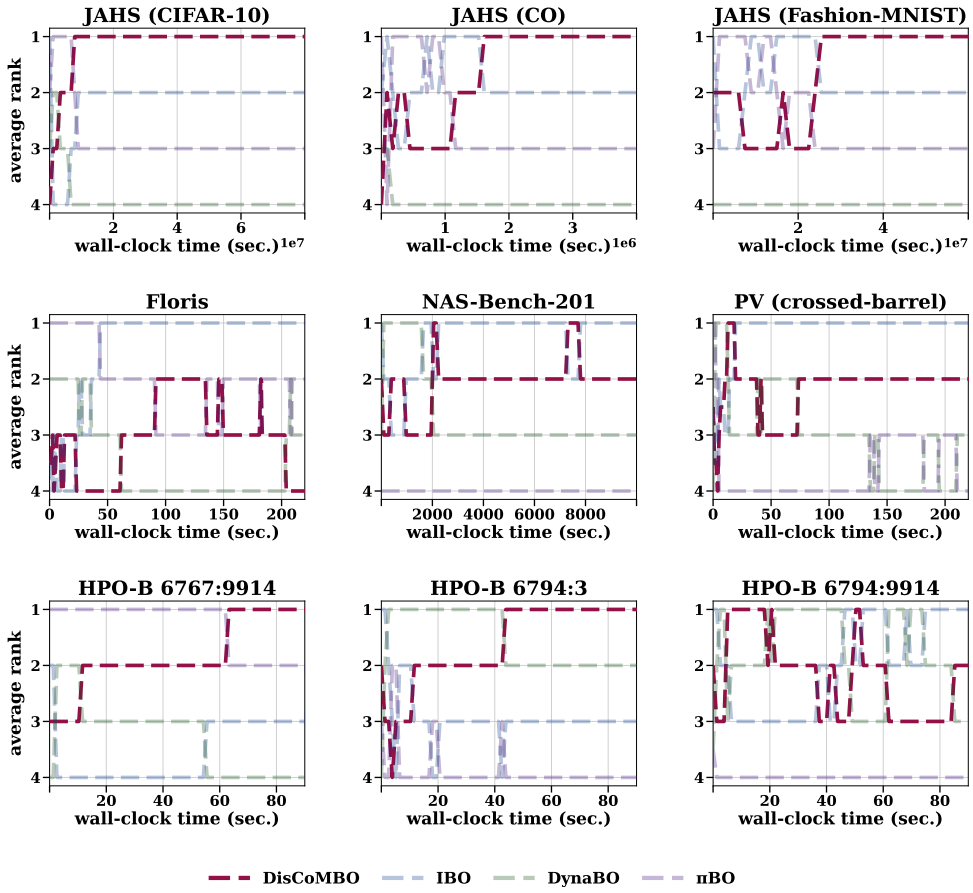



Figure 16: **Average algorithmic rank over time given a beneficial prior at iteration 5 (averaged across 50 seeds).** DisCoMBO (- - ) achieves the top final rank on **7/9** tasks. Across all benchmarks except Floris, the framework successfully capitalizes on the injected prior, demonstrating that DisCoMBO retains or enhances the targeted knowledge integration capabilities of purely generative approaches like IBO-HPC.

