Amortized Bayesian Experimental Design with In-Context Knowledge Conditioning

Zhanghu Zhao[1,2] **Daolang Huang**[1,2] **Xinyi Wen**[1,2,3] **Samuel Kaski**[1,2,4]

[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

{zhanghu.zhao, daolang.huang, xinyi.wen, samuel.kaski}@aalto.fi

Abstract

Amortized Bayesian experimental design (BED) enables real-time design strategies by shifting acquisition cost offline, but existing methods remain tied to the prior and task distribution they were trained on and cannot exploit additional information available at deployment. We introduce in-context amortized BED with unified knowledge encoding (IMBUE), a BED framework that admits external knowledge at deployment through a unified in-context interface. IMBUE accepts two forms of such knowledge without retraining, namely user-specified prior knowledge and observations from previous instances of the experiment. Both are encoded as additional input tokens and processed in-context with the accumulated observations by a shared Transformer, and a learned reliability filter scores each external token against the observations and excludes inconsistent ones. Across four benchmarks, IMBUE accelerates early-stage information acquisition when the external knowledge is reliable, and remains close to the no-external baseline when it is not.

1 Introduction

A key challenge across disciplines such as psychology [1, 2, 3], physics [4], and drug discovery [5, 6] is the design of experiments to infer the unobservable properties of the systems under study. Bayesian Experimental Design (BED) [7, 8, 9, 10] provides a principled framework for this challenge, optimizing the design by maximizing the expected information gain (EIG) about the parameters of interest. Because the traditional adaptive BED loop is computationally demanding and inherently myopic [11, 12], recent work has turned to amortized policy-based BED [13, 14, 15, 16, 17, 18], which trains a design policy π_ψ offline and at deployment maps the accumulated observations directly to the next design in a single forward pass.

A central limitation of existing amortized BED methods is that the policy π_ψ is trained to be optimal in expectation under the training prior $p(\theta)$ with no input pathway for additional information about the specific instance being run. In practice, however, experimenters rarely approach a new study without background information [19]. Two forms of such information are particularly common. The first is *user-specified prior knowledge* about the parameters of interest, drawn from domain expertise, published literature, or pilot studies, as when a clinician specifies an informed prior over pharmacokinetic parameters before a new trial [19]. The second is *auxiliary observations from previous instances* of the same experiment, as in psychophysics, where data collected from earlier participants serve as informative context for designing the experiment for a new subject. Both can sharpen the effective prior for the current instance and accelerate information acquisition, but current amortized BED frameworks have no mechanism to incorporate them at deployment.

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

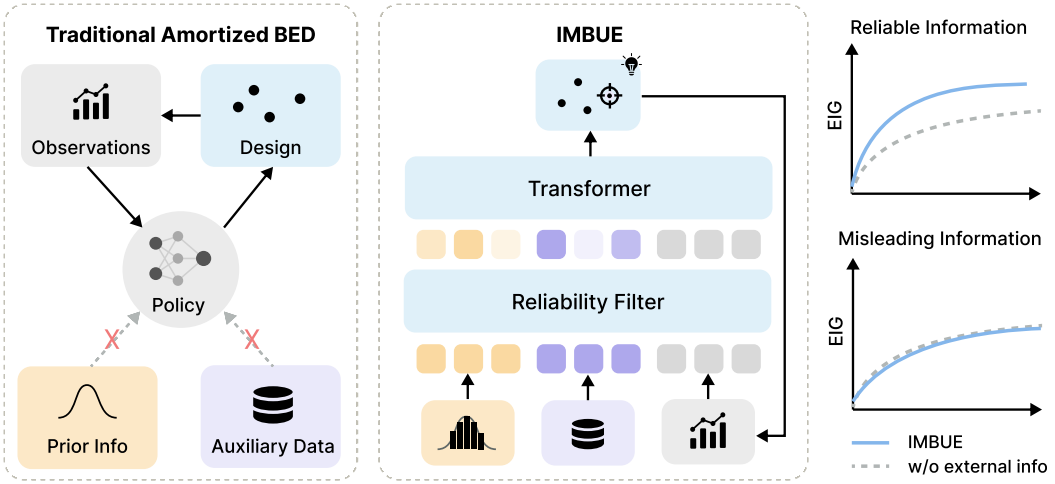


Figure 1: **Overview of IMBUE.** *Left*: Existing amortized BED policies fail to encode additional information available at deployment. *Middle*: IMBUE enables in-context adaptation at deployment by encoding external knowledge as additional input tokens, together with a reliability filter that excludes unreliable information. *Right*: IMBUE accelerates early information acquisition when external information is reliable, and remains robust to misleading information.

Yet simply admitting such inputs is not enough. External information can be noisy or only partially relevant, and may come from conditions that differ from the current instance [20, 21]. A user-specified prior may place probability mass in the wrong region of parameter space, and observations from previous participants may come from a population that does not match the current subject. Conditioning on such inputs without scrutiny can pull the policy away from what the accumulated observations actually support, leaving it worse off than a no-external baseline, a phenomenon known as negative transfer [22, 21]. The challenge, then, is to design an amortized BED framework that admits external inputs at deployment without retraining while remaining robust when those inputs are unreliable.

To address these shortcomings, we introduce **I**n-context a**M**ortized **B**ED with **U**nified knowledge **E**ncoding (IMBUE), an amortized BED framework that admits both forms of external knowledge at deployment within a shared Transformer architecture (Figure 1). External inputs, whether a user-specified prior or observations from previous instances of the experiment, are encoded as additional input tokens and processed in-context with the accumulated observations by the shared Transformer. A reliability filter scores each in-context token based on the current observations and excludes the tokens judged inconsistent from subsequent computation. During training, we expose the policy to in-context inputs from both reliable and unreliable regimes, so that a single trained policy can absorb whatever information is supplied at deployment. In the limit where all in-context tokens are excluded, the forward pass reduces to a history-only amortized BED policy.

We summarize our **contributions** below.

- We identify a structural failure mode of runtime conditioning in amortized BED. Existing methods can be extended to accept external inputs through naive concatenation, but this baseline can degrade performance even when the inputs are informative, since the policy has no learned mechanism to determine when and how to rely on them.
- We propose IMBUE, the first amortized BED framework that simultaneously supports (i) in-context knowledge available at deployment, including user-specified priors and observations from previous instances of the experiment, through a unified token interface trained under a mixture of reliable and unreliable inputs, and (ii) a learned reliability filter that assesses the relevance of each in-context token to the current experiment based on the accumulated observations, and filters out irrelevant tokens before they inform the design policy. The interface and training mixture provide the gains and most of the robustness; the filter further reduces negative transfer under conflicting inputs. Both components are realized within a single Transformer trained end-to-end.

