# Efficient Adaptive Data Acquisition via Pretrained Belief Representations

**Daolang Huang**[1,2] **Zhuoyue Huang**[5] **Conor Hassan**[1,2]

**Luigi Acerbi**[3] **Samuel Kaski**[1,2,4] **Tom Rainforth**[5]

[1] ELLIS Institute Finland [2] Department of Computer Science, Aalto University, Finland
[3] Department of Computer Science, University of Helsinki, Finland
[4] Department of Computer Science, University of Manchester, UK
[5] Department of Statistics, University of Oxford, UK

## Abstract

Learning effective policies for adaptive data acquisition remains challenging: methods that select queries using explicit posterior approximations can be limited by the difficulty of inference, while policies learned directly from observation histories must jointly learn useful representations and acquisition strategies, making learning harder. To address this, we introduce policy learning with belief representations (POLAR), based on the insight that optimal data acquisition depends on the observation history only through a sufficient belief state. Specifically, POLAR decouples representation learning from policy learning by leveraging pretrained predictive foundation models as belief-state encoders and training a policy head on top of their representations. This yields a simple, unified amortised policy learning framework for Bayesian experimental design, Bayesian optimisation, and active learning, differing only in the task-specific utility used to train the policy. Empirically, we find that POLAR outperforms state-of-the-art amortised methods across diverse tasks while requiring far fewer training samples, demonstrating a significant step in the scalability and efficiency of amortised data acquisition.

## 1 Introduction

Many sequential learning problems can be posed as *adaptive data acquisition*: at each round, a learner chooses which query to make next to improve a downstream decision based on the data observed so far, receives the resulting observation, and repeats until a budget is exhausted. This abstraction covers Bayesian experimental design (BED) (Lindley, 1956, Chaloner and Verdinelli, 1995, Rainforth et al., 2024), Bayesian optimisation (BO) (Garnett, 2023), and active learning (AL) (Settles, 2012). These settings differ in their downstream objectives, but share the same central challenge: *mapping a growing observation history to acquisition decisions under a finite budget*.

Learning effective acquisition policies in this setting is difficult. Existing approaches fall into two families: those that work through an explicit approximation of the posterior, and those that map observation histories directly to decisions. Both are problematic in different ways. Posterior-based methods (Chang et al., 2025, Müller et al., 2023, Li et al., 2025b) allow the model to be exploited in the policy architecture itself and thus efficiently identify what aspects of the data are most important or our task. However, they must either perform expensive inference during the experiment or learn complex amortised inference networks, which is sometimes more challenging than the adaptive design problem itself (Foster et al., 2021), while errors in the inference can compound through the sequential acquisition process. Mapping directly from observation histories directly to the next decision (Foster et al., 2021, Ivanova et al., 2021, Huang et al., 2024, Maraval et al., 2023) avoids

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

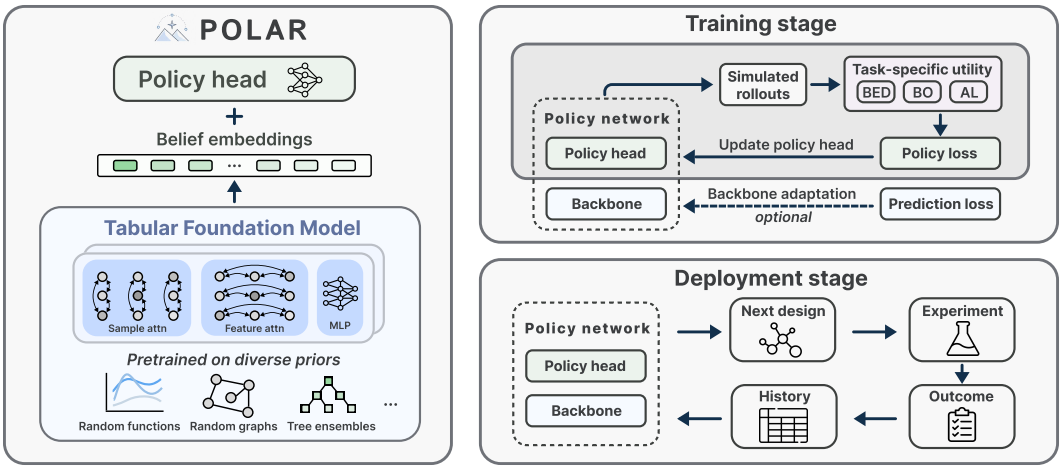



Figure 1: **Overview of POLAR.** *Left*: POLAR uses a pretrained tabular foundation model as a belief encoder and trains a policy head on top of it. *Top right*: The policy loss updates only the policy head using task-specific acquisition utilities. An optional auxiliary prediction loss adapts the backbone through supervised prediction on additional target points. *Bottom right*: At deployment, the policy maps the current history and the step to the next design in a single forward pass.

explicit posterior inference, but pays for this by not exploiting model structure and invariances, which makes the policy more difficult to learn, they must implicitly learn how the posterior changes with the data. In other words, they must simultaneously learn both how to *represent* the current dataset and how to *act* on that representation. The former can be as challenging as the latter, potentially becoming a bottleneck that increases the number of samples needed to train the policy.

Our key insight that allows us to address these limitations is that, while optimal sequential decisions depend only on the observation history through the corresponding posterior, a learned policy need not access this posterior explicitly: it may instead act on any *representation* that preserves the information needed to recover the posterior. This suggests a beneficial middle ground between raw observation histories and explicit posterior estimates that separates this representation learning from how to map representations to decisions. To exploit this insight, we therefore introduce a pre-trained intermediate *belief representation* that captures underlying features in how the posterior varies with the data, without over-committing to a single posterior approximation. This then simplifies policy learning while retaining information that an explicit posterior approximation may discard.

Pretrained predictive foundation models provide a natural starting point for this belief representation. Namely, rather than learning a representation from scratch, we can build on models whose pretraining has already shaped them to organise observed data into useful representations for a broad family of tasks. We instantiate this idea with Tabular Foundation Models (TFMs; Müller et al., 2025), such as TabPFN (Hollmann et al., 2025, Grinsztajn et al., 2025) and TabICL (Qu et al., 2025, 2026), which produce candidate-conditioned representations from a context set in a single forward pass.

Specifically, we introduce **PO**licy **LeA**rning with Belief **R**epresentations (POLAR), an amortised data acquisition framework that uses the hidden representations from a TFM as our belief representation and trains a policy head to select the next query from the encoding this provides. In doing so, POLAR shifts amortised acquisition from jointly learning representations and decisions to learning how to act on pretrained belief representations. An overview of POLAR is shown in Figure 1.

In summary, our main contributions are:

1. We give a unified decision-theoretic view of adaptive data acquisition as learning acquisition policies on top of belief representations, rather than learning sequential decision-making directly from raw histories or acting through explicit posterior approximations.
2. We introduce a simple decoupled policy architecture that uses pretrained predictive foundation models as belief-state encoders and trains a policy head on top.
3. Across diverse adaptive data acquisition tasks, the learned policy substantially outperforms greedy acquisition on the same backbone and exceeds state-of-the-art amortised methods while requiring up to $100\times$ fewer task-specific training samples.

## 2 Preliminaries

**Adaptive data acquisition.** We consider a finite-horizon adaptive data acquisition problem in which a learner repeatedly selects designs based on the data observed so far. A latent world state $\theta \sim p(\theta)$ determines observations through a model $p(y \mid x, \theta)$, where $x \in \mathcal{X}$ is a design and $y \in \mathcal{Y}$ is the corresponding outcome. At round $t$, the learner has observed $\mathcal{D}_{t-1} = \{(x_i, y_i)\}_{i=1}^{t-1}$, chooses a feasible design $x_t \in \mathcal{X}_t(\mathcal{D}_{t-1}) \subseteq \mathcal{X}$, observes $y_t \sim p_y(\cdot \mid x_t, \theta)$, and appends $(x_t, y_t)$ to the history. We aim to optimise an acquisition policy $\pi = (\pi_1, \ldots, \pi_T)$, consisting of a sequence of decision rules $\pi_t : \mathcal{H}_{t-1} \to \mathcal{X}$ satisfying $\pi_t(\mathcal{D}_{t-1}) \in \mathcal{X}_t(\mathcal{D}_{t-1})$. Different adaptive data acquisition problems share this sequential structure, but differ in the downstream objective and which aspect of the world state matters for this. For example, in many BED problems, we seek information about unknown system parameters $\theta$ (Chaloner and Verdinelli, 1995, Ryan et al., 2016, Rainforth et al., 2024); in active learning, we want to learn to make effective future predictions (Bickford Smith et al., 2023); and in BO, we seek evaluations that rapidly identify a global optimum (Garnett, 2023).

Existing approaches often choose each design by acting through an explicit approximation of the posterior: given $\mathcal{D}_{t-1}$, they approximate the posterior, either by directly performing inference on the underlying model or using an amortised neural predictor (Chang et al., 2025, Müller et al., 2023, Li et al., 2025b), and then select the next design by optimising an acquisition function. Examples include the traditional constructions of BO, AL, and BED, which iterate between updating the model and explicitly maximising an expected utility, such as the Expected Improvement (EI) in BO (Garnett, 2023) or the Expected Information Gain (EIG) in BED (Lindley, 1956, 1972). These methods rely on the quality of the posterior approximation exposed to the acquisition rule. Outside rare cases where exact inference is tractable, posterior approximation is hard, and its errors can propagate into suboptimal acquisitions (Lacoste-Julien et al., 2011, Maus et al., 2024), especially across multiple rounds.

An alternative is to learn a policy that maps observation histories directly to designs. In this case, a parameterised policy is trained from simulated trajectories to maximise a task-specific sequential utility, so that test-time decisions require only a forward pass rather than the optimisation of a hand-designed acquisition function. For example, amortised BED methods (Foster et al., 2021, Ivanova et al., 2021, Blau et al., 2022, 2023, Huang et al., 2024, 2026a, Bracher et al., 2025, Phillips et al., 2026) map observation histories directly to experimental designs, and related end-to-end policies have appeared for BO (Maraval et al., 2023, Zhang et al., 2026a) and AL (Huang et al., 2026a, Li et al., 2025a). This amortises test-time decision-making, but requires the same model to learn both a representation of the current dataset and an acquisition strategy from sequential supervision alone.

**Pretrained predictive models.** Recent work on in-context learning has produced models that can approximate predictive distributions in a single forward pass, without iterative fitting or gradient updates. Transformer Neural Processes (Nguyen and Grover, 2022) and Prior-Fitted Networks (Müller et al., 2021) learn to map a context set of observed input-output pairs to predictive distributions for new inputs, after offline training on large families of synthetic tasks. More recent Tabular Foundation Models (TFMs), such as TabPFN (Hollmann et al., 2025, Grinsztajn et al., 2025) and TabICL (Qu et al., 2025, 2026), scale this idea to practical tabular prediction, achieving strong performance across a wide range of supervised learning problems. Given a dataset $\mathcal{D}$ and a candidate input $x$, these models produce a predictive distribution $q_\psi(\cdot \mid x, \mathcal{D})$ for the corresponding outcome, together with internal representations $r_\psi(x, \mathcal{D})$ that depend on both $x$ and $\mathcal{D}$. These representations are not designed for any particular sequential task, but can encode predictive uncertainty, local structure, and context-dependent function behaviour.

## 3 A Unified Decision-Theoretic View of Adaptive Data Acquisition

To understand policy learning for adaptive data acquisition, we must first formalise what an ideal representation should capture. We do so by casting adaptive data acquisition under a unified Bayesian decision-theoretic framework.

**Bayes-optimal policy.** Consider a finite-horizon sequential data acquisition problem. Once all data has been acquired, a downstream action, $a \in \mathcal{A}$, is chosen, which incurs loss $\ell(a, z)$, where $z = \tau(\theta)$ represents the aspects of the overall world state that directly impact our loss. Depending on context, $a$ might be an estimate, a predictive distribution, or a physical decision. Following Bayesian decision theory (Savage, 1951, Lindley, 1972, Berger, 1985, Huang et al., 2026b) and defining $p(z \mid \mathcal{D}_T)$ as

the pushforward of our posterior onto $z$, the Bayes-optimal acquisition policy is now:

$$\pi^\star = \arg\min_{\pi\in\Pi} \mathbb{E}_{p(\mathcal{D}_T;\pi)}\left[\min_{a\in\mathcal{A}} \mathbb{E}_{p(z|\mathcal{D}_T)}[\ell(a,z)]\right]. \tag{1}$$

Here the expected loss of the Bayes action, $\min_{a\in\mathcal{A}} \mathbb{E}_{p(z|\mathcal{D}_T)}[\ell(a,z)]$, is a function only of our posterior beliefs for a given loss function. We can thus rewrite Equation (1) as an expected functional of the posterior (Rossa et al., 2026): defining $s(q,z) := \ell(a^\star(q),z)$ where $a^\star(q) := \arg\min_{a\in\mathcal{A}} \mathbb{E}_{z\sim q}[\ell(a,z)]$, then,

$$\pi^\star = \arg\min_{\pi\in\Pi} \mathbb{E}_{p(\mathcal{D}_T;\pi)p(z|\mathcal{D}_T)}[s(p_z(\cdot \mid \mathcal{D}_T),z)] = \arg\min_{\pi\in\Pi} \mathbb{E}_{p(\theta)p(\mathcal{D}_T|\theta;\pi)}[s(p_z(\cdot \mid \mathcal{D}_T),z)]\,. \tag{2}$$

It turns out that this $s$ is always a proper scoring rule for distributions on $z$ (Savage, 1971, Gneiting and Raftery, 2007). We can thus view Equation (2) as an *expected posterior uncertainty*, $\mathbb{E}_{p(\mathcal{D}_T;\pi)}[h_s[p_z(\cdot \mid \mathcal{D}_T)]]$, where our uncertainty measure is the *generalised entropy* $h_s[q] = \mathbb{E}_{z\sim q}[s(q,z)]$ (Dawid, 1998, Bickford Smith et al., 2025).

Under this unified foundational framework of sequential Bayesian decision-making, we can thus equivalently think about defining adaptive data gathering problems through defining either downstream losses, scoring rules, or uncertainty measures, all of which also (implicitly) define the target variables of interest for our data gathering, $z$. Most existing BED and active learning methods are explicitly based on defining uncertainty measures, which is typically taken to be entropy to yield the expected information gain (Lindley, 1956, Rainforth et al., 2024), though they can vary in what variables are targeted for learning, e.g., in AL, we can either take $z$ to be model parameters (Houlsby et al., 2011) or downstream predictions (Bickford Smith et al., 2023). In BO, it is more common to work directly with a downstream loss, with $z$ taken to be the true function values at the queried inputs. For example, we can recover a strategy equivalent to targeting expected improvement by taking $\ell_{\text{EI}}(a,f) = -f(a)$ and then $\mathcal{A}$ as the set of inputs we have queried (such that the minimisation over $a$ trivially yields the point with the highest posterior mean).