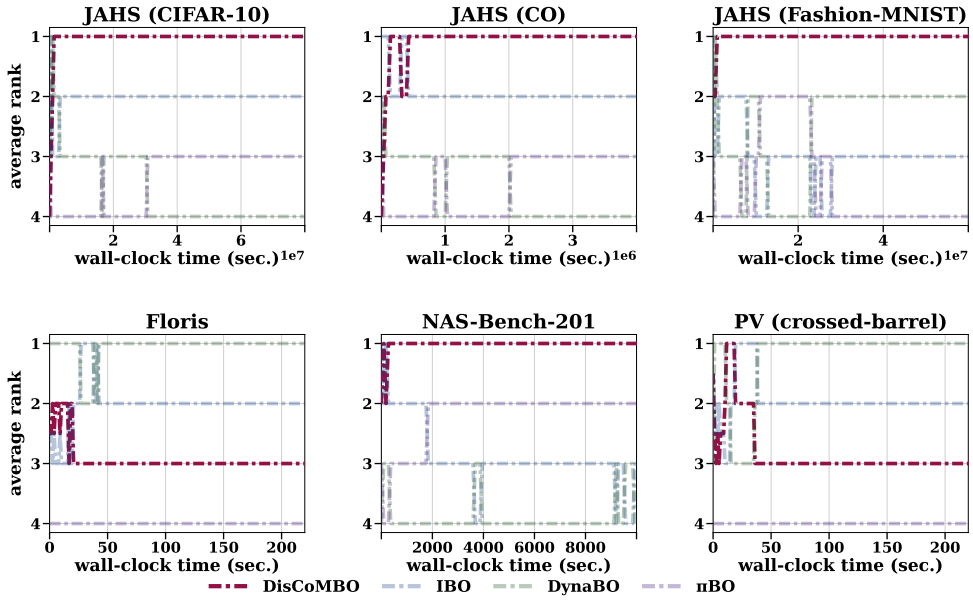



Figure 17: **Average algorithmic rank over time given a misleading prior at iteration 5 (averaged across 50 seeds).** DisCoMBO (-·-·-) achieves the top final rank on **4/6** tasks. Across all benchmarks except Floris and PV (crossed-barrel), the framework successfully recovers from a misleading prior given after 5 iterations, demonstrating that DisCoMBO retains or enhances the targeted knowledge integration capabilities of purely generative approaches like IBO-HPC.

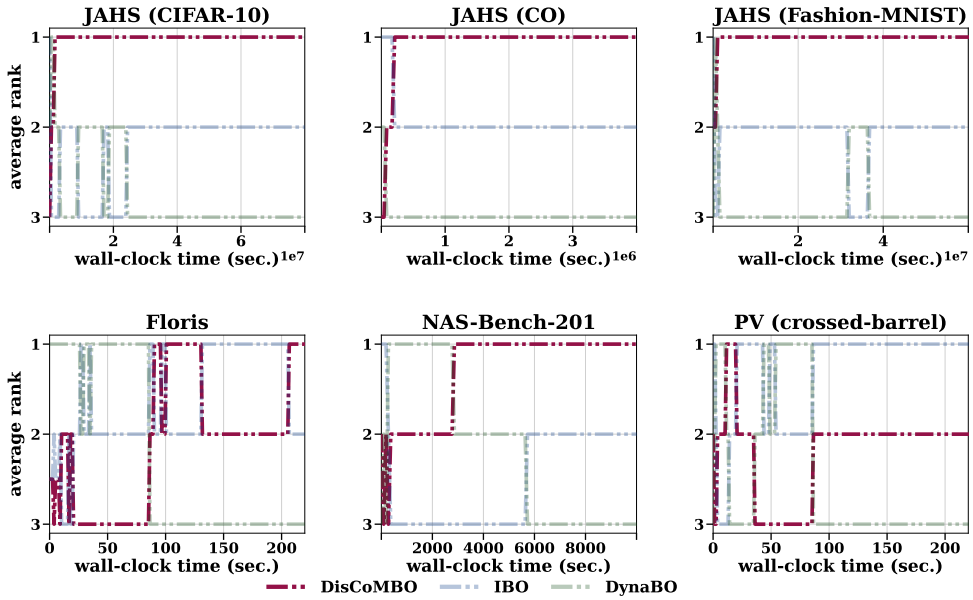



Figure 18: **Average algorithmic rank over time given a sequence of priors (averaged across 50 seeds).** DisCoMBO (-·-·-) achieves the top final rank on **5/6** tasks. Across all benchmarks except PV (crossed-barrel), the framework successfully recovers from a misleading prior given after 5 iterations, demonstrating that DisCoMBO retains or enhances the targeted knowledge integration capabilities of purely generative approaches like IBO-HPC.