- We evaluate IMBUE on four standard BED benchmarks. When external knowledge is reliable, IMBUE accelerates early-stage information acquisition. When it is not, IMBUE reduces the degradation that naive concatenation incurs and closely matches the no-external baseline. A staged ablation separates these contributions, and IMBUE does not degrade sharply under three deployment shifts absent from training.

2 Related work

Bayesian Experimental Design (BED). Sequential BED aims to maximize the EIG over a sequence of experiments. Traditional adaptive BED requires nested Monte Carlo or variational EIG estimation [11, 12, 23] together with Bayesian posterior inference at every step [9, 8, 7], which is computationally demanding. Amortized policy-based BED, introduced by Deep Adaptive Design [13], avoids this cost by training a design policy π_ψ offline and producing each subsequent design through a single forward pass at deployment. The framework has since been extended to implicit-likelihood models [14] and trained with reinforcement learning to remove the need for differentiable simulators [15, 24]. More recent work couples the policy with a jointly trained inference network in fully end-to-end architectures, including ALINE [17] and JADAI [18]. A parallel direction, Step-DAD [16], relaxes full amortization through policy refinement at deployment that requires gradient updates and simulator access during the live experiment, which limits its applicability when designs must be produced in real time. Across all of these methods, the prior under which π_ψ is trained is treated as a fixed property of the model, and there is no input pathway through which an experimenter can supply in-context knowledge about the current experiment at deployment. Our work addresses this gap by introducing a single design policy that accepts in-context knowledge at deployment.

In-context knowledge in Bayesian methods. A separate line of work studies how in-context knowledge can shape the behavior of a Bayesian model. In Bayesian optimization (BO), this takes the form of user-specified priors over the optimum or data from related tasks [25, 26, 27, 28, 29, 30, 31, 32, 33]. Closer to our setting, recent work supplies in-context knowledge to amortized meta-learners at deployment. Informed meta-learning [34] conditions a neural process on natural-language task descriptions. The Amortized Conditioning Engine (ACE) [35] represents user-specified priors as discretized histograms and injects them as additional input tokens, with related developments using Gaussian-mixture parameterizations [36] and broader distribution classes [37]. A complementary line replaces this token-based interface with test-time guidance [38]. In-context transfer methods [39, 40, 21] encode auxiliary datasets as additional input tokens. These approaches show that runtime conditioning is feasible within a single forward pass, but they target passive inference rather than sequential design. Our work extends this conditioning capability to sequential design and additionally addresses the reliability of the supplied in-context knowledge through a learned reliability filter.

3 Method

This section is organized as follows. Section 3.1 formalizes the setup and objective, Section 3.2 the token interface, Section 3.3 the reliability filter and its supervision, and Section 3.4 the backbone and training; notation is summarized in Appendix A.

3.1 Problem setup

Let $\theta \in \Theta \subseteq \mathbb{R}^L$ denote the latent parameters of the system under study. Given a design $x \in \mathcal{X}$, the experiment returns an outcome $y \sim p(y \mid \theta, x)$. We consider sequential experimentation over a horizon of T rounds. Starting from an empty experimental history $\mathcal{D}_0 = \emptyset$, at round t a design policy π_ψ proposes $x_t \sim \pi_\psi(\cdot \mid \mathcal{D}_{t-1})$, an outcome $y_t \sim p(y \mid \theta, x_t)$ is observed, and the history is updated to $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(x_t, y_t)\}$. The inference goal is captured by a target variable Z. Standard parameter-oriented BED [7] corresponds to $Z = \theta$, and prediction-oriented BED sets $Z = y^\star$ at test inputs $x^\star \sim p^\star(x^\star)$ with $y^\star \sim p(y^\star \mid \theta, x^\star)$ [41]. We call one such T-round experiment a *rollout*.

In-context knowledge. We extend the standard setup with in-context knowledge available at deployment, $C = (C_{\text{prior}}, C_{\text{obs}})$, where C_{prior} encodes user-specified marginal beliefs over the parameters $\theta = (\theta_1, \ldots, \theta_L)$ and $C_{\text{obs}} = \{\mathcal{A}_k\}_{k=1}^K$ contains K sets of observations from previous instances of the experiment, with $\mathcal{A}_k = \{(x_i^{(k)}, y_i^{(k)})\}_{i=1}^T$. We call these the *prior* and *previous-*

instance channels; the policy now proposes $x_t \sim \pi_\psi(\cdot \mid \mathcal{D}_{t-1}, C)$. Either component may be empty, recovering the standard BED setting in which the policy conditions only on the experimental history.

Information objective. We measure the utility of the policy under C by the conditional information gain

$$\mathcal{U}(\psi; C) = I_{p_\psi}(Z; \mathcal{D}_T \mid C) = H(Z \mid C) - H_{p_\psi}(Z \mid \mathcal{D}_T, C), \tag{1}$$

where H denotes entropy. Here $p_\psi(Z, \mathcal{D}_T \mid C)$ is the joint distribution of target and trajectory induced by running π_ψ for T rounds under C (Equation (13) in Appendix B). During training, C is sampled from the training context distribution $p_{\text{train}}(C)$, and the optimized objective is $\mathbb{E}_{C \sim p_{\text{train}}} \mathcal{U}(\psi; C)$.

Limitations of existing amortized BED. Existing amortized BED methods train a policy π_ψ to maximize $\mathcal{U}(\psi; C)$ under an empty channel $C = \emptyset$, with conditioning fixed at training time. Two issues arise when in-context knowledge is available at deployment. First, π_ψ has no input pathway through which a user-specified prior or observations from a previous instance can enter, so absorbing such information requires retraining. Even with an interface in place, priors and previous observations differ in form, length, and semantics, so a naive concatenation does not yield a unified handler for both. Second, external knowledge cannot be assumed reliable. A user-specified prior may be miscalibrated and a previous instance may belong to a different population, and conditioning on it without scrutiny can degrade performance below the no-external baseline (negative transfer [22, 21]). IMBUE addresses these two issues through a unified token interface (Section 3.2) and a reliability filter (Section 3.3), built on a shared Transformer backbone (Section 3.4).