**From posterior beliefs to belief representations.** We can further rearrange Equation (2) to make clear the dependency of the optimal policy on historical data as follows (where $\mathcal{D}_{t:T} = \{(x_i,y_i)\}_{i=t}^T$)

$$\pi^\star = \arg\min_{\pi\in\Pi} \mathbb{E}_{p(\mathcal{D}_{t-1};\pi)}\left[\mathbb{E}_{p(\theta|\mathcal{D}_{t-1})p(\mathcal{D}_{t:T}|\theta;\pi)}[s(p_z(\cdot|\mathcal{D}_T),z)]\right],$$

where because $p(z|\mathcal{D}_T)$ is a pushforward of $p(\theta|\mathcal{D}_T)$, it only depends on $\mathcal{D}_{t-1}$ through the intermediate world state posterior $p(\theta|\mathcal{D}_{t-1})$. Once $\mathcal{D}_{t-1}$ has been observed, the outer expectation disappears for the remaining decisions as $\mathcal{D}_{t-1}$ is fixed. This now yields a crucial insight: the Bayes-optimal policy depends on the history only through the current world state posterior $p(\theta|\mathcal{D}_{t-1})$ and the current step $t$ (which is required for knowing how many steps are left in our process, as this affects things like exploration-exploitation trade-offs). Equivalently, there exist maps $f_{\mathcal{D}}$, $f_{\text{post}}$ such that

$$\pi_t^\star(\mathcal{D}_{t-1}) \;=\; f_{\mathcal{D}}(\mathcal{D}_{t-1}) \;=\; f_{\text{post}}(p_\theta(\cdot \mid \mathcal{D}_{t-1}), t)\,. \tag{3}$$

Thus, any two datasets that induce the same world state posterior at the same step should lead to the same optimal next action. This invariance suggests that the right input to policy learning is not necessarily the raw dataset itself, but some representation that preserves the information in $\mathcal{D}_{t-1}$ needed to recover our world state belief. In particular, we want a representation that faithfully retains this information, while also providing the simplest possible mapping to the optimal policy decisions.

To this end, we form $e_t = (\text{enc}(D_{t-1}), t)$ using a belief encoder enc and the current step. We call $e_t$ a belief representation if there exists a decoder dec such that

$$\text{dec}(e_t) = (p_\theta(\cdot \mid \mathcal{D}_{t-1}), t)\,. \tag{4}$$

Whenever the decoding condition above holds, there also exists a policy head $g$ for which

$$\pi_t^\star(\mathcal{D}_{t-1}) = g(e_t), \tag{5}$$

because the posterior and current step are determined by $e_t$. Two extreme choices of $e_t$ recover the two dominant approaches in the literature:

- **Raw history:** $e_t = \mathcal{D}_{t-1}$. This is trivially a belief representation, since the posterior can be recovered from the history and the current step is implied by the dataset itself. This is the input used by direct policy-learning methods, but it leaves the policy head $g$ with the entire job of capturing the dependency of the posterior on the data, significantly complicating practical training.

- **Posterior:** $e_t = (p_\theta(\cdot \mid \mathcal{D}_{t-1}), t)$. This is the belief state which yields a trivial decoder (the identity function) and thus incorporates all desired posterior invariances (though there may be further invariances in the mapping from intermediary posteriors to expected future uncertainties), thus making the training of $g$ much easier. In practice, however, this belief state cannot usually be calculated exactly and is replaced by an approximation $\hat{p}_\theta(\cdot \mid \mathcal{D}_{t-1})$, no longer guaranteeing the decoding condition in Equation (4) and exposing $g$ to the approximation error of the inference step.

Rather than going with either of these extremes, our key idea is to use an encoder that avoids needing an overly complex $g$ while still retaining information useful for acquisition, without restricting the policy to an explicit posterior approximation. We achieve this by using encodings that provide effective *representations* of the posterior, without over-committing to using precise posterior approximation. Section 4 shows how predictive foundation models provide an appropriate and practical encoder as a by-product of their training.

## 4 Policy Learning with Belief Representations

Section 3 argued that acquisition policies should act on representations of the world state posterior and the current step: representations that retain the information in observation histories needed for optimal decisions, while structured enough to make the map to acquisitions easy to learn. Pretrained predictive foundation models are natural candidates for this role: to predict well across diverse tasks from in-context observations alone, such models are encouraged to summarise the observed data into representations that track how beliefs change as data accumulates, which is useful for acquisition. We now describe how POLAR instantiates this principle using Tabular Foundation Models (TFMs) as practical belief encoders.

**Foundation models as belief encoders.** A pretrained TFM $F_\psi$ is trained to map context sets to predictive distributions across a broad family of synthetic tasks (Hollmann et al., 2025, Qu et al., 2025). Given a dataset $\mathcal{D}_{t-1}$ and a candidate $x \in \mathcal{X}$, a forward pass returns (i) a predictive distribution $q_\psi(\cdot \mid x, \mathcal{D}_{t-1})$ over the outcome $y$, and (ii) a candidate-conditioned hidden representation $r_\psi(x, \mathcal{D}_{t-1}) \in \mathbb{R}^{d_r}$. The predictive distribution $q_\psi$ is decoded from $r_\psi$ through a fixed prediction head. It lives in outcome space, approximating a posterior predictive over $y$ at a candidate $x$, whereas the decision-theoretic object in Section 3 is the posterior over the world state $\theta$. Thus, although $q_\psi$ is useful for prediction and for some task-specific surrogates, it is not in general itself a belief representation for $\theta$.

We therefore use the hidden representation $r_\psi$, rather than $q_\psi$, as the interface for policy learning. The predictive head may discard useful features when mapping $r_\psi$ to $q_\psi$. Training a shared model to support predictions at arbitrary inputs may further encourage features that capture how the data constrain the underlying predictive function. These considerations do not, however, guarantee that $r_\psi$ is sufficient for the posterior over $\theta$. For a given history, even exact marginal predictions at every input do not determine the joint predictive distribution over future observations. The importance of this distinction depends on the acquisition objective: marginal predictions suffice for some myopic criteria, while joint dependencies can matter for acquisition decisions, especially when planning ahead. Motivated by Section 3, we therefore pair $r_\psi$ with the current step $t$ as an approximate belief representation and assess its usefulness empirically in Section 6.

While the framework of Section 3 applies to general design spaces, TFMs are far more efficient when the context $\mathcal{D}_{t-1}$ is processed once and reused across a finite candidate set. We therefore work in a pool-based setting: at each round, the policy chooses from a feasible pool $\mathcal{C}_t \subseteq \mathcal{C}$. A single batched pass over $\mathcal{C}_t$ returns the candidate-conditioned representations $\{r_\psi(x, \mathcal{D}_{t-1}) : x \in \mathcal{C}_t\}$. This setting matches many practical problems: hyperparameter optimisation over discrete configuration grids, molecular optimisation over finite molecule libraries, and active learning over finite unlabelled pools. For continuous design spaces, the same architecture can be combined with a candidate generator or evaluated over sampled candidate sets.

**Policy head.** We instantiate the policy head $g$ as a function $g_\eta : \mathbb{R}^{d_r+1} \to \mathbb{R}$ with learnable parameters $\eta$. For each candidate $x \in \mathcal{C}_t$, $g_\eta$ maps $r_\psi(x, \mathcal{D}_{t-1})$, concatenated with the number of remaining acquisitions $T - t + 1$ normalised by the training horizon, to a scalar logit, and normalising these logits across the candidate pool with a softmax gives the stochastic policy $\pi_\eta(\cdot \mid \mathcal{D}_{t-1}, \mathcal{C}_t, t)$. We parameterise $\pi_\eta$ as a distribution over candidates during training, which provides exploration and

admits standard stochastic-gradient policy training. At deployment, we select the candidate with the largest learned score.

**Task scoring rules.** The architecture is shared across all tasks; only the task-specific scoring rule changes. For each task, we define $\widehat{\Phi}(\mathcal{D}_T, z)$ as a tractable surrogate of $-s(p_z(\cdot|\mathcal{D}_T), z)$. In some tasks, this score is directly observable, while in others a surrogate estimator is needed. In BED, the natural scoring rule is the log loss on a target variable, whose expected reduction is the EIG. Since the EIG is intractable, we approximate it with the sPCE lower bound (Foster et al., 2021). Other EIG estimators or losses could be substituted without further change. In BO, the queried function values $y_t$ are directly observed, so the score reduces to a tractable function of the trajectory. We use the terminal best observed value, $\widehat{\Phi}_{\mathrm{BO}}(\mathcal{D}_T, z) = \max_{t \le T} y_t$, for which no posterior approximation is required. In AL, when the goal is to make future predictions (Bickford Smith et al., 2023), the terminal score is the predictive log-density on a target set, which is also intractable, so we use a variational surrogate built on $q_\psi$.

**Policy training.** We train the policy using reinforcement learning on simulated tasks to maximise the expected terminal score $\mathbb{E}_{p(\theta)p(\mathcal{D}_T|\theta;\pi_\eta)}[\widehat{\Phi}(\mathcal{D}_T, z)]$. Since the policy selects among discrete candidates, we use score-function gradients rather than reparameterisation; these also do not require differentiating through the simulator. To provide a dense training signal, we express the terminal objective through per-step utilities $u_t = \widehat{\Phi}(\mathcal{D}_t, z) - \widehat{\Phi}(\mathcal{D}_{t-1}, z)$, so that $\sum_{t=1}^{T} u_t = \widehat{\Phi}(\mathcal{D}_T, z) - \widehat{\Phi}(\mathcal{D}_0, z)$. For each sampled trajectory, we use the policy-gradient surrogate

$$\mathcal{L}_{\mathrm{pol}}(\eta) = -\sum_{t=1}^{T} G_t^{(\gamma)} \log \pi_\eta(x_t \mid \mathcal{D}_{t-1}, \mathcal{C}_t, t), \qquad G_t^{(\gamma)} = \sum_{s=t}^{T} \gamma^{s-t} u_s, \tag{6}$$

where $\gamma \in [0, 1]$. We average this loss over sampled trajectories, treating the sampled data and $G_t^{(\gamma)}$ as fixed when differentiating with respect to $\eta$.

With $\gamma = 1$, the update includes all subsequent utility increments, as required by the terminal-score objective. Smaller values of $\gamma$ provide a training approximation: in particular, $\gamma = 0$ uses only the immediate utility $u_t$, giving a myopic training signal. This can simplify learning and reduce gradient variance, but ignores an action's effects on later acquisitions. We use $\gamma = 0$ in our main experiments, where this approximation is empirically effective. Appendix D.6 additionally considers a dynamical task where planning ahead is important and $\gamma = 1$ substantially improves performance.

**Backbone adaptation.** The pretrained TFM backbone $F_\psi$ is trained on synthetic priors that may not perfectly match the downstream task distribution. We therefore optionally adapt $\psi$ using an auxiliary supervised prediction loss, $\mathcal{L}_{\mathrm{pred}}(\psi) = \sum_{t=1}^{T} \sum_{m=1}^{M} \ell_{\mathrm{pred}}\big(q_\psi(\cdot \mid x_m^\star, \mathcal{D}_{t-1}), y_m^\star\big)$, where $\{(x_m^\star, y_m^\star)\}_{m=1}^{M}$ are target points sampled from the same simulated task during training. The form of $\mathcal{L}_{\mathrm{pred}}$ depends on the prediction head: for density-based heads (Hollmann et al., 2023), it is negative log-likelihood, while for quantile-based heads (Qu et al., 2026), it is quantile regression loss. A key design choice is that policy gradients are not propagated into the pretrained backbone: $\mathcal{L}_{\mathrm{pol}}$ updates only the policy head, while $\mathcal{L}_{\mathrm{pred}}$ is the only loss that may update $\psi$. This adapts the backbone through supervised prediction while preventing high-variance policy gradients from destabilising it, keeping representation refinement and decision learning decoupled.

**Limitations and possible extensions.** We close this section by noting the scope of the particular realisation of POLAR described above, and the natural extensions it suggests. First, our policy acts over finite candidate pools. This covers the pool-based settings considered in this work, but problems requiring highly precise continuous designs may benefit from continuous-action variants, for example, by optimising the learned score over generated candidates or by introducing a global summary token that feeds a continuous policy head. Second, distribution shift can affect POLAR at two stages. A mismatch between TFM pretraining and downstream policy training may degrade the learned representation and thereby the decision rule learned by the downstream policy. In particular, histories requiring different acquisition decisions may be mapped to similar representations, leaving the policy head unable to recover the lost distinction. The auxiliary prediction loss can realign the representation, while designing more transferable TFM pretraining priors is a complementary direction (Zhang et al., 2026b, Bouadi et al., 2026). A mismatch between policy training and deployment can instead render the learned policy misspecified, even when the representation remains informative. Broader or deployment-matched training distributions and misspecification-robust policy-learning methods (Tang et al., 2025, Forster et al., 2025) are promising remedies.

## 5 Related Work

Early policy-based BED used approximate dynamic programming with online design optimisation (Huan and Marzouk, 2016). Amortised policy-based BED was subsequently proposed by DAD (Foster et al., 2021), which learns a design policy offline and deploys it in a single forward pass, with iDAD (Ivanova et al., 2021) extending the framework to implicit likelihood models. A parallel line recasts the problem as reinforcement learning (Blau et al., 2022, Lim et al., 2022). Hedman et al. (2025) introduce a semi-amortised variant that combines an offline policy with online refinement, while Guo et al. (2026) consider constraint-aware BED via online planning, and a recent thread improves training signals through alternative EIG estimators (Blau et al., 2023, Iqbal et al., 2024, Shen et al., 2025, Bracher et al., 2025). ScoreBED (Phillips et al., 2026) uses approximate marginal-likelihood scores for policy learning. Related decision-theoretic formulations target downstream losses (Huang et al., 2024, 2026b, Rossa et al., 2026). Huang et al. (2026a) show that prediction-driven supervision can support policy learning, but its shared backbone is trained from scratch, making it sample-inefficient. Our work is complementary to all of these; we reuse a pretrained predictive model as the belief encoder so that a small policy head can be trained with standard policy gradient.

In BO, one line of work retains the classical surrogate-plus-acquisition template while replacing the surrogate with increasingly powerful neural predictors, including PFN- and transformer-based models (Müller et al., 2023, Chang et al., 2025, Li et al., 2025b, Hung et al., 2025, Viering et al., 2025). A second line instead learns the acquisition policy end-to-end, mapping observed trajectories directly to the next query (Chen et al., 2017, 2022, Maraval et al., 2023, Song et al., 2024, Zhang et al., 2026a, 2025b). POLAR sits between these threads: it inherits the strong representations of a pretrained surrogate, but learns a policy on top of those representations rather than committing to a fixed acquisition rule.

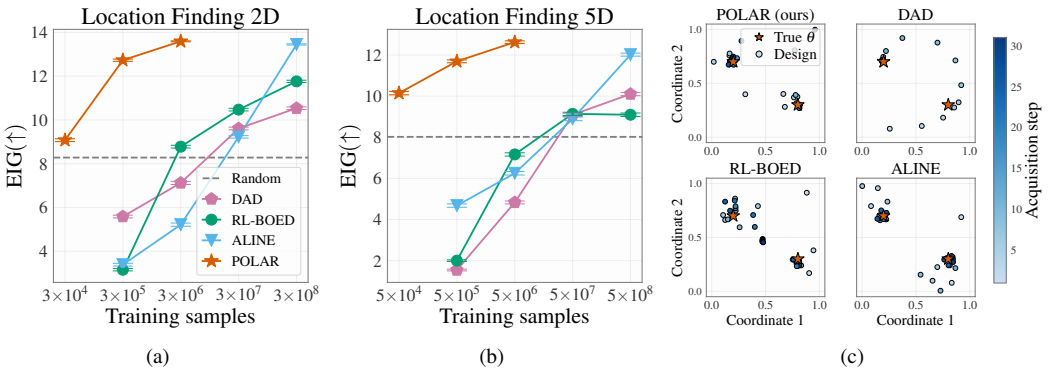
Our work is also related to the broader family of meta-learned predictors that map context sets to predictive distributions. Originating from the Conditional Neural Processes (Garnelo et al., 2018), these architectures evolved into Transformer Neural Processes (Nguyen and Grover, 2022) and early PFNs (Müller et al., 2021). Recent tabular foundation models such as TabPFN (Hollmann et al., 2023, 2025, Grinsztajn et al., 2025) and TabICL (Qu et al., 2025, 2026) scale this paradigm to broad supervised-learning regimes and have stimulated a growing literature (Müller et al., 2025, Ye et al., 2025, Hoo et al., 2024, Zhang et al., 2025a). Finally, since our backbone is adapted with an auxiliary prediction loss, our work is also related to recent studies on fine-tuning TFMs to better match downstream data distributions (Rubachev et al., 2025, Tanna et al., 2025, Garg et al., 2025).