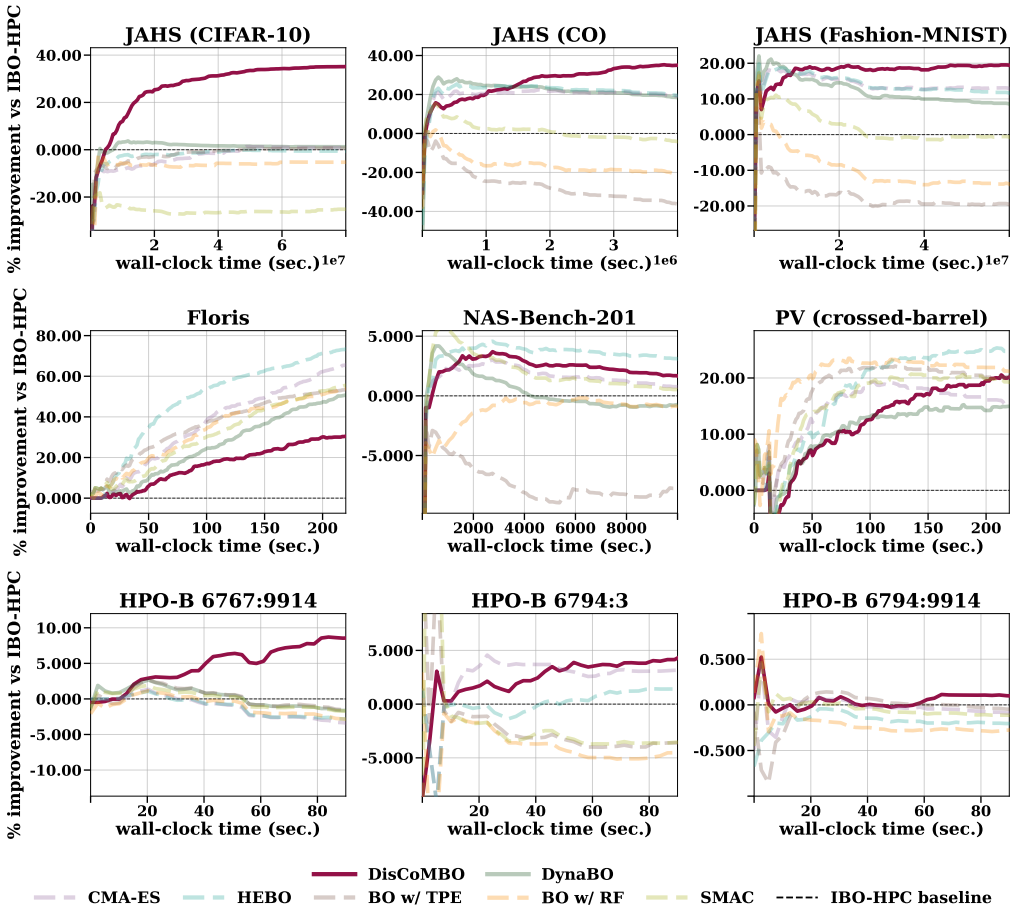



Figure 19: **Relative improvement of DisCoMBO against IBO-HPC over time without priors.** We observe that DisCoMBO (—) consistently achieves a significant improvement over its fully generative counterpart IBO-HPC on **8/9** tasks, demonstrating the effectiveness of our exploration-exploitation strategy driven by DisCo.