3.2 In-context token interface

IMBUE represents each piece of in-context knowledge as a token, so that a single Transformer accepts either C_{prior} or C_{obs} at deployment without retraining. We treat the two channels separately and describe the corresponding token types in turn.

Prior tokens. A user supplies marginal beliefs $\rho_l(\theta_l)$, $l \in \{1, \dots, L\}$, over individual parameters. We discretize each ρ_l into a normalized histogram over N_B uniform bins covering the support of θ_l,

$$\rho_l^{(b)} = \frac{\int_{B_b} \rho_l(u)\, du}{\sum_{b'=1}^{N_B} \int_{B_{b'}} \rho_l(u)\, du}, \qquad \boldsymbol{\rho}_l = \left(\rho_l^{(1)}, \dots, \rho_l^{(N_B)}\right) \in \mathbb{R}^{N_B}, \tag{2}$$

where B_b is the b-th bin. This non-parametric representation accepts arbitrary marginal beliefs, including multimodal, skewed, or truncated ones, without committing to a specific parametric family. The histogram is mapped to the model dimension D by an MLP $f_{\text{prior}} \colon \mathbb{R}^{N_B} \to \mathbb{R}^D$, and a learnable parameter-specific embedding $\boldsymbol{\zeta}_l \in \mathbb{R}^D$ ties the resulting token to its parameter,

$$\mathbf{e}_l^{\text{prior}} = f_{\text{prior}}(\boldsymbol{\rho}_l) + \boldsymbol{\zeta}_l \in \mathbb{R}^D. \tag{3}$$

The same $\boldsymbol{\zeta}_l$ is shared with the target token for θ_l in the backbone (Section 3.4), aligning the prior with its target inside the encoder. Parameters without a user-specified prior are represented by uniform histograms.

Instance tokens. Each instance $\mathcal{A}_k = \{(x_i^{(k)}, y_i^{(k)})\}_{i=1}^T$ contains T design–outcome pairs. Feeding all $K \cdot T$ raw observations into the main encoder would scale self-attention quadratically in $K \cdot T$ and couple the encoder input length to the previous-instance horizon. We instead compress each $\mathcal{A}_k$ into a single instance token $\mathbf{r}_k \in \mathbb{R}^D$ through a per-instance encoder g_{aux}, a separate Transformer that operates within each instance independently and is shared across all K instances,

$$\mathbf{r}_k = g_{\text{aux}}\left(\mathcal{A}_k\right) \in \mathbb{R}^D. \tag{4}$$

The compression follows summary-token schemes used for passive amortized inference [40, 21]. Architectural details, including the embedding of design–outcome pairs and the summary-slot pooling in g_{aux}, are provided in Appendix C.4. We refer to $\mathbf{e}_l^{\text{prior}}$ and $\mathbf{r}_k$ collectively as *in-context tokens*.

3.3 Reliability filter

External knowledge is not always reliable. A user-specified prior may concentrate mass in the wrong region of parameter space, and an instance may be drawn from a population that differs from the

current one. Conditioning on such tokens without scrutiny can pull the policy away from what the accumulated observations actually support, producing negative transfer. We attach a reliability filter that decides, conditional on the observations gathered so far, whether each in-context token participates in subsequent computation.

Mechanism. The reliability filter f_ω produces, for each in-context token $\mathbf{e}_j$, a scalar reliability score

$$\alpha_j^{(t)} = \sigma\big(f_\omega(\mathbf{e}_j;\, \mathbf{E}_t)\big) \in [0,1], \tag{5}$$

where σ is the logistic sigmoid, $\mathbf{E}_t$ denotes the embeddings of the experimental history $\mathcal{D}_t$, and f_ω is a cross-attention block [42] that uses $\mathbf{e}_j$ as the query and $\mathbf{E}_t$ as keys and values. We threshold $\alpha_j^{(t)}$ at $\tau = 0.5$ to obtain a binary keep-or-discard decision, and tokens with $\alpha_j^{(t)} \le \tau$ are excluded from self-attention in the backbone via a key-padding mask. When the history is empty, there is nothing to score, and all in-context tokens are excluded until the first observation arrives. We condition f_ω on $\mathbf{E}_t$ because reliability depends on consistency with gathered observations rather than on the in-context token alone. The same token may receive different filter outputs at different rounds as $\mathcal{D}_t$ grows.

Supervision. Because training rollouts are simulated (Section 3.1), the ground-truth parameter $\theta^\star$ of the current instance is available during training and gives an oracle definition of reliability. The labels use only latent variables that standard amortized BED training already draws [13, 17], and are neither observed nor required at deployment. For a prior token associated with parameter θ_l, reliability reflects whether the histogram concentrates mass near $\theta_l^\star$. We use the log-ratio of $\rho_l^{(b^\star)}$ against a uniform reference at the bin $b^\star$ that contains $\theta_l^\star$,

$$r_j^{\text{prior}} = \sigma\left(\beta \log \frac{\rho_l^{(b^\star)}}{1/N_B}\right). \tag{6}$$

For an instance token, reliability reflects whether the previous instance shares parameters with the current one. We use a Gaussian kernel in normalized parameter space,

$$r_j^{\text{inst}} = \exp\left(-\kappa \sum_{l=1}^{L} \left(\frac{\theta_l^{(k)} - \theta_l^\star}{r_l^{\max} - r_l^{\min}}\right)^2\right), \tag{7}$$

where $\theta^{(k)}$ denotes the parameters that generated instance k, and $[r_l^{\min}, r_l^{\max}]$ is the support of θ_l under the prior, or a fixed finite range when the support is unbounded. The task-specific ranges are reported in Appendix C.4. The parameters $\beta > 0$ and $\kappa > 0$ control label sharpness. Both labels are thresholded at $1/2$ to yield binary classification targets $c_j = \mathbb{I}[r_j > 1/2]$. For prediction-oriented BED with $Z = y^\star$, the parameter-distance label of Equation (7) does not directly apply, and we use a binary variant tied to the generating regime, deferred to Appendix D.3.

Mixed-regime training. To expose the filter to both regimes, instances are drawn from a $70/30$ relevant/irrelevant mixture during training. In the *relevant* regime, $\theta^{(k)}$ is a small Gaussian perturbation of $\theta^\star$ scaled per dimension by the prior support. In the *irrelevant* regime, $\theta^{(k)}$ is drawn independently of $\theta^\star$. Priors are sampled with an analogous mixture of calibrated, miscalibrated, and uniform regimes, with full procedures in Appendix C.3 and Appendix C.4.