## 6 Experiments

We now empirically evaluate POLAR across a range of tasks. Section 6.1 benchmarks POLAR on two standard BED tasks, location finding and constant elasticity of substitution, where we also report a series of ablations isolating the contribution of each component of our method. Section 6.2 evaluates POLAR for BO on a hyperparameter optimisation benchmark, and Section 6.3 turns to a real-world high-dimensional molecular optimisation task. We additionally study loss-driven active learning in Appendix C, which demonstrates that POLAR extends naturally to settings defined by user-specified predictive losses. Further experimental results and ablation studies are provided in Appendix D. Unless otherwise stated, all experiments use TabICL v2 (Qu et al., 2026) as the backbone; additional results with TabPFN v2.5 (Grinsztajn et al., 2025) are reported in Appendix D.1. The code to reproduce our experiments is available at: `https://github.com/huangdaolang/POLAR`.

### 6.1 Benchmarking on Bayesian experimental design tasks

We begin with two standard BED benchmarks, location finding and constant elasticity of substitution (CES), which have been used extensively in prior work (Foster et al., 2019, 2020, Blau et al., 2022, Huang et al., 2026a). The goal of location finding (Sheng and Hu, 2005) is to infer the locations of $K$ hidden sources in $d_x$-dimensional space from noisy distance measurements. We consider two variants to test scalability: a standard 2D setting ($d_x = 2, K = 2, T = 30$) and a higher-dimensional 5D setting ($d_x = 5, K = 2, T = 50$). CES (Arrow et al., 1961) involves eliciting the parameters of an economic utility function from a series of pairwise preference queries. We compare against Random, DAD (Foster et al., 2021), RL-BOED (Blau et al., 2022), and ALINE (Huang et al., 2026a). For both tasks, we report the EIG lower bound estimated by sPCE, and the upper bound estimated by sNMC (Foster et al., 2021). Full details on the task and training setups are provided in Appendix B.1.




(a) (b) (c)

Figure 2: **Location finding**. (a) EIG against total training samples in the 2D setting. Error bars denote standard error across 1000 runs. (b) The same comparison in the 5D setting. (c) Example design trajectories for all methods on a shared latent source configuration.

Table 1: **CES**. Comparison of EIG estimates and deployment efficiency on the CES task. Results are reported as mean ± s.e. over 1,000 independent runs.

| Methods | $10^5$ training samples | | $10^6$ training samples | | Deployment time (s) |
|---|---|---|---|---|---|
| | sPCE | sNMC | sPCE | sNMC | |
| DAD | 9.54±0.14 | 9.66±0.14 | 11.46±0.16 | 12.33±0.19 | 0.0003±0.00 |
| RL-BOED | 11.04±0.11 | 11.21±0.12 | 12.32±0.13 | 14.73±1.25 | 0.0005±0.00 |
| ALINE | 8.10±0.18 | 9.86±0.29 | 9.11±0.17 | 11.64±0.33 | 0.004±0.00 |
| POLAR | **11.15**±0.15 | **12.09**±0.19 | **13.10**±0.12 | **16.24**±0.25 | 0.03±0.00 |

**Main results.** To explicitly assess the training efficiency, we report performance as a function of the total number of training samples (see Appendix B.1 for details on how we calculate this metric for a fair comparison). For location finding, Figures 2a and 2b show that POLAR is dramatically more sample-efficient than prior amortised BED policies. In the standard 2D task, our method slightly outperforms ALINE, the current state-of-the-art, using approximately 100× fewer training samples, and surpasses the best reported results of both RL-BOED and DAD using 1,000× fewer samples. This dramatic efficiency gap further widens in the more complex 5D environment, where our method eclipses the peak performance of RL-BOED and DAD with 10,000× fewer samples, and exceeds ALINE with 100× fewer samples. This sample efficiency is particularly valuable in settings where simulation or real data collection is expensive, as is typical in scientific applications of experimental design. Figure 2c provides visualisations of design trajectories. The sNMC upper bound results are provided in Appendix D.1. The conclusion remains consistent for the CES task. Table 1 shows that POLAR outperforms all amortised baselines at matched training-sample budgets. Results at full convergence for each method are reported in Appendix D.2, where POLAR remains competitive with the strongest baseline. Owing to the cost of evaluating a foundation-model backbone, our method is slightly slower to deploy than the other policies, but the absolute overhead remains negligible in practice: at roughly 0.03 seconds per decision, the latency is still effectively imperceptible at the human timescale.

To better understand where these gains come from, we conduct three ablations on the 2D location finding task: a frozen backbone, training from scratch, and coupled policy gradients (Figure 3).

**Frozen vs. finetuned backbone.** We first compare our default setting, in which the backbone is adapted by the supervised prediction loss, against a frozen-backbone variant. Even without any weight updates, the frozen variant remains competitive with the peak performance of RL-BOED. However, it clearly falls short of our finetuned variant. Because the backbone is a general-purpose foundation model pretrained on a vast array of synthetic distributions, its representations are not tailored to a specific task. The auxiliary prediction loss acts as an alignment signal, finetuning the backbone to adapt its predictive representations to the target domain, which in turn provides a better belief state for the policy head to act upon.

**Pretrained backbone vs. training from scratch.** We next compare our method against the same architecture trained from scratch, without loading the pretrained checkpoint. The difference is substantial: training from scratch converges much more slowly and remains clearly worse within the same sample budget. This supports a central message of the paper: learning underlying belief representations is a critical part of the overall policy training for amortised adaptive design methods. A pretrained foundation model has already significantly alleviated this problem, and starting from it confers the bulk of our sample efficiency. Additional ablations (Appendix D.4) show that the same policy-head architecture performs substantially worse with randomly initialised, frozen backbones, further supporting that our gains rely on pretrained representations and cannot be explained by the policy head alone.

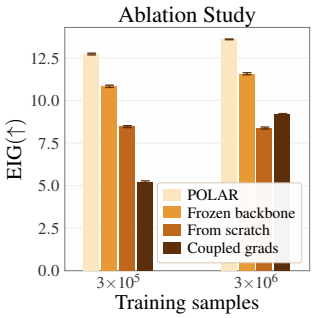



Figure 3: Ablations for 2D location finding, isolating the impact of our core architectural choices, including gradient decoupling, backbone finetuning and initialisation.

**Decoupled vs. coupled policy gradients.** Our default method updates the backbone only through the prediction loss. We compare this against a coupled variant in which policy gradients are also allowed to update the backbone. The coupled variant severely degrades performance. Once policy gradients are allowed to modify the backbone, the same parameters must simultaneously satisfy two rather different objectives: a dense, relatively stable prediction objective and a high-variance reinforcement-learning objective. In practice, this appears to destabilise the learning and weaken the benefits of pretraining. By decoupling the gradients, we effectively assign the optimal learning objective to each component: low-variance supervised learning for belief state estimation, and policy gradients for decision-making on a stable representation manifold.

Additional ablations on backbone choice and representation layer are reported in Figure A3, which shows POLAR is not tied to a single backbone, and later backbone layers provide more useful belief states for acquisition. Beyond acquisition performance, we also test whether the representations retain posterior information (Appendix D.1). A posterior decoder trained on the TFM embeddings achieves similar performance to one trained on raw observation histories.

### 6.2 Hyperparameter optimisation benchmarks

We next evaluate our method on BO using HPO-B (Arango et al., 2021), a large-scale hyperparameter optimisation benchmark containing the evaluations of a wide range of machine learning models across thousands of hyperparameter configurations and datasets. HPO-B is pre-partitioned into multiple search spaces, each corresponding to a model family with its own hyperparameter parameterisation. Following prior work (Maraval et al., 2023, Li et al., 2025b), we report results on six search spaces, spanning input dimensionalities from 2 to 16 and a representative range of model families. For each search space, we train our policy on the provided meta-training tasks and evaluate on the fixed test tasks and seeds defined by the benchmark.

We compare against baselines spanning both classical and amortised BO. RANDOM samples configurations uniformly from the candidate pool. GP is a non-amortised baseline which fits a Gaussian process (Rasmussen, 2003) to the current observations. META-GP, following Maraval et al. (2023), pretrains the kernel hyperparameters on the meta-training datasets and initialises the GP with these parameters at test time. PFNS4BO (Müller et al., 2023) uses a prior-fitted network (Müller et al., 2021) for amortised prediction. NAP (Maraval et al., 2023) learns an amortised acquisition policy using transformer neural processes (Nguyen and Grover, 2022). Finally, TABICL uses the same finetuned TabICL backbone as an amortised surrogate, but pairs it with a hand-designed acquisition function. All surrogate-based baselines use myopic EI as the acquisition strategy. Additional details on baseline configurations and the dataset are provided in Appendix B.2.

**Results.** Figure 4 shows the average normalised regret and average rank aggregated across all search spaces. Our method achieves the lowest regret and the best average rank throughout the acquisition trajectory, with the gap to the strongest baselines opening within the first ten queries and persisting through the full budget. Notably, compared with NAP, our method shows that using pretrained TFMs can yield stronger performance without requiring the policy to learn its own belief representation from scratch. Additional per-search-space regret and rank curves are reported in Appendix D.3.

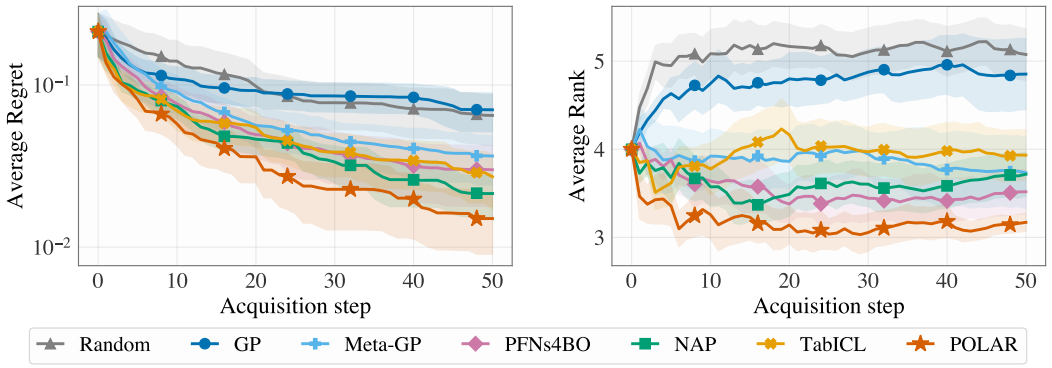



Figure 4: **Hyperparameter optimisation on HPO-B.** Average regret (left) and average rank (right) aggregated across the six search spaces. Shaded regions denote one standard error across the test tasks. POLAR achieves the lowest regret and the best average rank throughout the acquisition trajectory.

### 6.3 Molecular docking optimisation

Finally, we evaluate our method on a real-world high-dimensional molecular optimisation benchmark, DOCKSTRING (García-Ortegón et al., 2022), which provides docking scores for over 260,000 molecules against a panel of protein targets. Since the physicochemical interactions that govern binding affinity are inherently correlated across these structurally related targets, this dataset serves as a natural testbed for meta BO.

We formulate the task as identifying the molecule that minimises the docking score for a given protein target. We train our policy on a meta-training set of 8 protein targets and evaluate its zero-shot transfer performance on a held-out test set of 6 distinct targets; full target lists are provided in Appendix B.3. Each molecule is represented by a 512-bit Morgan fingerprint computed from its SMILES string.

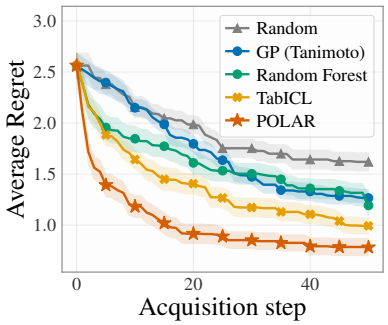



Figure 5: **DOCKSTRING**. Average regret (mean $\pm$ s.e.) across the six held-out targets.

Operating in a 512-dimensional space presents significant challenges for traditional BO. Standard GPs equipped with continuous kernels (e.g., RBF) notoriously struggle in high-dimensional, discrete spaces. Instead, we equip the GP with a Tanimoto kernel, which measures set overlap between binary fingerprints and is the standard choice for fingerprint-based molecular BO (Griffiths et al., 2023). We additionally compare against a Random Forest surrogate, a standard baseline in cheminformatics that is commonly paired with molecular fingerprints (Svetnik et al., 2003, Yang et al., 2019). For the amortised surrogate baseline, we use exactly the same finetuned TabICL backbone as in our method.

**Results.** Figure 5 reports average regret across the six test targets. POLAR achieves the lowest regret throughout the trajectory. Both POLAR and TabICL substantially outperform the non-amortised GP and Random Forest baselines, indicating that meta-training across targets allows the backbone to capture shared structure. The further gap between POLAR and TabICL confirms that learning the acquisition rule directly from the belief representation outperforms a hand-designed acquisition.

## 7 Conclusions

We introduced POLAR, an amortised framework for adaptive data acquisition based on a decision-theoretic view: policies need not act directly on raw histories or explicit posterior approximations, but can instead act on belief representations. By instantiating these representations with pretrained predictive foundation models, POLAR decouples belief representation from decision-making and trains an acquisition policy on them. Across tasks, POLAR surpasses classical non-amortised methods and state-of-the-art amortised baselines while requiring up to $100\times$ fewer training samples.

## Acknowledgments and Disclosure of Funding

DH, LA, and SK were supported by the Research Council of Finland (Flagship programme: Finnish Center for Artificial Intelligence FCAI, 359207). CH was supported by Business Finland (VirtualLab4Pharma, grant agreement 3597/31/2023) and the European Union (Horizon Europe, grant agreement 101214398, ELLIOT). LA was also supported by Research Council of Finland grants 358980 and 356498. SK was also supported by the UKRI Turing AI World-Leading Researcher Fellowship, [EP/W002973/1]. ZH is supported by the EPSRC CDT in Statistics and Machine Learning (EP/Y034813/1). TR is supported by the EPSRC grant EP/Y037200/1.

We acknowledge CSC – IT Center for Science, Finland, for computational resources provided by the LUMI supercomputer, owned by the EuroHPC Joint Undertaking and hosted by CSC and the LUMI consortium (LUMI projects 462000943 and 462000874). Access was provided through the Finnish LUMI-OKM allocation. We also acknowledge the computational resources provided by the Aalto Science-IT Project from Computer Science IT.