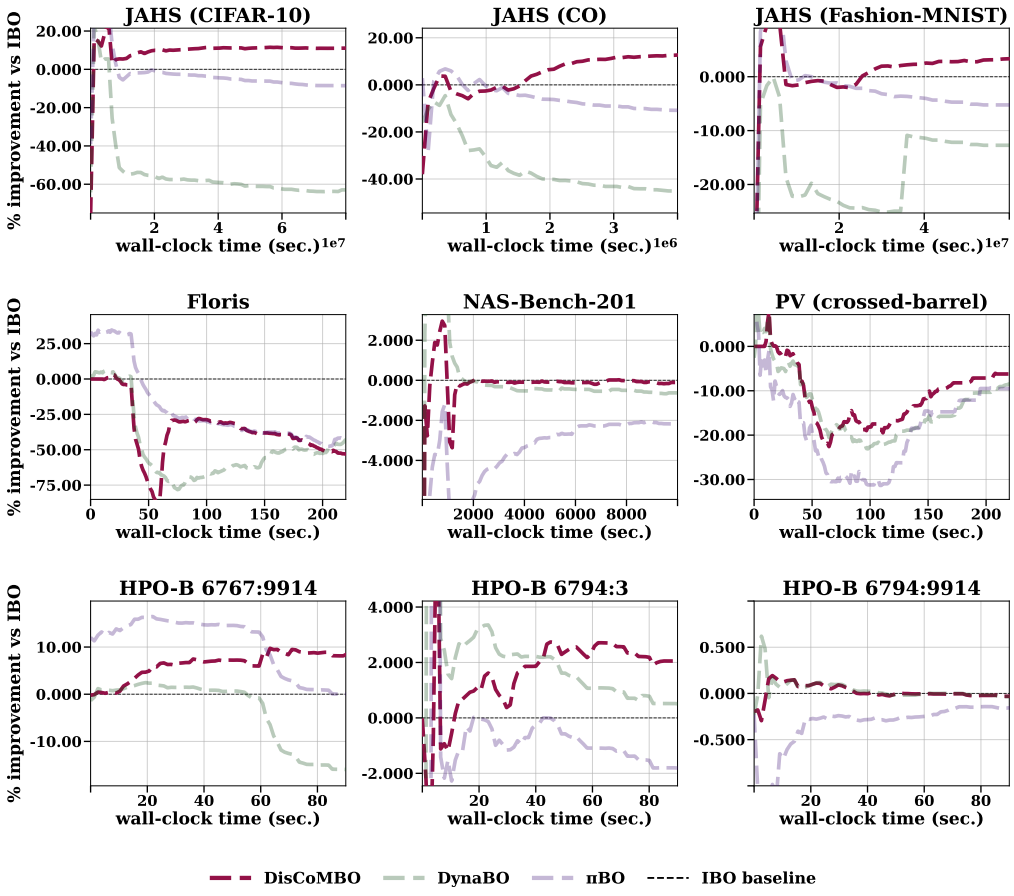



Figure 20: **Relative improvement of DisCoMBO against IBO-HPC over time with beneficial prior at 5 iterations.** We observe that DisCoMBO (- - ) consistently achieves a significant improvement over its fully generative counterpart IBO-HPC or matches its performance on **7/9** tasks when beneficial priors are provided early during optimization, demonstrating that DisCoMBO effectively incorporates priors.