3.4 Backbone and training

We adopt the shared Transformer backbone, the head structure, the per-step reward, and the variational training scheme of ALINE [17] unchanged. Our contributions are the in-context token interface (Section 3.2), trained under a mixture of reliable and unreliable inputs, and the reliability filter (Section 3.3). Figure 2 provides an overview of the network design; Appendix C.2 provides full details.

Variational lower bound. The posterior $p_\psi(Z \mid \mathcal{D}_T, C)$ entering Equation (1) is intractable for general likelihoods, so we train against a tractable variational lower bound on $\mathcal{U}(\psi; C)$. Following Barber and Agakov [43], for any variational distribution $q_\phi(Z \mid \mathcal{D}_T, C)$,

$$\mathcal{U}(\psi; C) \;\ge\; H(Z \mid C) + \mathbb{E}_{p_\psi(Z, \mathcal{D}_T \mid C)}\big[\log q_\phi(Z \mid \mathcal{D}_T, C)\big], \tag{8}$$

which is jointly maximized in ψ and ϕ (Appendix B). Since $H(Z \mid C)$ depends on neither ψ nor ϕ, this amounts to maximizing $\mathcal{J}(\psi, \phi; C) = \mathbb{E}_{p_\psi}[\log q_\phi(Z \mid \mathcal{D}_T, C)]$. We instantiate q_ϕ as an inference head on the shared Transformer encoder, trained jointly with the policy.

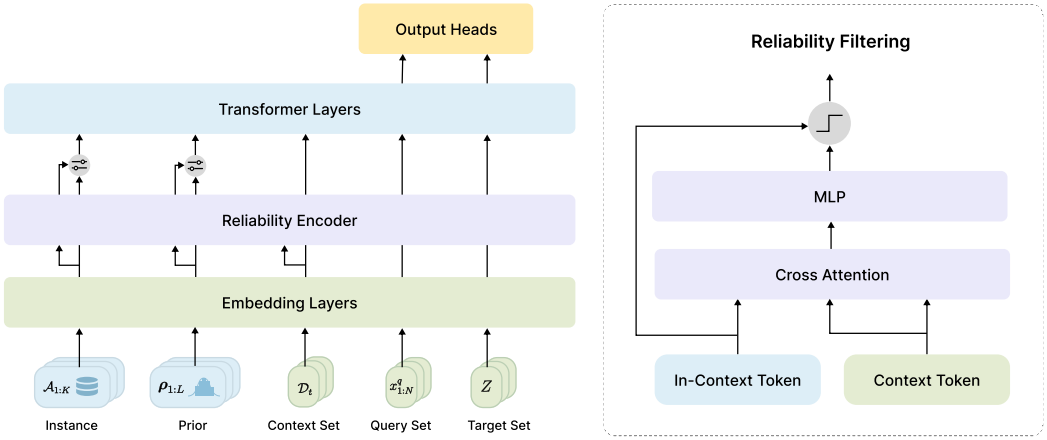


Figure 2: **Architecture of IMBUE.** Auxiliary previous-instance data and the user-prior histogram are encoded as in-context tokens and screened by the reliability filter (right). Retained tokens are then combined with context, query, and target tokens in the shared Transformer layers.

Tokens and heads. In-context tokens that survive the filter join three further token groups inside the shared encoder. *Context tokens* encode the experimental history $\mathcal{D}_t$, *query tokens* encode the candidate designs at round t, and *target tokens* are learnable slots, one per parameter when $Z = \theta$ and one per test input when $Z = y^\star$. Self-attention propagates information among the four groups under a structured attention mask, with the design head reading the query positions to produce π_ψ and the inference head reading the target positions to produce q_ϕ as a Gaussian mixture over the components of Z [35].

Training. Each training iteration draws $\theta^\star$ and C (Algorithm 2), computes the labels c_j of Section 3.3, and simulates a T-round rollout in which $x_t \sim \pi_\psi(\cdot \mid \mathcal{D}_{t-1}, C)$ and $y_t \sim p(y \mid \theta^\star, x_t)$ are drawn. After round t, the inference head evaluates $\log q_\phi^{(t)} = \log q_\phi(Z^\star \mid \mathcal{D}_t, C)$, with $Z^\star$ the realized target ($\theta^\star$, or $y^\star$ simulated from it), and the filter re-scores the in-context tokens, giving $\alpha_j^{(t)}$ in Equation (5). Averaged over a minibatch of rollouts, these quantities enter three losses, combined as

$$\mathcal{L}(\psi, \phi, \omega) = \lambda_1 \mathcal{L}_{\text{policy}} + \lambda_2 \mathcal{L}_{\text{inf}} + \lambda_3 \mathcal{L}_{\text{filter}}, \tag{9}$$

where

$$\mathcal{L}_{\text{inf}} = -\tfrac{1}{T+1} \textstyle\sum_{t=0}^{T} \log q_\phi^{(t)}, \tag{10}$$

$$\mathcal{L}_{\text{policy}} = -\tfrac{1}{T} \textstyle\sum_{t=1}^{T} \gamma^t \, R_t \log \pi_\psi(x_t \mid \mathcal{D}_{t-1}, C), \quad \text{with } R_t = \log q_\phi^{(t)} - \log q_\phi^{(t-1)}, \tag{11}$$

$$\mathcal{L}_{\text{filter}} = \tfrac{1}{T\,|C|} \textstyle\sum_{t=1}^{T} \sum_{j=1}^{|C|} \mathrm{BCE}(\alpha_j^{(t)}, c_j). \tag{12}$$

- *Inference loss.* $\mathcal{L}_{\text{inf}}$ trains the inference network q_ϕ to infer $Z^\star$ from the experimental history and retained in-context tokens at each round. Its final-step term optimizes Equation (8) with respect to ϕ and, for a fixed policy, tightens the bound. The intermediate-step terms provide accurate inference estimates for the policy rewards.
- *Policy loss.* $\mathcal{L}_{\text{policy}}$ trains the policy network π_ψ, which determines $\mathcal{D}_T$ through its selected designs. It uses the changes R_t in $\log q_\phi^{(t)}$ as rewards, discounted by $\gamma \in (0, 1]$; gradients are stopped at q_ϕ to ensure the policy cannot raise its reward by altering the inference head. These rewards telescope to the terminal log-density minus a policy-independent initial term, thereby favoring designs that increase the variational term in Equation (8).
- *Filter loss.* $\mathcal{L}_{\text{filter}}$ is the average binary cross-entropy (BCE) of the scores $\alpha_j^{(t)}$ against the labels c_j from Equations (6) and (7), where $|C|$ denotes the number of in-context tokens, and is set to 0 when $C = \emptyset$; it is not derived from the bound and is detached from the shared embedders.