## References

Arango, S. P., Jomaa, H. S., Wistuba, M., and Grabocka, J. (2021). Hpo-b: A large-scale reproducible benchmark for black-box hpo based on openml. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 9, 19, 20

Arrow, K. J., Chenery, H. B., Minhas, B. S., and Solow, R. M. (1961). Capital-labor substitution and economic efficiency. *The review of Economics and Statistics*, pages 225–250. 7, 17

Berger, J. O. (1985). Statistical decision theory and bayesian analysis. *Springer Series in Statistics*. 3

Bickford Smith, F., Kirsch, A., Farquhar, S., Gal, Y., Foster, A., and Rainforth, T. (2023). Prediction-oriented bayesian active learning. In *International conference on artificial intelligence and statistics*, pages 7331–7348. PMLR. 3, 4, 6, 22

Bickford Smith, F., Kossen, J., Trollope, E., Van Der Wilk, M., Foster, A., and Rainforth, T. (2025). Rethinking aleatoric and epistemic uncertainty. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 4345–4359. PMLR. 4

Blau, T., Bonilla, E. V., Chades, I., and Dezfouli, A. (2022). Optimizing sequential experimental design with deep reinforcement learning. In *International conference on machine learning*, pages 2107–2128. PMLR. 3, 7, 18

Blau, T., Chades, I., Dezfouli, A., Steinberg, D., and Bonilla, E. V. (2023). Statistically efficient bayesian sequential experiment design via reinforcement learning with cross-entropy estimators. *arXiv preprint arXiv:2305.18435*. 3, 7

Bouadi, M., Bouarour, N., Kulkarni, V., Dubey, S., Tanna, A., and Sankarapu, V. K. (2026). Shaping the prior: How synthetic task distributions determine tabular foundation model quality. *arXiv preprint arXiv:2605.18971*. 6

Bracher, N., Kühmichel, L., Ivanova, D. R., Intes, X., Bürkner, P.-C., and Radev, S. T. (2025). Jadai: Jointly amortizing adaptive design and bayesian inference. *arXiv preprint arXiv:2512.22999*. 3, 7

Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical science*, pages 273–304. 1, 3

Chang, P. E., Loka, N. R. B. S., Huang, D., Remes, U., Kaski, S., and Acerbi, L. (2025). Amortized probabilistic conditioning for optimization, simulation and inference. In *International Conference on Artificial Intelligence and Statistics*, pages 703–711. PMLR. 1, 3, 7

Chen, Y., Hoffman, M. W., Colmenarejo, S. G., Denil, M., Lillicrap, T. P., Botvinick, M., and Freitas, N. (2017). Learning to learn without gradient descent by gradient descent. In *International Conference on Machine Learning*, pages 748–756. PMLR. 7

Chen, Y., Song, X., Lee, C., Wang, Z., Zhang, R., Dohan, D., Kawakami, K., Kochanski, G., Doucet, A., Ranzato, M., et al. (2022). Towards learning universal hyperparameter optimizers with transformers. *Advances in Neural Information Processing Systems*, 35:32053–32068. 7

Cowen-Rivers, A. I., Lyu, W., Tutunov, R., Wang, Z., Grosnit, A., Griffiths, R. R., Maraval, A. M., Jianye, H., Wang, J., Peters, J., et al. (2022). Hebo: Pushing the limits of sample-efficient hyper-parameter optimisation. *Journal of Artificial Intelligence Research*, 74:1269–1349. 20

Dawid, A. P. (1998). Coherent measures of discrepancy, uncertainty and dependence, with applications to bayesian predictive experimental design. *Department of Statistical Science, University College London. http://www. ucl. ac. uk/Stats/research/abs94. html, Tech. Rep*, 139. 4

Forster, A., Ivanova, D. R., and Rainforth, T. (2025). Improving robustness to model misspecification in bayesian experimental design. In *7th Symposium on Advances in Approximate Bayesian Inference Workshop Track*. 6

Foster, A., Ivanova, D. R., Malik, I., and Rainforth, T. (2021). Deep adaptive design: Amortizing sequential bayesian experimental design. In *International conference on machine learning*, pages 3384–3395. PMLR. 1, 3, 6, 7, 17, 18, 19

Foster, A., Jankowiak, M., Bingham, E., Horsfall, P., Teh, Y. W., Rainforth, T., and Goodman, N. (2019). Variational bayesian optimal experimental design. In *Advances in Neural Information Processing Systems*, volume 32. 7, 17

Foster, A., Jankowiak, M., O'Meara, M., Teh, Y. W., and Rainforth, T. (2020). A unified stochastic gradient approach to designing bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*, pages 2959–2969. PMLR. 7

García-Ortegón, M., Simm, G. N., Tripp, A. J., Hernández-Lobato, J. M., Bender, A., and Bacallado, S. (2022). Dockstring: easy molecular docking yields better benchmarks for ligand design. *Journal of chemical information and modeling*, 62(15):3486–3502. 10, 21

Garg, A., Ali, M., Hollmann, N., Purucker, L., Müller, S., and Hutter, F. (2025). Real-tabpfn: Improving tabular foundation models via continued pre-training with real-world data. *arXiv preprint arXiv:2507.03971*. 7

Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D., and Eslami, S. A. (2018). Conditional neural processes. In *International conference on machine learning*, pages 1704–1713. PMLR. 7

Garnett, R. (2023). *Bayesian optimization*. Cambridge University Press. 1, 3

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378. 4

Griffiths, R.-R., Klarner, L., Moss, H., Ravuri, A., Truong, S., Du, Y., Stanton, S., Tom, G., Rankovic, B., Jamasb, A., et al. (2023). Gauche: a library for gaussian processes in chemistry. *Advances in Neural Information Processing Systems*, 36:76923–76946. 10, 21

Grinsztajn, L., Flöge, K., Key, O., Birkel, F., Jund, P., Roof, B., Jäger, B., Safaric, D., Alessi, S., Hayler, A., et al. (2025). Tabpfn-2.5: Advancing the state of the art in tabular foundation models. *arXiv preprint arXiv:2511.08667*. 2, 3, 7, 16

Guo, Y., Huang, D., Zhang, X., Katt, S., Kaski, S., and Bharti, A. (2026). Constrained bayesian experimental design via online planning. *arXiv preprint arXiv:2605.26990*. 7

Hedman, M., Ivanova, D. R., Guan, C., and Rainforth, T. (2025). Step-dad: Semi-amortized policy-based bayesian experimental design. In *International Conference on Machine Learning*, pages 22904–22923. PMLR. 7

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. (2023). Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*. 6, 7

Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S. B., Schirrmeister, R. T., and Hutter, F. (2025). Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326. 2, 3, 5, 7

Hoo, S. B., Müller, S., Salinas, D., and Hutter, F. (2024). The tabular foundation model tabpfn outperforms specialized time series forecasting models based on simple features. In *NeurIPS workshop on time series in the age of large models*. 7

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*. 4, 22

Huan, X. and Marzouk, Y. M. (2016). Sequential bayesian optimal experimental design via approximate dynamic programming. *arXiv preprint arXiv:1604.08320*. 7

Huang, D., Guo, Y., Acerbi, L., and Kaski, S. (2024). Amortized bayesian experimental design for decision-making. *Advances in Neural Information Processing Systems*, 37:109460–109486. 1, 3, 7

Huang, D., Wen, X., Bharti, A., Kaski, S., and Acerbi, L. (2026a). Aline: Joint amortization for bayesian inference and active data acquisition. *Advances in Neural Information Processing Systems*, 38:54068–54102. 3, 7, 17, 18, 22

Huang, Z., Bickford Smith, F., and Rainforth, T. (2026b). Loss-driven bayesian active learning. In *The 29th International Conference on Artificial Intelligence and Statistics*. 3, 7, 22, 23, 24

Hung, Y. H., Lin, K.-J., Lin, Y.-H., Wang, C.-Y., Sun, C., and Hsieh, P.-C. (2025). Boformer: Learning to solve multi-objective bayesian optimization via non-markovian rl. In *The Thirteenth International Conference on Learning Representations*. 7

Iqbal, S., Corenflos, A., Särkkä, S., and Abdulsamad, H. (2024). Nesting particle filters for experimental design in dynamical systems. In *International Conference on Machine Learning*, pages 21047–21068. PMLR. 7, 29

Ivanova, D. R., Foster, A., Kleinegesse, S., Gutmann, M. U., and Rainforth, T. (2021). Implicit deep adaptive design: Policy-based experimental design without likelihoods. *Advances in neural information processing systems*, 34:25785–25798. 1, 3, 7, 17

Lacoste-Julien, S., Huszár, F., and Ghahramani, Z. (2011). Approximate inference for the loss-calibrated bayesian. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 416–424. JMLR Workshop and Conference Proceedings. 3

Li, C.-Y., Toussaint, M., Rakitsch, B., and Zimmer, C. (2025a). Amortized safe active learning for real-time data acquisition: Pretrained neural policies from simulated nonparametric functions. *arXiv preprint arXiv:2501.15458*. 3

Li, D., Cho, K., and Liu, C. (2025b). None to optima in few shots: Bayesian optimization with mdp priors. *arXiv preprint arXiv:2511.01006*. 1, 3, 7, 9

Lim, V., Novoseller, E., Ichnowski, J., Huang, H., and Goldberg, K. (2022). Policy-based bayesian experimental design for non-differentiable implicit models. *arXiv preprint arXiv:2203.04272*. 7

Lindley, D. V. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005. 1, 3, 4

Lindley, D. V. (1972). *Bayesian statistics: A review*. SIAM. 3

Maraval, A., Zimmer, M., Grosnit, A., and Bou Ammar, H. (2023). End-to-end meta-bayesian optimisation with transformer neural processes. *Advances in Neural Information Processing Systems*, 36:11246–11260. 1, 3, 7, 9, 19, 20, 28

Maus, N., Kim, K., Pleiss, G., Eriksson, D., Cunningham, J. P., and Gardner, J. R. (2024). Approximation-aware bayesian optimization. *Advances in Neural Information Processing Systems*, 37:21114–21140. 3

Müller, S., Feurer, M., Hollmann, N., and Hutter, F. (2023). Pfns4bo: In-context learning for bayesian optimization. In *International Conference on Machine Learning*, pages 25444–25470. PMLR. 1, 3, 7, 9, 20

Müller, S., Hollmann, N., Arango, S. P., Grabocka, J., and Hutter, F. (2021). Transformers can do bayesian inference. *arXiv preprint arXiv:2112.10510*. 3, 7, 9, 20

Müller, S., Reuter, A., Hollmann, N., Rügamer, D., and Hutter, F. (2025). Position: The future of bayesian prediction is prior-fitted. In *International Conference on Machine Learning*, pages 81861–81875. PMLR. 2, 7

Nguyen, T. and Grover, A. (2022). Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In *International Conference on Machine Learning*, pages 16569–16594. PMLR. 3, 7, 9, 20

Phillips, A., Kerrigan, G., and Rainforth, T. (2026). Bayesian experimental design via score matching. *arXiv preprint arXiv:2607.08335*. 3, 7

Qu, J., Holzmüller, D., Varoquaux, G., and Morvan, M. L. (2025). Tabicl: A tabular foundation model for in-context learning on large data. *arXiv preprint arXiv:2502.05564*. 2, 3, 5, 7, 16

Qu, J., Holzmüller, D., Varoquaux, G., and Morvan, M. L. (2026). Tabiclv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*. 2, 3, 6, 7, 16, 20

Rainforth, T., Foster, A., Ivanova, D. R., and Bickford Smith, F. (2024). Modern bayesian experimental design. *Statistical Science*, 39(1):100–114. 1, 3, 4

Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer. 9

Rossa, T., Phillips, A., and Rainforth, T. (2026). Action-bed: Task-driven bayesian experimental design with singly intractable objectives. *Advances in neural information processing systems*, 39. 4, 7, 29

Rubachev, I., Kotelnikov, A., Kartashev, N., and Babenko, A. (2025). On finetuning tabular foundation models. *arXiv preprint arXiv:2506.08982*. 7

Ryan, E. G., Drovandi, C. C., McGree, J. M., and Pettitt, A. N. (2016). A review of modern computational algorithms for bayesian optimal design. *International Statistical Review*, 84(1):128–154. 3

Savage, L. J. (1951). The theory of statistical decision. *Journal of the American Statistical association*, 46(253):55–67. 3

Savage, L. J. (1971). Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801. 4

Settles, B. (2012). *Active learning*. Morgan & Claypool Publishers. 1

Shen, W., Dong, J., and Huan, X. (2025). Variational sequential optimal experimental design using reinforcement learning. *Computer Methods in Applied Mechanics and Engineering*, 444:118068. 7

Sheng, X. and Hu, Y.-H. (2005). Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE transactions on signal processing*, 53(1):44–53. 7, 17

Song, L., Gao, C., Xue, K., Wu, C., Li, D., Hao, J., Zhang, Z., and Qian, C. (2024). Reinforced in-context black-box optimization. *arXiv preprint arXiv:2402.17423*. 7

Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., and Feuston, B. P. (2003). Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6):1947–1958. 10, 21

Tang, R., Sloman, S. J., and Kaski, S. (2025). Generalization analysis for bayesian optimal experiment design under model misspecification. *arXiv preprint arXiv:2506.07805*. 6

Tanna, A., Seth, P., Bouadi, M., Avaiya, U., and Sankarapu, V. K. (2025). Tabtune: A unified library for inference and fine-tuning tabular foundation models. *arXiv preprint arXiv:2511.02802*. 7

Trott, O. and Olson, A. J. (2010). Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461. 21

Viering, T. J., Adriaensen, S., Rakotoarison, H., Müller, S., Hvarfner, C., Hutter, F., and Bakshy, E. (2025). $\alpha$-pfn: In-context learning entropy search. In *Frontiers in Probabilistic Inference: Learning meets Sampling*. 7

Yang, K., Swanson, K., Jin, W., Coley, C., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., et al. (2019). Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388. 10, 21

Ye, H.-J., Liu, S.-Y., and Chao, W.-L. (2025). A closer look at tabpfn v2: Understanding its strengths and extending its capabilities. *arXiv preprint arXiv:2502.17361*. 7

Zhang, Q., Tan, Y. S., Tian, Q., and Li, P. (2025a). Tabpfn: One model to rule them all? *arXiv preprint arXiv:2505.20003*. 7

Zhang, X., Hassan, C., Martinelli, J., Huang, D., and Kaski, S. (2026a). In-context multi-objective optimization. In *The Fourteenth International Conference on Learning Representations*. 3, 7

Zhang, X., Huang, D., Kaski, S., and Martinelli, J. (2025b). Pabbo: Preferential amortized black-box optimization. In *The Thirteenth International Conference on Learning Representations*. 7

Zhang, X., Maddix Robinson, D., Yin, J., Erickson, N., Ansari, A. F., Han, B., Zhang, S., Akoglu, L., Faloutsos, C., Mahoney, M., et al. (2026b). Mitra: Mixed synthetic priors for enhancing tabular foundation models. *Advances in neural information processing systems*, 38:15795–15840. 6

# Appendix

The appendix is organized as follows:

- In Appendix A, we provide additional details of POLAR, including the backbone architectures, the policy head, and the training details.
- In Appendix B, we describe the experimental setups in detail, covering task specifications, baseline implementations, and evaluation protocols for the Bayesian experimental design, hyperparameter optimisation, and molecular docking benchmarks.
- In Appendix C, we extend POLAR to a loss-driven active learning task.
- In Appendix D, we report additional benchmark results and ablations examining belief representations, task-score estimation, temporal credit assignment, and transfer under controlled distribution shifts.
- In Appendix E, we provide an overview of the computational resources and software dependencies used in this work.