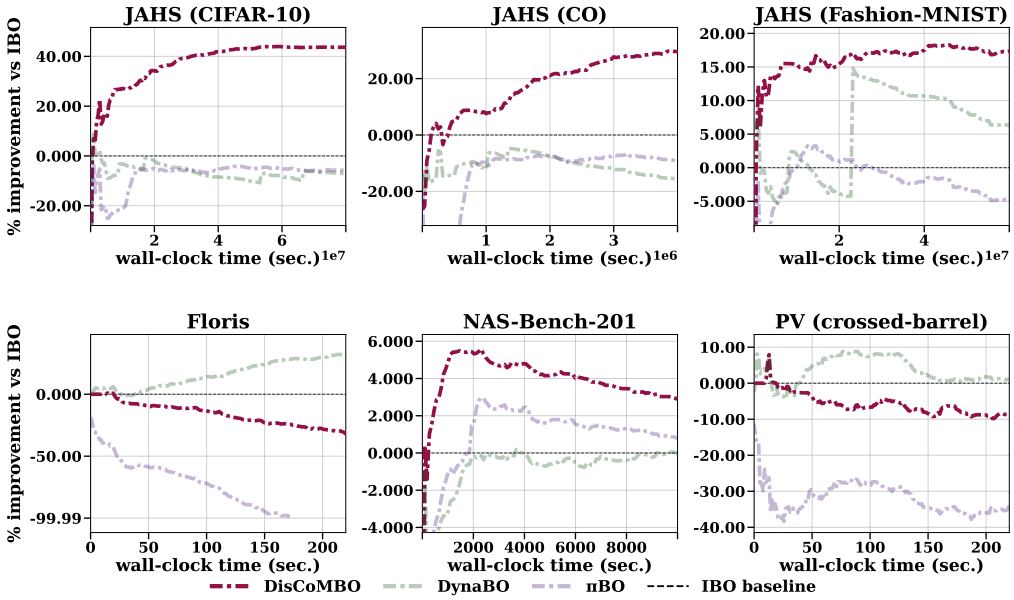



Figure 21: **Relative improvement of DisCoMBO against IBO-HPC over time with misleading prior at 5 iterations.** We observe that DisCoMBO (-·-) consistently achieves a significant improvement over its fully generative counterpart IBO-HPC or matches its performance on **4/6** tasks when misleading priors are provided early during optimization, demonstrating that DisCoMBO effectively recovers from misleading priors.

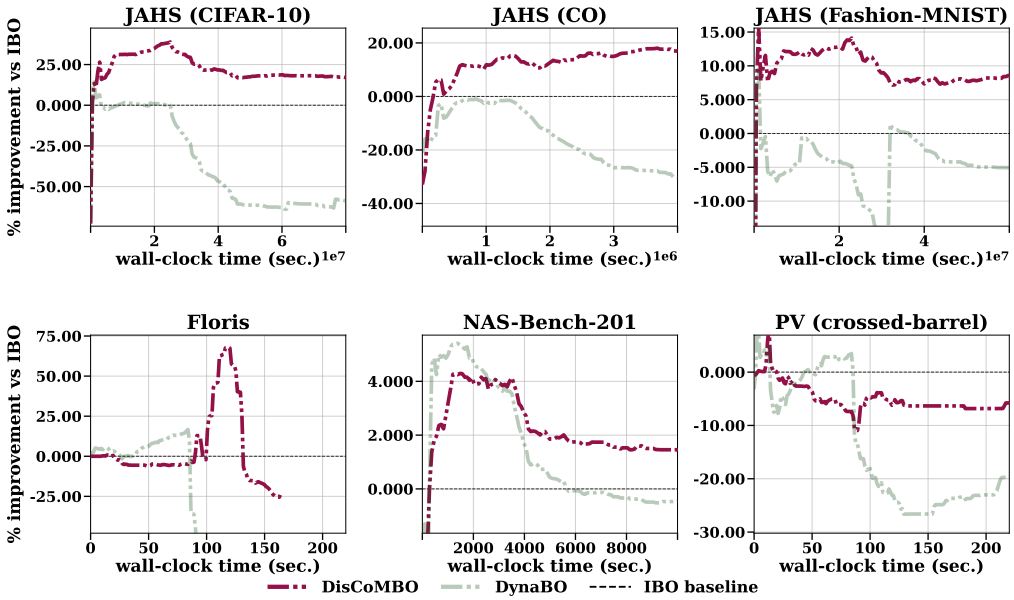



Figure 22: **Relative improvement of DisCoMBO against IBO-HPC over time with a sequence of priors given.** We observe that DisCoMBO (-·-) consistently achieves a significant improvement over its fully generative counterpart IBO-HPC or matches its performance on **4/6** tasks when a sequence of priors with contradicting information is provided during optimization, demonstrating that DisCoMBO effectively handles multiple priors.

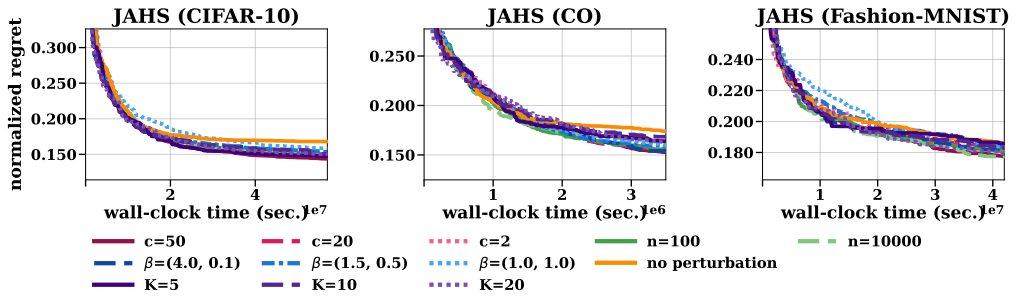



Figure 23: **DisCoMBO is robust across different hyperparameter settings.** It can be seen that DisCoMBO achieves similar performances regardless of how hyperparameters $c$, $n$, $\beta$, and $k$ are set. This demonstrates that no careful tuning of DisCoMBO's hyperparameters is necessary to achieve strong performance.