Because R_t is informative only once q_ϕ is accurate, the first E_{warm} epochs draw designs uniformly and optimize only $\mathcal{L}_{\text{inf}}$ and $\mathcal{L}_{\text{filter}}$; afterwards π_ψ selects the designs and all three losses are optimized jointly. Algorithm 1 gives the full procedure and Appendix C.2 the hyperparameters, with $\lambda_1 = \lambda_2 = \lambda_3 = 1$ and $\gamma = 1$ throughout.

4 Experiments

We evaluate IMBUE on four BED benchmarks. The Constant Elasticity of Substitution (CES) and Location Finding tasks support our two main claims about reliable and unreliable in-context knowledge across both channels (Sections 4.1 to 4.3). Section 4.4 separates the contributions of training exposure, the token interface, and the filter. The active learning task tests whether IMBUE filters instances individually rather than as a block (Section 4.5), and the psychometric task evaluates the previous-instance channel on a realistic scientific application (Section 4.6). Full simulator specifications, baselines, and hyperparameters are in Appendix D. Baselines without a native pathway for in-context knowledge receive it through naive concatenation to the experimental history (Appendix D.2), denoted by *. Code is available at https://github.com/DralnMatthew/IMBUE.

4.1 Naive concatenation fails

The CES task [44] infers a preference parameter $\theta \in \mathbb{R}^5$ from a respondent's comparisons of two baskets of goods over $T = 10$ rounds. The target is $Z = \theta$ and the metric is the sequential Prior Contrastive Estimation (sPCE) lower bound on the total EIG [13]. We use CES in Sections 4.1 and 4.3 for the previous-instance channel, with C_{obs} supplying K instances from past respondents drawn from the relevant or irrelevant regime of Section 3.3.

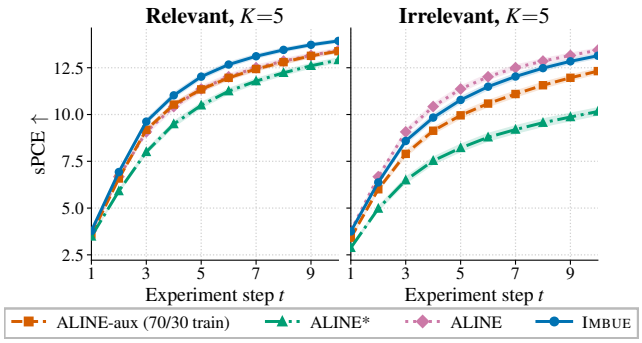


Figure 3: CES with $K = 5$ previous instances drawn from the relevant or irrelevant regime. sPCE (mean $\pm$ 95% confidence interval (CI)) over $T = 10$ steps across 2,000 rollouts.

We first ask whether an existing amortized BED policy can make use of in-context knowledge supplied at deployment or at training time. We compare three ALINE-based variants on CES with $K = 5$ instances. ALINE* concatenates instances only at deployment, ALINE-aux is retrained on the same $70/30$ relevant/irrelevant training mixture used by IMBUE, and ALINE is the original policy. Figure 3 shows that ALINE* falls below ALINE in both regimes, indicating that previous instances unseen during training disrupt the policy regardless of their reliability. ALINE-aux roughly matches ALINE under relevant input and trails it under irrelevant input, so retraining alone extracts little from reliable instances and only partly suppresses misleading information. IMBUE exceeds ALINE under relevant input and stays close to it under irrelevant input. The gap between ALINE-aux and IMBUE is decomposed in Section 4.4, and the full pattern across $K \in \{0, 1, 5\}$ is reported in Appendix E.2.

4.2 Reliable in-context knowledge accelerates acquisition

We then ask whether IMBUE reaches a given level of information about θ earlier when reliable in-context knowledge is supplied, focusing on the prior channel. The Location Finding benchmark [45] localizes two hidden sources in $[0, 1]^2$ from $T = 20$ noisy measurements, with $Z = \theta$ and the sPCE metric. We supply C_{prior} as one histogram per coordinate, and contrast a strong prior concentrated near $\theta^\star$ against a miscalibrated prior shifted away from it (Appendix C.3). Baselines are DAD [13], RL-BOED [15], ALINE [17], and a random reference.

Figure 4 shows that under a strong prior, IMBUE is consistently higher than ALINE through the early and middle rollout, and attains ALINE's final sPCE several steps earlier in the rollout. The advantage narrows toward the end as the contribution of the prior diminishes relative to the accumulated observations. Under a miscalibrated prior, IMBUE

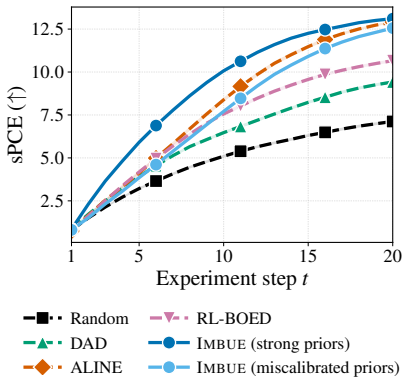


Figure 4: Location Finding. sPCE (↑) over $T = 20$ steps, mean $\pm$ 95% CI across 2,000 rollouts.

stays close to the ALINE baseline through the rollout and finishes well above DAD, RL-BOED, and the random reference. Appendix E.1 reports how the per-token reliability outputs evolve as observations accumulate, along with additional results under weak and uniform priors.

The same acceleration pattern holds on the previous-instance channel. On CES, supplying IMBUE with K relevant previous instances raises its sPCE above the $K = 0$ baseline, and the gain levels off as K increases (see Table 1 and the full K-sweep in Appendix E.2).

4.3 IMBUE remains robust to unreliable in-context knowledge

We test whether IMBUE reduces negative transfer on CES (Section 4.1), supplying $K \in \{0, 1, 5\}$ relevant or irrelevant previous instances (Section 3.3). Baselines are DAD*, RL-BOED*, ALINE*, and a random reference.