## A Additional details about POLAR

### A.1 Backbone

The default backbone in our experiments is the TabICLv2 regression variant (Qu et al., 2026). TabICLv2 is a transformer-based in-context learner pretrained entirely on synthetic tabular data. Architecturally, TabICLv2 inherits the three-stage compress-then-ICL design of TabICLv1 (Qu et al., 2025): a column-wise transformer that embeds each feature in isolation, a row-wise transformer that aggregates feature embeddings into a single fixed-dimensional representation per row, and a dataset-wise transformer that performs in-context learning across the entire context-query set. TabICLv2 extends this design with target-aware row embeddings, repeated feature grouping, and a query-aware scalable softmax that improves generalisation to longer contexts than those seen during pretraining; the model is pretrained with the Muon optimiser on a large corpus of synthetic datasets generated to maximise prior diversity. We use the official regression checkpoint released by the authors, which is trained to predict 999 conditional quantiles of $y^\star$ via the aggregated pinball loss. As a robustness check, we additionally evaluate POLAR with the TabPFNv2.5 backbone (Grinsztajn et al., 2025), using the public `tabpfn-v2.5-regressor-v2.5_default.ckpt` checkpoint; results are reported in Appendix D.1.

The auxiliary prediction loss aligns the backbone with the current task distribution by supervising its predictive head on target points sampled within each trajectory. The exact form depends on which quantity the backbone is trained to predict.

TabPFNv2.5 outputs a categorical distribution over discretised output bins, which we treat as an approximate predictive density. The auxiliary loss is the standard negative log-likelihood

$$\ell_{\text{pred}}^{\text{NLL}} = -\log q_\psi(y_m \mid x_m, \mathcal{D}_{t-1}). \tag{A1}$$

TabICLv2 outputs $K = 999$ conditional quantile estimates $\hat{q}_\psi^{(\tau_k)}(x_m, \mathcal{D}_{t-1})$ for evenly spaced quantile levels $\tau_k = k/(K+1)$, $k = 1, \ldots, K$. We follow the original training objective and adopt the aggregated pinball loss

$$\ell_{\text{pred}}^{\text{pinball}}\big(q_\psi(\cdot \mid x_m, \mathcal{D}_{t-1}),\, y_m\big) = \frac{1}{K}\sum_{k=1}^{K} \rho_{\tau_k}\big(y_m - \hat{q}_\psi^{(\tau_k)}(x_m, \mathcal{D}_{t-1})\big), \tag{A2}$$

where $\rho_\tau(u) = \max(\tau u, (\tau-1)u)$ is the standard pinball (tilted absolute) loss for quantile level $\tau$. In all experiments, the policy loss in Equation (6) and the auxiliary prediction loss are weighted equally, $\lambda_{\text{pol}} = \lambda_{\text{pred}} = 1$. For density-based rewards and NLPD evaluation in Appendix C, following Qu et al. (2026), we convert the predicted quantiles into a continuous distribution by enforcing monotonicity, linearly interpolating the quantile function between adjacent levels, extrapolating exponential tails beyond the extreme predicted quantiles, and then evaluating $\log q_\psi(y \mid x, \mathcal{D}) = -\log \partial_\tau Q_\psi(\tau \mid x, \mathcal{D})|_{\tau = F_\psi(y|x,\mathcal{D})}$; the predictive mean is computed as $\mu_\psi(x, \mathcal{D}) = \int_0^1 Q_\psi(\tau \mid x, \mathcal{D})\, d\tau$.

### A.2 Policy head

The default policy head used throughout the main experiments is a lightweight MLP with two hidden layers, hidden width 128, and GELU activations. It maps each candidate-conditioned representation, concatenated with the normalised remaining budget, to a scalar logit, and the policy distribution over candidates follows the softmax.

For the loss-driven active learning experiments in Appendix C, where the goal is to optimise predictive performance over a user-specified target set, we use a target-aware variant of the policy head. This variant uses a single transformer encoder layer (4 attention heads, hidden width 1024, GELU activation), allowing each candidate token to attend to the target-point representations produced by the backbone.

During training, we optimise the policy head and the backbone using AdamW. Gradients are clipped to unit global norm. We sample candidates stochastically from the policy distribution to ensure sufficient exploration during training. At test time, we instead select the argmax candidate.

## B Experimental details

### B.1 Benchmarking on Bayesian experimental design tasks

**Location finding** (Sheng and Hu, 2005) is a widely adopted benchmark for sequential BED (Foster et al., 2019, 2021, Ivanova et al., 2021, Huang et al., 2026a). The objective is to recover the positions of $K$ unknown sources in $\mathbb{R}^d$, denoted $\theta = \{\theta_k \in \mathbb{R}^d\}_{k=1}^K$, by adaptively choosing measurement locations $x \in \mathbb{R}^d$ at which noisy signal intensities are recorded. Each source radiates a signal whose magnitude attenuates with distance under an inverse-square law, so that a measurement taken at location $x$ aggregates contributions from every source:

$$\mu(\theta, x) \;=\; b \;+\; \sum_{k=1}^{K} \frac{\alpha_k}{m + \|\theta_k - x\|^2}, \tag{A3}$$

where the constants $\alpha_k$ specify the strength of each source, while $b > 0$ and $m > 0$ govern the background offset and the saturation behaviour of the signal at short range, respectively. The experimenter does not observe $\mu$ directly; instead, the log-intensity is observed under additive Gaussian noise:

$$\log y \mid \theta, x \;\sim\; \mathcal{N}\big(\log \mu(\theta, x), \sigma^2\big). \tag{A4}$$

Following standard practice, we instantiate the task with $K = 2$, $\alpha_k = 1$, $b = 0.1$, $m = 10^{-4}$, and $\sigma = 0.5$. Designs are constrained to the unit hypercube $x \in [0, 1]^d$, and the prior over each source places independent uniform distributions on every coordinate, $\theta_{k,j} \sim \mathrm{Unif}[0, 1]$ for $j = 1, \ldots, d$.

**Constant elasticity of substitution** (CES; Arrow et al., 1961) originates in economic theory and models how a consumer values bundles of goods. In the experimental design formulation, the experimenter aims to identify a participant's latent preference structure by repeatedly presenting them with pairs of baskets and recording a (potentially noisy) judgement about their relative desirability. A basket $c \in [0, 100]^K$ encodes the quantities held of each of $K$ goods. The participant's preferences are captured by latent parameters $\theta = (\rho, \boldsymbol{\alpha}, u)$, where $\rho \in (0, 1)$ determines the degree of substitutability between goods, $\boldsymbol{\alpha} \in \Delta^{K-1}$ assigns simplex-valued weights to the goods, and $u > 0$ modulates the overall responsiveness of the participant's reports. A single experimental design takes the form of a basket pair $x = (c, c') \in [0, 100]^{2K}$, and elicits a bounded preference rating $y \in [0, 1]$.

The utility assigned to a basket follows the canonical CES form: $U(c) \;=\; \Big(\sum_{i=1}^K c_i^\rho\, \alpha_i\Big)^{\frac{1}{\rho}}$. We adopt the following prior specification over the latent parameters:

$$\rho \sim \mathrm{Beta}(1, 1), \qquad \boldsymbol{\alpha} \sim \mathrm{Dirichlet}(\mathbf{1}_K), \qquad \log u \sim \mathcal{N}(1, 3^2). \tag{A5}$$

For a given query $x = (c, c')$ and parameter setting $\theta$, the underlying (unobserved) utility differential is generated from

$$\eta \sim \mathcal{N}\Big(u\,(U(c) - U(c')),\; u^2\,\tau^2\,(1 + \|c - c'\|)^2\Big), \tag{A6}$$

and then transformed through a sigmoid link and clipped to a bounded interval to yield the reported outcome: $y \;=\; \mathrm{clip}\big(\sigma(\eta),\; \epsilon,\; 1 - \epsilon\big)$, with $\sigma(\cdot)$ the standard sigmoid. We use $K = 3$, $\tau = 0.005$,

$\epsilon = 2^{-22}$, and query budget $T = 10$ in all experiments. Since TFMs are pretrained on standardised inputs and outputs, we standardise the features before feeding them to the backbone.

### B.1.1 Baselines

**Deep Adaptive Design** (Foster et al., 2021) learns an amortised design policy by directly maximising the sPCE lower bound on the total EIG over $T$-step trajectories. The policy network consists of a two-layer MLP encoder with hidden width 128 that maps each design-observation pair $(x_i, y_i)$ to a 16-dimensional embedding, a permutation-invariant pooling operation that aggregates these embeddings into a history representation, and a one-layer MLP emitter that maps the pooled representation to the next design. We train with Adam (learning rate $5 \times 10^{-5}$, $\beta = (0.8, 0.998)$), gradient clipping at 1.0, and an exponential learning-rate decay with factor 0.98 every 1000 epochs. For location finding, we use $L = 2 \times 10^4$ contrastive samples for sPCE estimation during training. For CES, we use $L = 10^5$.

**RL-BOED** (Blau et al., 2022) formulates sequential BED as a hidden-parameter Markov decision process and learns an amortised design policy using reinforcement learning. The reward is defined stepwise as the marginal contribution of the newly acquired design-observation pair to the sPCE. We use Randomized Ensembled Double Q-learning (REDQ) to train the continuous-design policy. The history encoder follows a DAD-style permutation-invariant architecture: each concatenated design-observation pair $(x_i, y_i)$ is passed through a two-layer MLP with 128 hidden units and ReLU activations, followed by a 64-dimensional linear output layer; the resulting embeddings are summed to form the history representation. The policy emitter outputs the mean and log-variance of independent Tanh-Gaussian distributions over the design dimensions. The critic networks use the same history encoder and concatenate the encoded history with the candidate design before passing it through a two-layer 128-unit ReLU MLP. We train with $L = 10^5$ contrastive samples for the sPCE reward. The remaining RL hyperparameters are listed in Table A1.

Table A1: Hyperparameters used for RL-BOED.

| Parameter | Location Finding | CES |
|---|---|---|
| Critics $N$ | 2 | 2 |
| Random target subset $M$ | 2 | 2 |
| Discount factor $\gamma$ | 0.9 | 0.9 |
| Target update rate $\tau$ | $10^{-3}$ | $5 \times 10^{-3}$ |
| Policy learning rate | $10^{-4}$ | $3 \times 10^{-4}$ |
| Critic learning rate | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ |
| Replay buffer size | $10^7$ | $10^6$ |
| Minimum buffer size | $10^5$ | $10^5$ |
| Entropy-temperature learning rate | $3 \times 10^{-4}$ | $3 \times 10^{-4}$ |

**ALINE** (Huang et al., 2026a) jointly amortises sequential design and posterior inference with a shared masked transformer architecture. Each design $x$ and observation $y$ is embedded by separate two-layer MLPs with hidden width 128 into 32-dimensional representations; context tokens are formed by summing the design and observation embeddings, candidate-query designs are represented by design embeddings alone, and one learnable target token is introduced for each latent parameter dimension. These tokens are processed by a 3-layer transformer encoder with model width 32, feedforward width 128, and 4 attention heads. A one-hidden-layer MLP acquisition head maps the query representations to a categorical distribution over the candidate design pool, while a separate one-hidden-layer MLP posterior head predicts a 10-component Gaussian mixture for each target parameter token. We train ALINE with a joint objective consisting of a negative log-likelihood term for posterior prediction and a REINFORCE-style design loss based on one-step improvements in target log-likelihood, weighted equally with $\alpha = 1$ and discount factor $\gamma = 1$. For the sample-efficiency experiments, we use AdamW with cosine annealing and layer-wise learning rates of $10^{-3}$ for the acquisition and posterior heads and $2 \times 10^{-4}$ for the shared embedder and transformer.

### B.1.2 Evaluation details

Following Foster et al. (2021), we evaluate all amortised policies using the sequential Prior Contrastive Estimation (sPCE) lower bound and the sequential Nested Monte Carlo (sNMC) upper bound on the

total expected information gain. Given a sampled trajectory $(\theta_0, h_T) \sim p(\theta, h_T \mid \pi)$ together with $L$ contrastive samples $\theta_{1:L} \sim p(\theta)$ drawn independently from the prior, the sPCE and sNMC estimators are defined as

$$\mathcal{L}_T(\pi, L) = \mathbb{E}\left[\log \frac{p(h_T \mid \theta_0, \pi)}{\frac{1}{L+1}\sum_{\ell=0}^{L} p(h_T \mid \theta_\ell, \pi)}\right], \quad \mathcal{U}_T(\pi, L) = \mathbb{E}\left[\log \frac{p(h_T \mid \theta_0, \pi)}{\frac{1}{L}\sum_{\ell=1}^{L} p(h_T \mid \theta_\ell, \pi)}\right], \tag{A7}$$

where the outer expectation is taken over $\theta_0, h_T \sim p(\theta, h_T \mid \pi)$ and $\theta_{1:L} \sim p(\theta)$. Both bounds become tight as $L \to \infty$ at a rate $O(L^{-1})$ (Foster et al., 2021). For training, POLAR uses $L = 10^5$ contrastive samples for both tasks. For the evaluations, we use $L = 10^6$ contrastive samples for location finding and $L = 10^7$ for CES.

We report performance against the total number of training samples consumed during policy learning, where a "sample" denotes a single design-observation pair $(x, y)$ produced by the simulator. This convention provides a uniform unit of comparison that is independent of architectural choices such as batch size, replay-buffer capacity, or auxiliary loss terms.

For DAD and ALINE, every gradient step consumes one fresh batch of trajectories of length $T$, so the cumulative number of training samples after $E$ epochs is $E \cdot B \cdot T$, where $B$ is the batch size and $T$ is the experiment horizon. For RL-BOED, which maintains a replay buffer, only the trajectories newly synthesised at each epoch contribute to the sample count; the batch size of the RL update governs how many transitions are drawn from the buffer per gradient step, but does not generate new simulator data, and is therefore excluded from the accounting. Concretely, if RL-BOED rolls out $B_{\text{env}}$ new trajectories per epoch, the cumulative count after $E$ epochs is $E \cdot B_{\text{env}} \cdot T$.

For POLAR, the auxiliary prediction loss requires evaluating the backbone on $M$ target points per trajectory in addition to the $T$ design-observation pairs visited by the policy. Each target point corresponds to a fresh sample drawn from the simulator and therefore contributes to the total sample budget. We thus account for both sources, giving a per-epoch consumption of $B \cdot (T + M)$ and a cumulative count of $E \cdot B \cdot (T + M)$ after $E$ epochs. We use $M = 20$ target points and $B = 50$ trajectories per update for location finding, and $M = 10$ and $B = 20$ for CES. Our sample-efficiency comparison measures target-task simulator samples consumed during policy learning, conditional on access to an existing pretrained backbone; it does not include foundation-model pretraining. We consider this accounting appropriate because the backbones are pretrained entirely on synthetic tabular data that is task-agnostic, cheap to generate, and shared across all downstream applications. Once pretrained, the same checkpoint is reused without modification across BED, BO, and AL tasks, so the pretraining cost is amortised across all downstream uses rather than attributable to any single task. The reported gains should be interpreted as improvements in target-task sample efficiency, rather than reductions in total data or compute over the full pretraining-and-adaptation lifecycle.