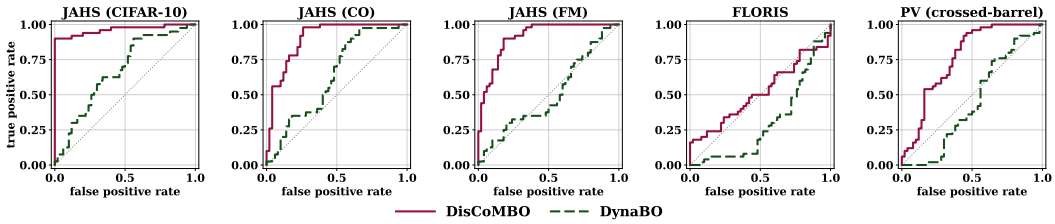



Figure 24: **DisCoMBO can discriminate between beneficial and adversarial priors.** On **4/5** datasets, our method (—) is able to identify most of the 10 good and 10 bad interventions provided to the model correctly as such. On **5/5**, DisCoMBO directly outperforms DynaBO (- - ), indicating it is well-suited for human-in-the-loop optimization scenarios even without extensive domain knowledge.

### C.5 Hyperparameters of DisCoMBO

DisCoMBO introduces a set of hyperparameters governing the annealing schedule, sample selection, and surrogate model updates. For the annealing schedule, which controls the length of exploration and exploitation phases as well as the skew of bucket definitions, we used the periodic function

$$s(t) = \frac{\beta_0 + \beta_1}{2} + \frac{\beta_0 - \beta_1}{2} \cos\left(\frac{\pi t}{c}\right)$$

with $\beta_0 = 3$ and $\beta_1 = 0.3$, and a cycle length of $c = 20$. At each iteration, we draw $N = 1000$ candidate samples from the conditional distribution $p(\mathbf{x}|y = y^*)$ and select $K$ samples, one for each out of the $B = K$ bins. Following Seng et al. [2025], drawn samples are perturbed by $5\%$. The size of the initial dataset $J$ is set equal to $K$. Specifically, we set $K = J = 20$ for JAHS and Nas201; $K = J = 10$ for HPO-B; and $K = J = 5$ for Floris and PV. For the surrogate model updates, we fix the interval during which the surrogate remains unchanged to $L = 5$ iterations, and set the decay factor to $\gamma = 0.9$. The underlying Probabilistic Circuit (PC) surrogates and their corresponding structure learning algorithm also require specific hyperparameter choices. Structure learning is driven by K-means clustering and Randomized Dependence Coefficient (RDC) independence tests, where we set the RDC threshold for independence detection to $0.3$. To handle varying data regimes during the optimization trajectory, the minimum number of instances per leaf node is adapted dynamically based on the total number of configurations evaluated up to the current iteration. To ensure a fair comparison across all evaluated methods, we fix the maximum optimization budget to $50$, $100$, or $250$ iterations, depending on the benchmark task. For all baseline methods, we set identical hyperparameters if applicable, or set the total number of evaluations equal to the budget of DisCoMBO. Finally, we repeated each experiment across 50 different seeds to ensure statistical significance. Sensitivity analyses and ablations evaluating alternative hyperparameter configurations are provided in App. C.4.

### C.6 Hardware and Computational Cost

Although our experimental setup does not require much compute, we ran all our experiments on DGX-A100 machines to speed up evaluation. We used 10 CPUs for each run, thus parallelizing some sub-routines (e.g., learning of PCs). We did not use any GPUs as we queried the benchmarks employed to provide the performance of configurations. The JAHS benchmark requires a relatively large RAM ($> 16GB$) to run smoothly as it loads large ensemble models.