Table 1: CES. sPCE (mean $\pm$ 95% CI, 2,000 rollouts) under K previous instances. At $K = 0$ nothing is concatenated, so the $K = 0$ entries of the starred methods are the standard DAD, RL-BOED, and ALINE without any in-context knowledge.

	$K = 0$	$K = 1$		$K = 5$	
Method	—	Relevant	Irrelevant	Relevant	Irrelevant
Random	9.05 ± 0.26	—	—	—	—
DAD*	11.52 ± 0.17	7.38 ± 0.21	7.37 ± 0.21	5.16 ± 0.21	5.40 ± 0.21
RL-BOED*	$\mathbf{14.34 \pm 0.13}$	4.54 ± 0.20	4.74 ± 0.21	2.56 ± 0.15	2.84 ± 0.17
ALINE*	13.37 ± 0.15	13.27 ± 0.15	11.10 ± 0.22	13.07 ± 0.15	10.16 ± 0.24
IMBUE	13.46 ± 0.15	$\mathbf{13.63 \pm 0.14}$	$\mathbf{13.42 \pm 0.15}$	$\mathbf{13.93 \pm 0.12}$	$\mathbf{13.14 \pm 0.16}$

Table 1 shows that IMBUE's sPCE increases with K when the instances are relevant, consistent with the gains from reliable prior knowledge in Section 4.2. When the instances are irrelevant, IMBUE stays close to its $K = 0$ score at both values of K, while ALINE* degrades as K grows. DAD* and RL-BOED* drop under any added input regardless of its reliability. The same robustness appears on the prior channel under a miscalibrated prior (Figure 4).

We further evaluate the trained models under three deployment shifts absent from training: biased but partially relevant previous instances, a different noise model in the previous instances, and priors whose support excludes $\theta^\star$ (Appendix E.6). Across these settings, IMBUE does not degrade sharply, consistent with the robustness observed above.

4.4 Contributions of training, tokenization, and filtering

We use a staged ablation on CES with $K = 5$ previous instances (Table 2): starting from ALINE*, we successively add mixed-regime training (ALINE-aux), the instance-token interface (IMBUE w/o filter), and reliability filtering (IMBUE). IMBUE w/o filter is retrained without the filter and keeps all instance tokens. Retrieve-then-condition uses ALINE-aux with only the two instances that best explain the observations collected so far (Appendix D.2).

Table 2 shows that mixed-regime training and the instance-token interface improve mean sPCE over ALINE* under both reliable and unreliable inputs. Adding the filter further increases mean sPCE under biased and irrelevant inputs, while performance under relevant inputs remains similar. This is consistent with the filter's purpose: limiting negative transfer from unreliable instances and simply retaining reliable ones, for which no additional gain is expected. Under irrelevant inputs, the benefit of filtering is largest in the middle of CES rollouts, grows with K, and is more persistent on the psychometric task (Appendix E.5). Compared with ALINE-aux, Retrieve-then-condition improves sPCE under irrelevant inputs but reduces it under relevant inputs, since its fixed top-two selection must discard useful instances. IMBUE has no such budget and can keep all five instances or drop all five at any round, depending on the observations accumulated so far. The parameter and latency overhead of the token interface and reliability filter is reported in Appendix E.8.

On the prior channel in Location Finding, however, the filter has little effect on sPCE across all four tested prior conditions (Appendix E.5), and IMBUE w/o filter retains the robustness to miscalibrated

Table 2: Staged ablation on CES with $K = 5$ previous instances. sPCE (mean $\pm$ 95% CI, 2,000 rollouts). Biased instances are a deployment shift absent from training (Appendix E.6).

	$K = 0$	$K = 5$		
Method	—	Relevant	Biased	Irrelevant
ALINE*	13.37 ± 0.15	13.07 ± 0.15	11.26 ± 0.21	10.16 ± 0.24
ALINE-aux	13.19 ± 0.15	13.52 ± 0.13	12.72 ± 0.16	12.23 ± 0.18
IMBUE w/o filter	13.39 ± 0.15	$\mathbf{13.99 \pm 0.12}$	13.10 ± 0.15	12.97 ± 0.16
IMBUE	$\mathbf{13.46 \pm 0.15}$	13.93 ± 0.12	$\mathbf{13.23 \pm 0.15}$	$\mathbf{13.14 \pm 0.16}$
Retrieve-then-condition	—	13.29 ± 0.14	—	12.69 ± 0.17

priors observed in Figure 4. Together with the CES and psychometric filter ablations, this indicates that filtering provides additional protection primarily against unreliable previous-instance input.

4.5 IMBUE handles mixed-relevance inputs per instance

In Sections 4.1 and 4.3, C_{obs} is supplied as either completely relevant or irrelevant. We test whether IMBUE can retain relevant instance tokens and exclude irrelevant ones within a single forward pass. The benchmark is 1D active learning. At each step the policy chooses a design $x_t \in [-5, 5]$ and observes a noisy evaluation of a function $f^\star$. The target is $Z = y^\star$ at held-out test inputs, and the metric is the root mean squared error (RMSE) of the predictions on that test set. *Synthetic* is a fresh draw from the same GP prior used during training, and probes the in-distribution case. *Forrester*, *Higdon*, and *Gramacy & Lee* are deterministic analytic functions, and probe out-of-distribution generalization. Further benchmark details are in Appendix D.3. We fix $K = 5$ previous instances and sweep the number of irrelevant ones from 0 to 5. Relevant instances are per-instance perturbations of $f^\star$ and irrelevant instances are unrelated 1D functions. Baselines are GP* (a Gaussian process that pools the previous-instance observations with the experimental ones as if all were evaluations of $f^\star$), ALINE*, and a multi-task GP (MT-GP) that treats each previous instance as a separate task and learns its correlation with the function $f^\star$ (Appendix D.3).