Both POLAR and ALINE operate over a finite candidate pool at each acquisition step, whereas DAD and RL-BOED produce continuous designs directly. To match the original formulations of all baselines while keeping the pool-based methods well-resourced, we use a candidate pool of $|\mathcal{C}| = 2000$ designs sampled uniformly from the design space at the start of each trajectory for location finding, and $|\mathcal{C}| = 20000$ for CES. During training, POLAR uses smaller pools of $|\mathcal{C}| = 1000$ and $|\mathcal{C}| = 2000$, respectively.

### B.2 Hyperparameter optimisation

#### B.2.1 Task description

HPO-B (Arango et al., 2021) is a large-scale meta-dataset for black-box hyperparameter optimisation (HPO) assembled from the OpenML repository. It contains 6.4 million hyperparameter evaluations. We use the curated HPO-B-v3 split, which retains the 16 search spaces and provides predefined meta-train, meta-validation, and meta-test partitions, together with five fixed initialisation seeds per test task to support reproducible comparisons. Each search space corresponds to the hyperparameter space of a particular machine-learning algorithm (e.g., SVM, XGBoost, glmnet), and each meta-dataset within a search space corresponds to evaluations on a specific tabular dataset. The hyperparameter ranges are normalised to $[0, 1]^d$ and categorical hyperparameters are one-hot encoded. Following Maraval et al. (2023), we evaluate on six search spaces that together cover a representative range of underlying model families and input dimensionalities: `glmnet` (search space 5860, $d = 2$), `rpart.preproc`

(search space 4796, $d = 3$), `rpart` (search space 5859, $d = 6$), `ranger` (search space 5889, $d = 6$), `svm` (search space 5527, $d = 8$), and `xgboost` (search space 5906, $d = 16$). We train on the provided meta-training datasets and evaluate on the held-out meta-test tasks. We adopt the recommended evaluation protocol of Arango et al. (2021): each test task is run with the five fixed initialisation seeds for $T = 50$ acquisition steps, and we report the average normalised regret and average rank as defined in Arango et al. (2021).

#### B.2.2 Baselines

**GP.** A standard Gaussian process surrogate fitted directly on the test task, with no use of meta-training data. We use an exact GP with a constant mean, a scaled RBF kernel, and a Gaussian likelihood. The kernel and noise hyperparameters are optimised by maximizing the exact marginal likelihood on the observed configurations after each BO round.

**Meta-GP.** Following Maraval et al. (2023), a meta-trained Gaussian process pretrains the RBF kernel hyperparameters on the meta-training datasets of the corresponding search space and uses these pretrained values to initialise the GP at test time. Subsequent acquisitions then refit the kernel hyperparameters on the growing test-time observation set, starting from this meta-learned initialisation rather than from a generic prior.

**PFNs4BO (Müller et al., 2023).** PFNs4BO uses Prior-data Fitted Networks (Müller et al., 2021) as transformer-based surrogates for BO: a single transformer is meta-trained offline to approximate the posterior predictive distribution of a chosen prior over functions, after which inference for any new context set is performed in a single forward pass without further fitting. The strongest variant of the model is trained on the so-called HEBO+ prior, which augments the HEBO Gaussian process prior (Cowen-Rivers et al., 2022) with input and output warping and additional hyperparameter randomisation, yielding a surrogate that has been shown to outperform standard GP surrogates on the HPO-B benchmark. We use the publicly released `hebo-plus-model` checkpoint from the official PFNs4BO repository without any further finetuning on the meta-training tasks.

**NAP (Maraval et al., 2023).** NAP is an end-to-end differentiable meta-BO framework that jointly meta-learns a transformer-based neural process (Nguyen and Grover, 2022) surrogate and an acquisition function via reinforcement learning, with an auxiliary supervised likelihood loss that stabilises training and provides an inductive bias toward valid probabilistic predictions. NAP is the closest amortised baseline to POLAR, in that both methods learn the acquisition rule rather than relying on a hand-designed one. Since the HPO-B search spaces, test tasks, and initialisation seeds used by Maraval et al. (2023) are identical to ours, we directly use the per-step regret values reported in their official release for fair comparison.

**TabICL (Qu et al., 2026).** A surrogate-based ablation that uses the same finetuned TabICL backbone as POLAR, but discards the learned policy head and instead pairs the backbone's predictive distribution with the EI acquisition function. This baseline isolates the contribution of learning the acquisition rule from the contribution of the meta-learned belief representation: the difference between TabICL and POLAR can be attributed entirely to replacing a hand-designed acquisition with one trained on top of the same belief representation.

#### B.2.3 Evaluation details

For each HPO-B test task, all methods start from the same official initialisation and are evaluated on the same candidate pool. We report the normalised regret

$$r_t = \frac{f^\star - f_t^{\text{best}}}{f^\star - f_{\min}}, \tag{A8}$$

where $f_t^{\text{best}}$ is the best objective value observed by the method up to step $t$, $f^\star$ is the best value in the full task pool, and $f_{\min}$ is the worst value in the same pool. We average normalised regret over test datasets and initialisation seeds.

We also report the normalised rank to compare methods across tasks with different regret scales. For each task, seed, and optimisation step, methods are ranked by their normalised regret, with lower

regret receiving a better rank and ties assigned the average rank. These ranks are then averaged over datasets, seeds, and search spaces.

### B.3 Molecular docking optimisation

#### B.3.1 Task description

DOCKSTRING (García-Ortegón et al., 2022) is a benchmark for molecular optimisation built around AutoDock Vina (Trott and Olson, 2010) docking simulations against a curated panel of medically relevant protein targets. The full dataset comprises docking scores and binding poses for over 260000 drug-like molecules evaluated against 58 targets, where each docking score quantifies the predicted binding affinity between a ligand and a target. Lower scores indicate stronger predicted binding, so the optimisation objective is to identify molecules that minimise the docking score for a given target. We frame this as a pool-based BO problem. Each molecule is represented by its SMILES string and converted to a 512-bit Morgan fingerprint with radius 2. We randomly subsample 10000 molecules from the full dataset and use this fixed subset throughout all experiments. We split the 14 protein targets used in our experiments into a meta-training set of 8 targets, JAK2, KIT, LCK, SRC, IGF1R, ABL1, MET, EGFR, and a held-out test set of 6 targets, PTK2, FGFR1, CSF1R, CDK2, MAPK14, KDR. All 14 are protein kinases, which share a conserved ATP-binding pocket and substantial structural homology across the family. This shared geometry means that the physicochemical features predictive of strong binding are correlated across targets, making the kinase family a natural setting for meta-learning.

#### B.3.2 Baselines

**GP with Tanimoto kernel.** Standard continuous kernels such as the RBF or Matérn kernels are poorly suited to high-dimensional binary fingerprint inputs, where Euclidean distance is a weak measure of chemical similarity. We instead equip the GP with the Tanimoto kernel, a standard choice for fingerprint-based molecular BO (Griffiths et al., 2023). For two binary fingerprints $x, x' \in \{0,1\}^d$, the Tanimoto kernel is defined as

$$k_{\text{Tanimoto}}(x, x') = \sigma^2 \cdot \frac{\langle x, x' \rangle}{\|x\|^2 + \|x'\|^2 - \langle x, x' \rangle}, \tag{A9}$$

where $\sigma^2$ is a learnable output scale and $\langle \cdot, \cdot \rangle$ denotes the inner product. The kernel returns 1 for identical fingerprints and 0 for fingerprints with no shared bits, providing a meaningful similarity signal for sparse binary representations. We refit the kernel hyperparameters by maximum marginal likelihood after each acquisition step.

**Random Forest.** Random forests are a standard surrogate in cheminformatics and are commonly paired with molecular fingerprints (Svetnik et al., 2003, Yang et al., 2019). We use an ensemble of 256 regression trees with bootstrapped training sets and the standard $\sqrt{d}$ feature-subsampling rule at each split. The surrogate is refit on the full set of observed molecules at each acquisition step.

**TabICL.** We use the same setup as we used in our HPO-B experiments, where the same finetuned TabICL backbone as POLAR is used but replaces the learned policy head with a hand-designed acquisition function.

#### B.3.3 Evaluation protocol

For each of the 6 held-out test targets, we evaluate all methods under the same protocol: 10 randomly sampled molecules form the initial context, after which the policy selects $T = 50$ acquisitions, with a candidate pool of size $|\mathcal{C}| = 2000$. We report the average regret and aggregate by averaging across the 6 test targets and 5 random seeds for each test target.

## C Loss-driven active learning

In this section, we evaluate POLAR on a loss-driven active learning task, where the goal is to select query points that improve predictive performance on a downstream task distribution $p^\star$ under a user-specified loss.

Active learning seeks to acquire labelled data that maximises the predictive performance of a downstream model. In practice, a clinician may care primarily about predicting high-risk patients accurately, a structural engineer about responses near a failure threshold, and a drug discovery pipeline about molecules with high binding affinity. Standard information-theoretic acquisition objectives such as BALD (Houlsby et al., 2011) and EPIG (Bickford Smith et al., 2023) treat all outcomes symmetrically and therefore cannot directly target such loss-weighted criteria. Huang et al. (2026b) formalise this gap and derive a principled extension via weighted Bregman divergences, yielding myopic acquisition rules such as weighted variance reduction (GP-VR$_w$) that explicitly target a user-specified weighted loss. We adopt the same loss-driven framing but instantiate it in our amortised setting.

Concretely, we consider pool-based active learning under a weighted loss $\ell_\omega(y, q) = \omega(y)\ell(y, q)$, where $\omega : \mathcal{Y} \to \mathbb{R}_+$ encodes which outcomes matter most for the downstream task. At round $t$, the learner observes a context $\mathcal{D}_{t-1} = \{(x_i, y_i)\}_{i=1}^{t-1}$, selects a query $x_t$ from a finite pool $\mathcal{C}_t$, and observes $y_t \sim p(y \mid x_t, \theta)$ where $\theta$ is the latent function. After $T$ rounds, performance is evaluated on a held-out target set drawn from the target input distribution $p_\star(x^\star)$.

POLAR instantiates this setting using the same two-component architecture as in our other experiments. TFM plays the role of an amortised surrogate model: given the current context $\mathcal{D}_{t-1}$ and a query input $x^\star$, a single forward pass yields the approximate posterior predictive $q_\psi(y^\star \mid x^\star, \mathcal{D}_{t-1})$, replacing the GP surrogate that classical AL methods refit at every round. The policy head then scores each candidate in $\mathcal{C}_t$ from the same forward pass and samples the next query $x_t$. For the utility, we use the weighted one-step improvement in predictive log-density on the target set,

$$u_t = \mathbb{E}_{x^\star \sim p^\star}\big[\omega(y^\star)\big(\log q_\psi(y^\star \mid x^\star, \mathcal{D}_t) - \log q_\psi(y^\star \mid x^\star, \mathcal{D}_{t-1})\big)\big], \tag{A10}$$

which measures the weighted improvement in the backbone's own predictive log-density on the target set after observing the new query, and which generalises the dense sEPIG reward of Huang et al. (2026a) from the unweighted to the weighted setting. In our experiments, we adopt the exponential weighting $\omega(y) = \exp(\beta y)$ with $\beta = 10$, prioritising regions of high outcome value. At deployment, no GP fitting or acquisition optimisation is required: the next query is produced in a single forward pass per round.

### C.1 Training distribution

We train POLAR on synthetic functions drawn from a distribution over GPs with randomised kernels and hyperparameters, following a procedure similar to ALINE (Huang et al., 2026a). Each task is sampled by:

1. Drawing a kernel uniformly from {RBF, Matérn-3/2, Matérn-5/2};
2. Drawing a length-scale $\ell \sim \text{LogUniform}(0.1, 2.0)$ and an output scale $\sigma_f \sim \text{Uniform}(0.1, 1.0)$;
3. Sampling a function $f \sim \mathcal{GP}(0, k_{\ell,\sigma_f})$.

Before each forward pass, we normalise each function using task-level statistics computed independently of the observed context: for inputs, we use the known domain bounds, and for outputs, we estimate the normalisation mean and variance from a large set of reference points sampled from that function. The horizon is $T = 20$, with an initial context of 2 points and a candidate pool of size $K = 200$.

### C.2 Baselines

We compare against four non-amortised GP baselines.

**GP-RS** (random sampling) selects $x_t$ uniformly at random from the pool.

**GP-US** (uncertainty sampling) selects the candidate with the highest posterior predictive variance,

$$\text{US}(x) = \mathbb{V}[y \mid x, \mathcal{D}_{t-1}]. \tag{A11}$$

**GP-VR** (variance reduction) selects the candidate that maximally reduces total predictive variance over the target set $\{x_m^\star\}_{m=1}^M$:

$$\mathrm{VR}(x) = \sum_{m=1}^{M} \frac{\mathrm{Cov}_{t-1}(x_m^\star, x)^2}{\mathbb{V}[y \mid x, \mathcal{D}_{t-1}]}, \tag{A12}$$

where $\mathrm{Cov}_{t-1}(x^\star, x)$ is the GP posterior covariance between the latent function values at $x^\star$ and $x$ given $\mathcal{D}_{t-1}$.

**GP-VR**$_\omega$ (weighted variance reduction; Huang et al., 2026b) is the loss-matched counterpart of GP-VR, derived from the same weighted-Bregman framework as our reward. It targets the weighted predictive variance under the reweighted predictive $p_\omega(y \mid x^\star) \propto \omega(y)p(y \mid x^\star)$ rather than the unweighted variance.

### C.3 Evaluations

We evaluate on two distributions of test functions.

**GP synthetic.** We sample 100 functions from the same generative procedure used during training. This measures in-distribution performance.

**Benchmark functions.** We additionally evaluate on four standard regression benchmarks to measure out-of-distribution generalisation: Forrester, Gramacy-Lee, Higdon, and Sine-Gaussian Bumps. The functional forms are:

$$\begin{aligned}
f_{\text{Forr}}(x) &= (6x-2)^2 \sin(12x-4), \\
f_{\text{GL}}(x) &= \frac{\sin(10\pi x)}{2x} + (x-1)^4, \\
f_{\text{Higdon}}(x) &= \sin(2\pi x/10) + 0.2\sin(2\pi x/2.5), \\
f_{\text{SGB}}(x) &= 2\sin(2x) + 8\phi_{2.5,0.5}(x) + 10\phi_{7.5,0.25}(x) - 6\phi_{-4.5,0.5}(x).
\end{aligned}$$

where $\phi_{\mu,\sigma}$ is a Gaussian density with mean $\mu$ and standard deviation $\sigma$. Before evaluation, we apply a fixed per-function normalisation: inputs are centred and scaled using the benchmark domain, and outputs are standardised using statistics estimated from Sobol samples over the same domain. All methods, including GP and POLAR, are evaluated on these same normalized benchmark versions.

For each benchmark function, we run 100 independent seeds. Each seed draws an initial context of 2 points, a candidate pool of 200 points, and a target set of 100 points; all three sets are sampled uniformly. Performance is reported as weighted RMSE and weighted NLPD on the target set:

$$\mathrm{RMSE}_\omega = \sqrt{\frac{\sum_{m=1}^M \omega(y_m^\star)(y_m^\star - \mu(x_m^\star))^2}{\sum_{m=1}^M \omega(y_m^\star)}}, \qquad \mathrm{NLPD}_\omega = -\frac{\sum_{m=1}^M \omega(y_m^\star)\log q(y_m^\star \mid x_m^\star)}{\sum_{m=1}^M \omega(y_m^\star)}, \tag{A13}$$

where $\mu(x^\star)$ is the predictive mean under the surrogate at the end of the acquisition trajectory.