### C.7 Used Assets

In our experimental evaluation, we used the official implementations and assets for JAHS [Bansal et al., 2022], NAS-Bench-201 [Dong and Yang, 2020], HPO-B [Pineda-Arango et al., 2021], and PV (crossed-barrel) [Gongora et al., 2020]. All of these are open source and open for free use under the license CC BY 4.0. The Floris (see `https://github.com/NatLabRockies/floris`) simulator is licensed under the BSD 3-Clause License and is free to use.

### C.8 Released Assets

Besides code, we also release two additional surrogate benchmarks for PV (crossed-barrel) and Floris.

**PV** We used the data provided by Gongora et al. [2020] and trained a surrogate model on that data to simulate a non-discretized version of the benchmark. Following best practices of surrogate benchmarking, we trained a Gradient Boosting Tree model with 300 trees. To ensure model fitting quality, we applied 5-fold cross-validation. We provide the code for model training in our repository and provide the learned models in `https://doi.org/10.6084/m9.figshare.32183937`.

**Floris** Floris is a simulation-based benchmark. To decouple our evaluation setup from the simulator, we evaluated $10,000$ random configurations using the simulator provided at `https://github.com/NatLabRockies/floris` with default settings. Based on the obtained data, we trained a Gradient Boosting Tree model with 300 trees and applied 5-fold cross-validation for quality assurance. We provide the code for model training in our repository and provide the trained models at `https://doi.org/10.6084/m9.figshare.32183937`.

Table 5: Fisher's method ranks cases similarly to raw multiplication (a–c, h, i), but accounts for the number of observations. In g), nine ones and one $10^{-2}$ give the same product as d), yet Fisher's method reports $p \approx 0.98$ instead of $\approx 0.06$ — a single weak signal among nine nulls is correctly judged uninformative. The arithmetic mean stays above $0.1$ even when nine of ten observations are tiny (h, i), washing out localized signal.

| | input | multiplication | arithmetic mean | Fisher $p$ |
|---|---|---|---|---|
| a) | (1e-10, 1e-10) | $1.00e-20$ | $1.00e-10$ | $4.71e-19$ |
| b) | (1e-10, 0.01) | $1.00e-12$ | $5.00e-3$ | $2.86e-11$ |
| c) | (1e-10, 1) | $1.00e-10$ | $5.00e-1$ | $2.40e-9$ |
| d) | (0.01, 1) | $1.00e-2$ | $5.05e-1$ | $5.61e-2$ |
| e) | (1, 1) | $1.00e0$ | $1.00e0$ | $1.00e0$ |
| f) | 1×1e-10, 9×1 | $1.00e-10$ | $9.00e-1$ | $7.93e-4$ |
| g) | 1×0.01, 9×1 | $1.00e-2$ | $9.01e-1$ | $9.80e-1$ |
| h) | 9×1e-10, 1×1 | $1.00e-90$ | $1.00e-1$ | $2.03e-75$ |
| i) | 9×0.01, 1×1 | $1.00e-18$ | $1.09e-1$ | $1.26e-9$ |