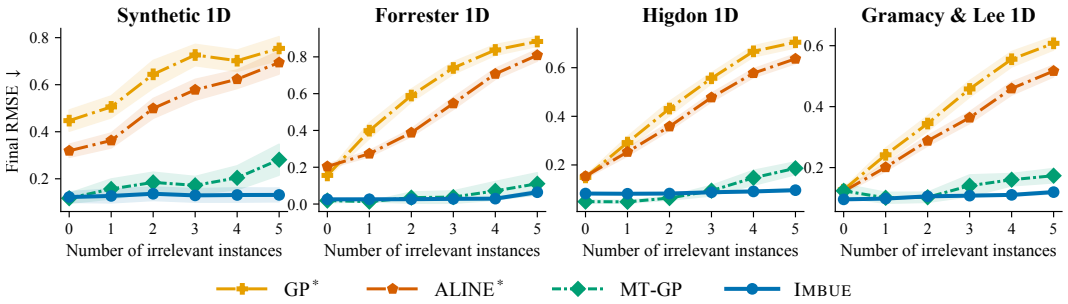


Figure 5: Mixed previous-instance input on four 1D GP active learning benchmarks. Final RMSE (↓, mean $\pm$ 95% CI over 100 rollouts). With $K = 5$ instances, the x-axis varies the number that are irrelevant from 0 to 5.

Figure 5 shows that GP*, ALINE*, and MT-GP degrade as the number of irrelevant instances grows, whereas IMBUE remains essentially flat on all four benchmarks. The contrast with ALINE*, which shares the backbone with IMBUE but concatenates the instances at deployment, shows that the backbone architecture alone does not provide this robustness.

MT-GP and IMBUE handle irrelevant input differently. MT-GP fuses all K instances into a joint model with learned task-correlation weights. IMBUE instead applies a binary keep-or-discard decision per token, excluding tokens judged unreliable from self-attention. RMSE curves over experimental steps, an extended baseline comparison, and an inference-time comparison are in Appendix E.3.

4.6 Realistic psychometric task

Our final experiment infers the parameters of the psychometric model [3] from binary yes/no responses to stimulus intensities $x \in [-5, 5]$ over $T = 30$ steps. The target is $Z = \theta = (\theta_{\text{threshold}}, \theta_{\text{slope}}, \theta_{\text{guess}}, \theta_{\text{lapse}}) \in \mathbb{R}^4$. The threshold $\theta_{\text{threshold}}$ and slope θ_{slope} are the parameters of scientific interest, while the guess rate and lapse rate are nuisance parameters constrained to small physiological supports [46]. The metric is the joint RMSE on the threshold–slope marginals of the inferred posterior. We supply C_{obs} as $K = 5$ instances from previous participants drawn from the relevant or irrelevant regime of Section 3.3. As baselines we include the amortized policy ALINE [17] and two classical psychometric methods, QUEST+ [3] and Psi-marginal [46]. All serve as no-external references.

Figure 6 shows that the patterns observed on CES in Section 4.3 carry over to the psychometric task. IMBUE exploits the previous-instance channel that the classical methods do not natively accept. With relevant previous instances, it attains lower joint RMSE than the no-external baselines from the second step onward; its RMSE drops sharply in the first few steps, and the gap narrows toward the end. With irrelevant previous instances, IMBUE tracks the no-external baselines closely. Its joint RMSE is also lower than that of IMBUE w/o filter, with an advantage that grows and persists longer as K increases (Appendix E.5). Appendix E.4 extends this comparison to other values of K. IMBUE is comparable to ALINE, QUEST+, and Psi-marginal without previous-instance input ($K = 0$), and its gains with relevant instances are larger at higher K. ALINE*, by contrast, does not improve over its $K = 0$ RMSE even with relevant instances, and its RMSE grows with K under irrelevant ones.

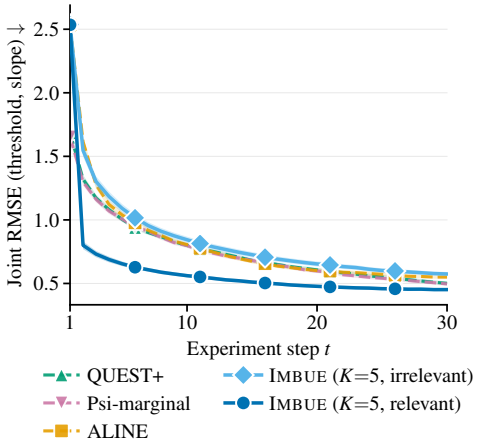


Figure 6: Psychometric task. Joint RMSE on $(\theta_{\text{threshold}}, \theta_{\text{slope}})$ (↓) over $T = 30$ steps, mean ± 95% CI across 500 rollouts.

5 Conclusion

We introduced IMBUE, an amortized BED framework that lets a trained design policy incorporate in-context knowledge supplied at deployment without retraining, either as a user-specified prior or as observations from previous instances of the experiment. A unified token interface encodes priors and observations from previous instances as additional input tokens, and a reliability filter scores each in-context token against the accumulated observations to provide an interpretable per-token decision. The staged ablation attributes the gains and most of the robustness to this interface and its mixed-regime training, with the filter adding protection when tokens conflict with the observations, mainly on the previous-instance channel. Across four standard BED benchmarks, IMBUE accelerates early-stage information acquisition with reliable external inputs, remains close to the no-external baseline with unreliable ones, and does not degrade sharply under three deployment shifts absent from training.

Limitations and future work. The in-context knowledge we consider is structured, with priors represented as histograms and previous instances as sequences of design–outcome pairs. Priors enter as parameter-wise marginals, so parameter dependencies are not represented, and the likelihood, design, and observation spaces are assumed to stay within the trained task family (Appendix E.6 probes three specific deployment shifts). Admitting richer forms, such as natural-language task descriptions, would bring amortized BED closer to the settings in which domain experts naturally encode their knowledge. The main qualitative conclusions hold across the tested candidate-pool sizes and constructions (Appendix E.7), but candidate enumeration does not scale to very high-dimensional designs; extending IMBUE to continuous design spaces through actor-critic formulations [47] would broaden its applicability. Finally, the filter is less reliable in the first few rounds, when the limited experimental history makes it difficult to distinguish informative from misleading external inputs.

Acknowledgments and Disclosure of Funding

This work was supported by EU funding ERC ODD-ML 101201120 and the Research Council of Finland (Flagship programme: Finnish Center for Artificial Intelligence FCAI, decision 359207). The authors acknowledge the research environment provided by ELLIS Institute Finland. SK was also supported by the UKRI Turing AI World-Leading Researcher Fellowship, [EP/W002973/1]. The authors wish to thank the Aalto Science-IT project and CSC–IT Center for Science, Finland, for the computational and data storage resources provided. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.