### C.4 Results

Table A2 reports final-step weighted RMSE and weighted NLPD on both evaluation distributions. First, the two loss-aware methods GP-VR$_\omega$ and POLAR substantially outperform the loss-agnostic baselines (GP-RS, GP-US, GP-VR) on both metrics and both evaluation distributions. This confirms the central message of Huang et al. (2026b): when the downstream loss assigns non-uniform weight to outcomes, acquisition rules that ignore the weighting are systematically suboptimal. Second, POLAR consistently outperforms GP-VR$_\omega$ across both evaluation distributions, suggesting that learning the acquisition rule directly from data outperforms a hand-designed one. Figure A1 visualises the acquisition behaviour of GP-VR, GP-VR$_\omega$, and POLAR on the Sine-Gaussian Bumps function from Huang et al. (2026b).

Table A2: **Loss-driven active learning**. Final-step weighted RMSE and weighted NLPD on GP synthetic functions (in-distribution) and benchmark functions (out-of-distribution).

| Methods | GP Synthetic | | Benchmark Functions | |
|---|---|---|---|---|
| | $\text{RMSE}_\omega$ (↓) | $\text{NLPD}_\omega$ (↓) | $\text{RMSE}_\omega$ (↓) | $\text{NLPD}_\omega$ (↓) |
| GP-RS | $0.40 \pm 0.06$ | $-0.18 \pm 0.28$ | $0.74 \pm 0.06$ | $3.09 \pm 0.78$ |
| GP-US | $0.13 \pm 0.02$ | $-1.26 \pm 0.16$ | $0.45 \pm 0.05$ | $4.20 \pm 0.75$ |
| GP-VR | $0.17 \pm 0.03$ | $-1.18 \pm 0.16$ | $0.33 \pm 0.03$ | $3.30 \pm 0.78$ |
| $\text{GP-VR}_\omega$ | $0.11 \pm 0.02$ | $-1.65 \pm 0.15$ | $0.15 \pm 0.02$ | $-0.83 \pm 0.33$ |
| POLAR (ours) | $\mathbf{0.09} \pm 0.03$ | $\mathbf{-2.07} \pm 0.13$ | $\mathbf{0.05} \pm 0.01$ | $\mathbf{-1.68} \pm 0.10$ |

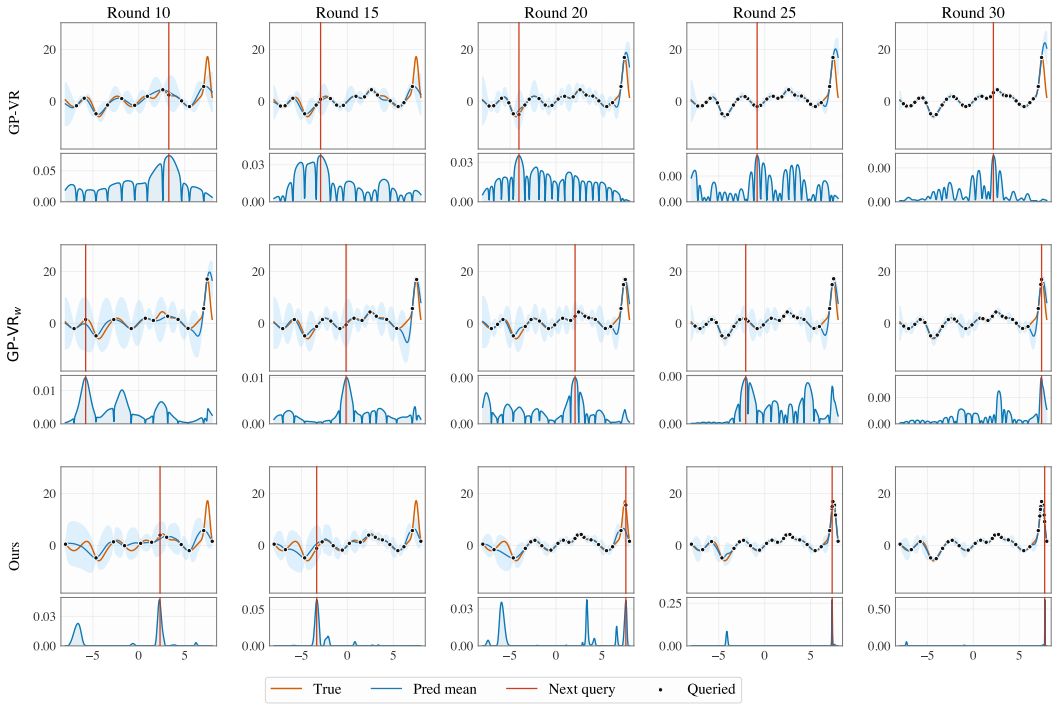



Figure A1: **Loss-driven active learning**. Acquisition behaviour on the Sine-Gaussian Bumps function from Huang et al. (2026b). The loss-agnostic GP-VR (top) spreads queries across the domain, while $\text{GP-VR}_\omega$ and POLAR (middle, bottom) concentrate queries near the high-value region.

# D Additional experimental results

## D.1 Location finding

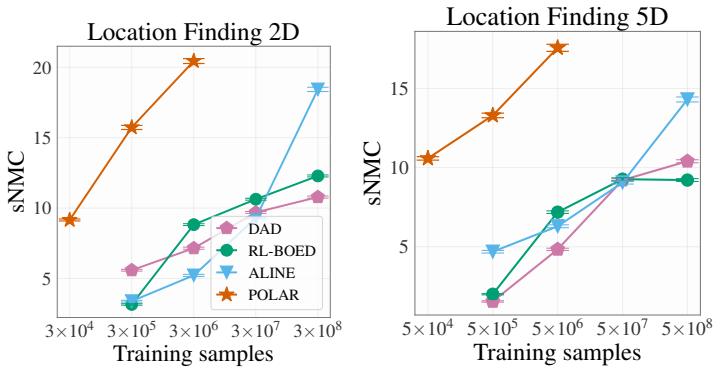
**sNMC results.** Figure A2(a) and Figure A2(b) show the sNMC upper bound on EIG against total training samples in the 2D and 5D settings, respectively. The relative ordering of methods mirrors the sPCE comparison in Figure 2.

**Backbone choice.** Figure A3(a) examines whether POLAR's gains depend critically on the choice of pretrained backbone by replacing TabICLv2 with TabPFNv2.5 on the 2D task. The TabPFN-backed policy converges more slowly during the early stages of training, but the two variants reach essentially the same EIG at convergence. This suggests that POLAR is not tied to a particular backbone: as long as the pretrained model produces representations from which the belief can be approximately decoded, a lightweight policy head can turn them into an effective acquisition policy.

**Representation layer.** Another natural question is which layer of the backbone supplies the most informative belief representation for policy learning. To test this, we re-train POLAR on the 2D task

Table A3: Held-out sPCE (mean ± s.e.) on 2D location finding with different training contrastive-sample budgets.

| Training samples | $L = 10^2$ | $L = 10^3$ | $L = 10^4$ | $L = 10^5$ |
|---|---|---|---|---|
| $3 \times 10^4$ | $8.50 \pm 0.07$ | $8.55 \pm 0.07$ | $9.46 \pm 0.07$ | $\mathbf{9.53} \pm 0.06$ |
| $3 \times 10^5$ | $9.14 \pm 0.05$ | $9.80 \pm 0.06$ | $12.09 \pm 0.06$ | $\mathbf{12.57} \pm 0.05$ |
| $3 \times 10^6$ | $9.75 \pm 0.06$ | $12.67 \pm 0.06$ | $13.58 \pm 0.02$ | $\mathbf{13.60} \pm 0.02$ |




(a) (b)

Figure A2: **Location finding**. (a) EIG upper bound (sNMC) against total training samples in the 2D setting. Error bars denote standard error across 1000 runs. (b) The same comparison in the 5D setting.

using representations taken from the last, second-to-last, fourth-to-last, and eighth-to-last transformer layers of the TabICLv2 backbone, keeping all other components fixed. Figure A3(b) shows the last and second-to-last layers perform comparably throughout training, with the second-to-last layer slightly ahead in the low-sample regime and the last layer marginally ahead at convergence. Earlier layers degrade performance monotonically. A natural concern is that the layer ordering in Figure A3(b) reflects the effect of finetuning rather than the structure of the pretrained backbone. Figure A3(c) repeats the comparison with a frozen backbone and shows the same ordering, confirming that later layers provide more informative belief representations regardless of whether the backbone is adapted. Therefore, we adopt the last layer as the default in all other experiments.

**Posterior decoding.** We investigate whether POLAR's representations retain information about the latent parameters. After freezing the trained backbone and acquisition policy, we extract the final-layer history embeddings following 30 acquisitions. Four learned coordinate tokens cross-attend to these detached embeddings, and separate heads predict a 10-component Gaussian mixture for each coordinate, trained by marginal maximum likelihood. Sources are ordered by their first coordinate to resolve label-exchange ambiguity. As a control, a capacity-matched estimator encodes the corresponding raw histories with a small Transformer followed by an analogous marginal decoder, using the same trajectory split. On 2,000 held-out trajectories, both estimators achieve a posterior-mean RMSE of 0.04, compared with an expected RMSE of 0.26 for the canonicalised prior mean. Figure A4 shows a selected held-out example where both estimators localise the sources accurately. Together, these results provide empirical evidence that the frozen representations retain posterior information.

**Sensitivity to the training-score estimator.** We vary the number of contrastive samples used in the sPCE training objective, $L_{\text{train}} \in \{10^2, 10^3, 10^4, 10^5\}$. Table A3 shows that larger contrastive budgets improve policy learning at matched target-task sample budgets, with $10^4$ and $10^5$ attaining similar final performance.

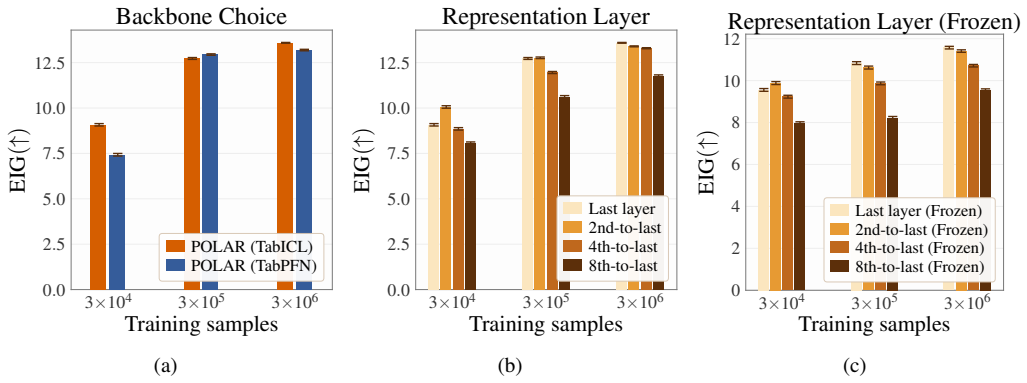



Figure A3: **Location finding**. (a) Ablation study on the backbone choice. (b) Ablation study on the representation layer with backbone finetuning. (c) Ablation study on the representation layer with a frozen backbone.

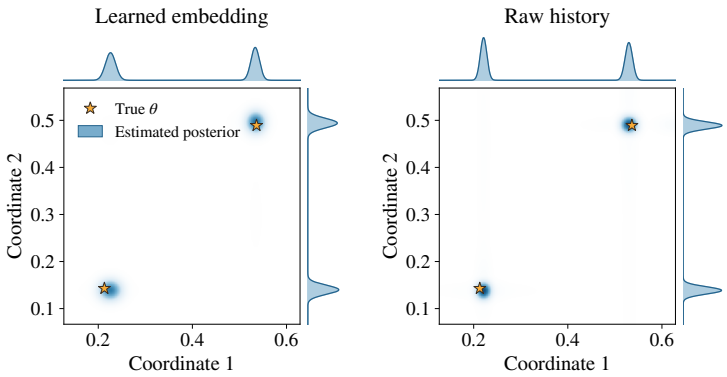



Figure A4: **Posterior decoding.** Factorised source-location densities from frozen POLAR embeddings (left) and raw history (right), for the same held-out trajectory after 30 acquisitions. Stars mark true source locations.

### D.2 Constant elasticity of substitution

Table A4 reports each method's best-attained performance. POLAR achieves the second strongest sPCE among all methods, close to the strongest baseline RL-BOED. During training, we observed that both DAD and RL-BOED were prone to instability: their training losses diverged in the late stages, with sPCE peaking and then degrading. We therefore applied early stopping based on validation sPCE for both baselines and report the checkpoint at which validation sPCE was highest. A separate observation concerns the sPCE–sNMC gap. For RL-BOED, this gap is markedly larger than for the other methods. A wide gap at finite contrastive samples indicates that the EIG bounds are loose for this particular policy, rather than that the true EIG is correspondingly large.

Table A4: **CES**. Best performance comparison on the CES task. Results are reported as mean ± s.e. over 1,000 independent runs.

| Methods | sPCE | sNMC |
|---|---|---|
| Random | 9.18±0.18 | 11.86±0.34 |
| DAD | 12.87±0.14 | 16.44±0.39 |
| RL-BOED | 14.19±0.09 | 33.33±1.80 |
| ALINE | 13.43±0.10 | 18.13±0.28 |
| POLAR | 14.11±0.09 | 20.09±0.47 |

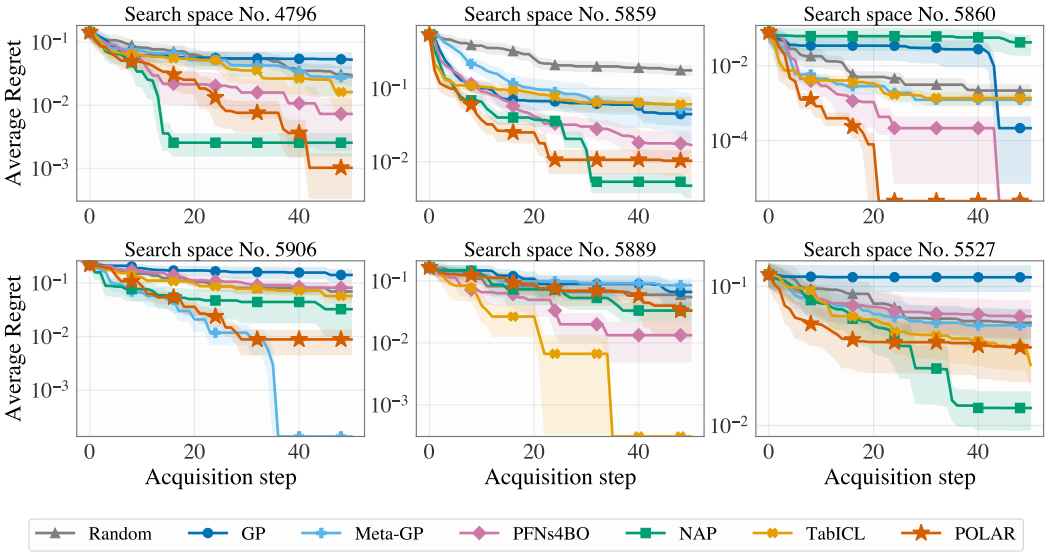



Figure A5: **HPO-B**. Per-search-space normalised regret over the acquisition trajectory. Shaded regions denote one standard error across test tasks and initialisation seeds.