# D On Fisher's method in DisCo

To quantify the degree to which an observation $\mathbf{x}$ conforms to our learned distribution $p$, our score should fulfill the following desiderata: First, it should be bounded in $[0, 1]$, where $1$ indicates conformance, related to $\mathbf{x}$ being close to a mode of $p$, and $0$ indicates surprise, related to $\mathbf{x}$ being truly outside of the critical mass of $p$. Second, surprise in one dimension should translate into an aggregated value representing this surprise accurately. If $\mathbf{x}$ is highly unusual in any dimension, we want to treat it as being nonconformant w.r.t. $p$. Third, it should be symmetric and monotone to enable coherent ranking of observations. Lastly, it should be computationally efficient, i.e., provide linear runtime inference when computing the score in a probabilistic circuit. We found that Fisher's method [Fisher, 1932], originally designed to aggregate evidence in hypothesis testing, provides our desired properties. Since its result stems from a cumulative distribution function, it is guaranteed to be bound in $[0, 1]$. Moreover, its results can be recursively combined across arbitrary depths while retaining scale-invariance. Importantly, surprise found in one dimension of $\mathbf{x}$ is correctly reflected in the aggregated value opposed to e.g. mean-based aggregates, i.e., among a growing number of dimensions that are conformant, the non-conformant magnitude determines the overall reported value. Numeric examples are provided in Tab. 5. Since Fisher's method is also symmetric and monotone, it allows for the order-isomorphism property to symmetric kernel-based approaches of our method. Lastly, the evaluation of the $\chi^2$ cumulative distribution function is tractable, preserving the tractability of computing DisCo in PCs. We emphasize that the resulting scores can not be treated as the result of a hypothesis test and have no formal implications in the sense of testing methodology. However, our empirical evaluations revealed the consistency of the score to quantify the conformance of observations to our learned distribution on diverse benchmarks, which is also shown in Fig. 6.

# E Broader Impact

As a general-purpose black-box optimization framework, DisCoMBO carries both positive and negative downstream societal and environmental implications depending on its domain of application.

**Positive Impacts** The primary utility of DisCoMBO lies in its data efficiency when optimizing complex, mixed-variable systems under expert prior knowledge. By accelerating workflows in scientific discovery and renewable energy—such as optimizing wind farm layouts to maximize power generation or streamlining materials discovery—the framework can directly contribute to resource and cost reductions. Furthermore, by matching or outperforming traditional SMBO baselines with fewer function evaluations, DisCoMBO can reduce the overall computational footprint and associated carbon emissions of intensive hyperparameter tuning and neural architecture search.

**Negative Impacts & Dual-Use Risks** Conversely, as an application-agnostic tool, DisCoMBO introduces dual-use risks. The framework's optimization efficiency could be exploited to enhance the performance of objectionable technologies, such as mass surveillance software or automated weapon systems. Additionally, because DisCoMBO natively integrates external knowledge, malicious actors could deliberately inject biased or adversarial priors to manipulate optimization trajectories toward harmful or discriminatory outcomes in sensitive socio-technical deployments.

**Mitigation** We mitigate these risks by releasing our framework under open-source licenses that encourage transparent peer review. We urge practitioners deploying DisCoMBO in high-stakes domains to strictly audit the alignment and validity of external priors before integration.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide proofs of our theoretical claims (Appendix B) and provide an extensive empirical evaluation (Section 3 and Appendix C.4), confirming our claims.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a dedicated paragraph discussing limitations in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: For all propositions, proofs are provided in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide pseudo-code (Section 2.3), key hyperparameters that were set (Appendix C.5), ablation analysis on hyperparameters (Appendix C.4), and the hardware specification we used to run the experiments on (Appendix C.6). Further, we provide all benchmarks via a figshare link (`https://doi.org/10.6084/m9.figshare.32183937`) (Appendix C).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide an git repository at `https://github.com/bewit/DisCoMBO` including instructions on how to build the Docker container and how to run experiments. All data needed for our experiments (e.g., benchmarks used) are shared via a figshare URL (`https://doi.org/10.6084/m9.figshare.32183937`, see also Appendix C).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All hyperparameters we have set are described in Appendix C. Also, we provide ablation studies showing the effect of varying these hyperparameters (Appendix C.4).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide critical difference plots to compare our method with our baselines aggregated over all benchmarks. This naturally comes with statistical significance information (Wilcoxon signed rank test used with $p = 0.05$). Also, all our regret plots are plotted with standard error (see Section 3).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide a description of the hardware resources required to run our experiments in Appendix C.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have carefully read the code of ethics, and we do not think that our contribution exhibits risks beyond the risks of general AI and ML applications.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: We provide a discussion on the broader impact of our research in Appendix E.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not publish artifacts beyond surrogate benchmarks of black-box optimization tasks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the papers associated with each asset properly. We provide license information and URLs to the assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All new assets released are described in Appendix C.8.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our research does not include research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: See above (14.).

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were only used for writing purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.