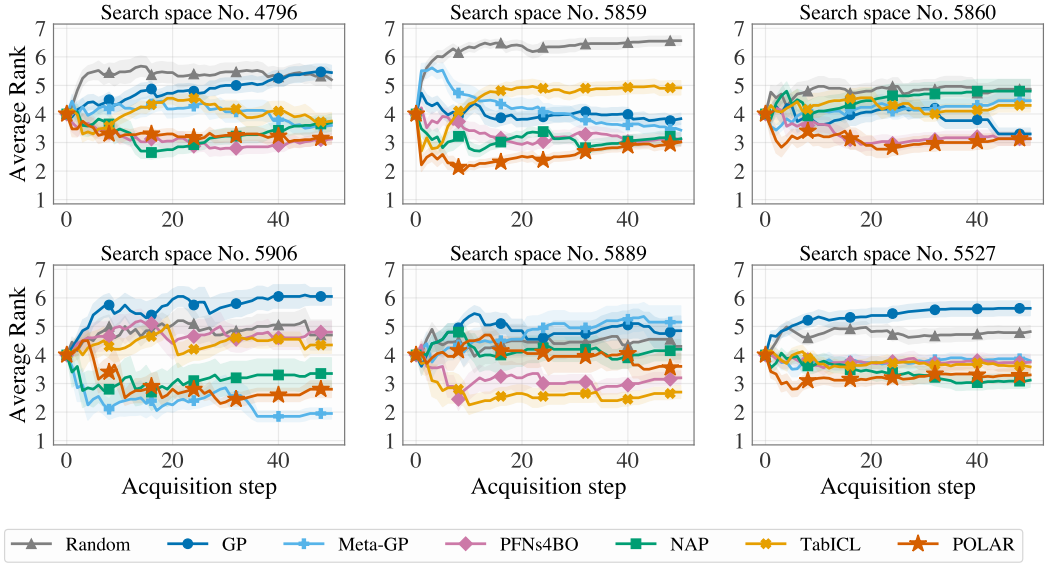



Figure A6: **HPO-B**. Per-search-space average rank over the acquisition trajectory. Shaded regions denote one standard error across test tasks and initialisation seeds.

### D.3 Hyperparameter optimisation

Figure A5 and Figure A6 break down the aggregate results of Section 6.2 by search space, reporting the normalised regret and the average rank, respectively, for each of the six HPO-B search spaces. POLAR attains the lowest regret averaged over the acquisition trajectory on four of the six search spaces (4796, 5859, 5860, and 5527) and finishes in the top two on five of them, placing third on 5889. No single baseline matches this consistency: the best final-step method varies by search space, with NAP leading on 5859 and 5527, Meta-GP on 5906, and TabICL on 5889.

Table A5: Random-backbone control for loss-driven AL. Final-step weighted prediction errors (mean $\pm$ s.e.; lower is better).

| | GP synthetic | | Benchmark functions | |
|---|---|---|---|---|
| Method | $\mathrm{RMSE}_\omega(\downarrow)$ | $\mathrm{NLPD}_\omega(\downarrow)$ | $\mathrm{RMSE}_\omega(\downarrow)$ | $\mathrm{NLPD}_\omega(\downarrow)$ |
| POLAR | $\mathbf{0.09} \pm 0.03$ | $\mathbf{-2.07} \pm 0.13$ | $\mathbf{0.05} \pm 0.01$ | $\mathbf{-1.68} \pm 0.10$ |
| Random backbone | $0.56 \pm 0.07$ | $0.16 \pm 0.21$ | $0.30 \pm 0.05$ | $0.27 \pm 0.22$ |

Table A6: Random-backbone control for the MLP BO policy. Final simple regret on standardised objectives (mean $\pm$ s.e.; lower is better).

| Method | GP synthetic | Benchmark functions |
|---|---|---|
| POLAR | $\mathbf{0.075} \pm 0.024$ | $\mathbf{0.011} \pm 0.002$ |
| Random backbone | $0.545 \pm 0.077$ | $0.248 \pm 0.032$ |

### D.4 Role of pretrained belief representations

In Section 6.1, we compare pretrained frozen, pretrained adapted (POLAR), and randomly initialised trainable backbones. These experiments establish the benefits of pretraining and task-specific adaptation, but do not directly test whether the policy head can learn an effective acquisition rule from fixed random features. To further isolate the contribution of the pretrained belief representation, we replace the policy's backbone with an architecturally identical, randomly initialised frozen backbone while keeping the policy head and training procedure unchanged. We perform this control in two settings: loss-driven active learning (Section C) with the target-aware transformer head, and synthetic meta-BO with the lightweight MLP head.

**Loss-driven active learning.** We follow the setup in Section C, using the target-aware Transformer policy head. Since the backbone also supplies the predictive distributions used to estimate rewards, we use a dual-backbone control: a pretrained backbone continues to provide predictive rewards and evaluation, while the policy head reads candidate and target representations from the frozen random encoder. This separates the quality of the policy's input representation from the validity of reward estimation. Table A5 shows substantial degradation on both in-distribution GP functions and held-out benchmark functions.

**Synthetic meta-BO.** We additionally test the standard two-hidden-layer MLP policy head on a meta-BO task. Here, meta-BO (Maraval et al., 2023) means learning a single acquisition policy across a distribution of optimisation tasks and deploying it on unseen functions without further policy training. Each training episode samples a new function using a two-dimensional variant of the synthetic-GP sampler used for loss-driven AL in Section C. Unlike loss-driven AL, here the reward is the improvement in the best observed value, equivalently the reduction in simple regret, normalised by the episode's objective range.

Both policies use the same two-hidden-layer MLP head and are trained for 5,000 updates with batches of 32 trajectories and 30 acquisitions per trajectory. We evaluate on 50 new GP functions and four standard 2D benchmarks: Branin, Himmelblau, Styblinski–Tang, and McCormick, with 20 initialisations per benchmark. Each evaluation trajectory starts from five observations and uses a candidate pool of 2,048 points. Table A6 reports final simple regret on standardised objectives after 30 acquisitions.

In both settings, the random-backbone policy performs substantially worse than POLAR, showing that neither head can learn an effective acquisition rule on its own and that the pretrained belief representation contributes materially to performance.

### D.5 Controlled distribution shift

To test the robustness of POLAR, we study a simple, controlled shift between policy training and deployment. We use ordinary, unweighted active learning on one-dimensional RBF GP functions, training separate policies under lengthscale priors $\ell \sim \mathcal{U}(0.1, 0.8)$ and $\ell \sim \mathcal{U}(0.4, 0.8)$. Each trajectory starts from two observations and selects 20 acquisitions from a pool of 200 candidates, with

Table A7: Controlled lengthscale shift in ordinary AL. Final unweighted RMSE (mean ± s.e.; lower is better).

| Training prior | $\ell = 0.1$ | $\ell = 0.2$ | $\ell = 0.3$ | $\ell = 0.4$ | $\ell = 0.6$ | $\ell = 0.8$ |
|---|---|---|---|---|---|---|
| $\mathcal{U}(0.1, 0.8)$ | $0.45 \pm 0.01$ | $0.09 \pm 0.01$ | $0.02 \pm 0.00$ | $0.01 \pm 0.00$ | $0.01 \pm 0.00$ | $0.01 \pm 0.00$ |
| $\mathcal{U}(0.4, 0.8)$ | $0.48 \pm 0.02$ | $0.10 \pm 0.01$ | $0.03 \pm 0.00$ | $0.01 \pm 0.00$ | $0.01 \pm 0.00$ | $0.01 \pm 0.00$ |

100 target points. We retain the target-aware policy head and auxiliary prediction loss. Evaluation uses unweighted RMSE.

Each trained policy is evaluated at fixed lengthscales $\ell \in \{0.1, 0.2, 0.3, 0.4, 0.6, 0.8\}$. As shown in Table A7, the restricted-prior policy performs comparably to the broad-prior policy within its training support and incurs only a small additional error outside it. Importantly, shorter-lengthscale functions are harder for both policies: the modest mismatch penalty refers to the difference between policies at the same test lengthscale, not to a small absolute increase in RMSE. This experiment provides evidence of graceful transfer under the tested parametric shift within a fixed kernel family; we do not claim robustness to arbitrary out-of-distribution tasks.

### D.6 Stochastic pendulum

We study whether one-step utility provides sufficient training credit when actions influence the informativeness of future observations.

**Task setup.** We adopt the stochastic single-pendulum setup used in Action-BED (Rossa et al., 2026), based on the conditionally linear model of Iqbal et al. (2024). The observed state consists of angular position $q_t$ and velocity $v_t = \dot{q}_t$, and the design is an applied torque $x_t \in [-1, 1]$. The discretised dynamics are

$$\begin{aligned} q_{t+1} &= q_t + \Delta t\, v_t, \\ v_{t+1} &= v_t + \Delta t\, [-\theta_1 \sin(q_t) - \theta_2 v_t + \theta_3 x_t] + 0.1\sqrt{\Delta t}\, \epsilon_t, \end{aligned} \tag{A14}$$

where $\epsilon_t \sim \mathcal{N}(0, 1)$ independently, $\Delta t = 0.05$, and the horizon is $T = 50$. Each trajectory starts at $(q_0, v_0) = (0, 0)$. The unknown restoring, damping, and torque-response coefficients have prior

$$\theta \sim \mathcal{N}\big((14.7, 0, 3)^\top,\, \mathrm{diag}(0.1, 0.01, 0.1)\big)\,.$$

At the initial state, the immediate velocity increment is insensitive to $\theta_1$ and $\theta_2$. Earlier torques must first excite the position and velocity before subsequent observations become informative about these coefficients. An action can therefore have limited immediate utility while improving future measurement opportunities.

**Implementation.** The backbone receives history rows $x_t = (q_t, v_t, x_t)$ and observations $y_t = v_{t+1} - v_t$. At each step, a two-hidden-layer MLP with width 128 and GELU activations scores candidate-conditioned TFM representations, augmented with the candidate torque and remaining-horizon fraction. Training uses 101 uniformly spaced torque candidates.

**Training and evaluation.** We compare one-step credit ($\gamma = 0$) with undiscounted reward-to-go ($\gamma = 1$) in Equation (6), keeping the simulator, architecture, training budget, and other hyperparameters fixed. Both variants use the same dense sPCE utility increments, computed with $L = 10{,}000$ contrastive samples. Each policy is trained for 5,000 updates with batches of 64 trajectories. The policy-head and backbone learning rates are $3 \times 10^{-4}$ and $10^{-5}$, respectively.

For final evaluation, each policy uses its own adapted backbone and selects torques greedily from 201 uniformly spaced candidates. We report the terminal sPCE bound on 2,000 independent trajectories with $L = 50{,}000$ contrastive samples. The random baseline samples torques uniformly from $[-1, 1]$.

**Results.** Table A8 shows that reward-to-go substantially outperforms one-step credit. The utility definition and model architecture are unchanged: the difference is whether later informational benefits are credited to earlier actions. This supports choosing the credit-assignment horizon according to the task's temporal structure. One-step credit is effective in our original benchmarks, whereas reward-to-go is preferable in this dynamical task with delayed informational consequences.

Table A8: Temporal credit assignment on stochastic-pendulum BED. Terminal sPCE (mean $\pm$ s.e.; higher is better).

| Method | sPCE |
|---|---|
| Random | $1.39 \pm 0.03$ |
| POLAR, one-step credit ($\gamma = 0$) | $2.39 \pm 0.03$ |
| POLAR, reward-to-go ($\gamma = 1$) | $\mathbf{3.96} \pm 0.04$ |

# E Computational resources and software

All experiments presented in this work, including model development, hyperparameter tuning, baseline evaluations, and preliminary analyses, were performed on GPU clusters equipped with NVIDIA H200 and AMD MI250x GPUs. The total computational resources consumed for this research, across all development stages and experimental runs, are estimated to be approximately 2000 GPU hours. The wall-clock cost of training POLAR varies with the task: on the HPO-B benchmark, a strong policy is typically obtained within 30 minutes per search space; on the location finding, policy training takes around 5 hours; on the CES, it takes about 16 hours to get the final best performance; on the DOCKSTRING benchmark, the high input dimensionality increases the per-step cost of the backbone forward pass, and a full training run takes approximately one day. The core code base is implemented in PyTorch (https://pytorch.org, License: modified BSD license), which also underlies our policy head as well as the TabPFN and TabICL backbones. We use the publicly released TabPFNv2.5 backbone from the PriorLabs repository (https://github.com/PriorLabs/TabPFN, License: Apache 2.0), with the corresponding checkpoint hosted on Hugging Face (https://huggingface.co/Prior-Labs/tabpfn_2_5). For the TabICLv2 backbone, we use the official implementation released by the authors (https://github.com/soda-inria/tabicl, License: BSD 3-Clause), with the regression checkpoint hosted on Hugging Face (https://huggingface.co/jingang/TabICL). BED baselines are adapted from the original authors' publicly available code. For the HPO-B baselines, we use the publicly released hebo-plus-model checkpoint from the official PFNs4BO repository (https://github.com/automl/PFNs4BO, License: Apache 2.0), and we use the per-step regret values released as part of the official NAP codebase (https://github.com/huawei-noah/HEBO/tree/master/NAP) for the NAP comparison. The Gaussian process baselines are implemented using GPyTorch (https://github.com/cornellius-gp/gpytorch, License: MIT). The HPO-B benchmark is obtained from its official repository (https://github.com/releaunifreiburg/HPO-B, License: MIT), and the DOCKSTRING benchmark is obtained from its official repository (https://github.com/dockstring/dockstring, License: Apache 2.0).

## 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: The abstract and introduction state our three main contributions: a unified decision-theoretic view of adaptive data acquisition through belief representations, a decoupled architecture using pretrained tabular foundation models as belief encoders with a lightweight policy head, and empirical gains across BED, BO, and AL with far fewer training samples. Each of which is supported by the corresponding sections. Claims about scope and limitations, including the reliance on a finite candidate pool and the dependence on backbone–task alignment, are discussed in Section 4.

   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: Limitations of our work are discussed explicitly in Section 4.

   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: [N/A]

   Justification: The paper does not include theoretical results.

   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: Details for all our experiments and the architecture used in our paper are available in Appendix A, as well as in the provided code.

   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 have included the code in the supplementary materials, and it will be publicly released.

   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: We describe the experimental settings for all experiments in Appendix B.

   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: All the experiment results are accompanied by error bars. We also report the method for calculating the error bars.

   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 an estimate of the resources needed for each experiment, type of compute workers, and an estimation of the total compute in Appendix E.

   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: Our work adheres to the NeurIPS Code of Ethics.

   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: [N/A]

    Justification: The work is foundational research and not tied to particular applications. We believe there is no negative societal impact of the work.

    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: The paper does not pose such risks.

    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: For each asset (code, data) used in the paper, we provide citations, links, and additional references as appropriate, in Appendix E.

    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: [N/A]

    Justification: No specific additional assets are introduced in the paper aside from research code, which will be released under a permissible MIT license.

    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: The paper does not involve crowdsourcing nor 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: The paper does not involve crowdsourcing nor research with human subjects.

    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: The core method development in this research does not involve LLMs as any important, original, or non-standard component.

    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.