References

[1] Jay I Myung, Daniel R Cavagnaro, and Mark A Pitt. A tutorial on adaptive design optimization. *Journal of Mathematical Psychology*, 57(3-4):53–67, 2013. 1

[2] Benjamin T Vincent and Tom Rainforth. The DARC toolbox: automated, flexible, and efficient delayed and risky choice experiments using Bayesian adaptive design. *PsyArXiv preprint*, 2017. 1

[3] Andrew B Watson. QUEST+: A general multidimensional Bayesian adaptive psychometric method. *Journal of Vision*, 17(3):10–10, 2017. 1, 10, 27, 28

[4] Sergey Dushenko, Kapildeb Ambal, and Robert D McMichael. Sequential Bayesian experiment design for optically detected magnetic resonance of nitrogen-vacancy centers. *Physical Review Applied*, 14(5):054036, 2020. 1

[5] Jiankun Lyu, Sheng Wang, Trent E Balius, Isha Singh, Anat Levit, Yurii S Moroz, Matthew J O'Meara, Tao Che, Enkhjargal Algaa, Kateryna Tolmachova, et al. Ultra-large library docking for discovering new chemotypes. *Nature*, 566(7743):224–229, 2019. 1

[6] Muhammad Arslan Masood, Tianyu Cui, and Samuel Kaski. Deep Bayesian experimental design for drug discovery. In *International Workshop on AI in Drug Discovery*, pages 149–159. Springer, 2024. 1

[7] Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. Modern Bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024. 1, 3

[8] Elizabeth G Ryan, Christopher C Drovandi, James M McGree, and Anthony N Pettitt. A review of modern computational algorithms for Bayesian optimal design. *International Statistical Review*, 84(1):128–154, 2016. 1, 3

[9] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995. 1, 3

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

[11] Tom Rainforth, Rob Cornish, Hongseok Yang, Andrew Warrington, and Frank Wood. On nesting Monte Carlo estimators. In *International Conference on Machine Learning*, pages 4267–4276. PMLR, 2018. 1, 3

[12] Adam Foster, Martin Jankowiak, Elias Bingham, Paul Horsfall, Yee Whye Teh, Thomas Rainforth, and Noah Goodman. Variational Bayesian optimal experimental design. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 1, 3

[13] Adam Foster, Desi R Ivanova, Ilyas Malik, and Tom Rainforth. Deep adaptive design: Amortizing sequential Bayesian experimental design. In *International Conference on Machine Learning*, pages 3384–3395. PMLR, 2021. 1, 3, 5, 7, 22, 23, 38

[14] Desi R Ivanova, Adam Foster, Steven Kleinegesse, Michael U Gutmann, and Thomas Rainforth. Implicit deep adaptive design: Policy-based experimental design without likelihoods. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 1, 3

[15] Tom Blau, Edwin V Bonilla, Iadine Chades, and Amir Dezfouli. Optimizing sequential experimental design with deep reinforcement learning. In *International Conference on Machine Learning*, pages 2107–2128. PMLR, 2022. 1, 3, 7, 23, 24, 38

[16] Marcel Hedman, Desi R. Ivanova, Cong Guan, and Tom Rainforth. Step-DAD: Semi-amortized policy-based Bayesian experimental design. In *International Conference on Machine Learning*, pages 22904–22923. PMLR, 2025. 1, 3

[17] Daolang Huang, Xinyi Wen, Ayush Bharti, Samuel Kaski, and Luigi Acerbi. ALINE: Joint amortization for Bayesian inference and active data acquisition. In *Advances in Neural Information Processing Systems*, volume 38, 2025. 1, 3, 5, 7, 10, 23, 24, 28, 38

[18] Niels Bracher, Lars Kühmichel, Desi R Ivanova, Xavier Intes, Paul-Christian Bürkner, and Stefan T Radev. JADAI: Jointly amortizing adaptive design and Bayesian inference. In *International Conference on Machine Learning*, pages 9467–9493. PMLR, 2026. 1, 3

[19] Petrus Mikkola, Osvaldo A Martin, Suyog Chandramouli, Marcelo Hartmann, Oriol Abril Pla, Owen Thomas, Henri Pesonen, Jukka Corander, Aki Vehtari, Samuel Kaski, et al. Prior knowledge elicitation: The past, present, and future. *Bayesian Analysis*, 19(4):1129–1161, 2024. 1

[20] Petru Tighineanu, Lukas Grossberger, Paul Baireuther, Kathrin Skubch, Stefan Falkner, Julia Vinogradska, and Felix Berkenkamp. Scalable meta-learning with Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1981–1989. PMLR, 2024. 2

[21] Yucen Lily Li, Samuel Daulton, Samuel Müller, Andrew Gordon Wilson, and Eytan Bakshy. Robust transfer for Bayesian optimization with prior-data fitted networks. In *NeurIPS 2025 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2025. 2, 3, 4

[22] Michael T. Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G. Dietterich. To transfer or not to transfer. In *NIPS 2005 Workshop on Transfer Learning*, 2005. 2, 4

[23] Adam Foster, Martin Jankowiak, Matthew O'Meara, Yee Whye Teh, and Tom Rainforth. A unified stochastic gradient approach to designing Bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*, pages 2959–2969. PMLR, 2020. 3, 23

[24] Tom Blau, Iadine Chades, Amir Dezfouli, Daniel M Steinberg, and Edwin V Bonilla. Cross-entropy estimators for sequential experiment design with reinforcement learning. In *NeurIPS 2023 Workshop on Adaptive Experimental Design and Active Learning in the Real World*, 2023. 3

[25] Artur Souza, Luigi Nardi, Leonardo B Oliveira, Kunle Olukotun, Marius Lindauer, and Frank Hutter. Bayesian optimization with a prior for the optimum. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 265–296. Springer, 2021. 3

[26] Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. πBO: Augmenting acquisition functions with user beliefs for Bayesian optimization. In *International Conference on Learning Representations*, 2022. 3

[27] Carl Hvarfner, Frank Hutter, and Luigi Nardi. A general framework for user-guided Bayesian optimization. In *International Conference on Learning Representations*, 2024. 3

[28] Daolang Huang, Louis Filstroff, Petrus Mikkola, Runkai Zheng, and Samuel Kaski. Bayesian optimization augmented with actively elicited expert knowledge. *arXiv preprint arXiv:2208.08742*, 2022. 3

[29] Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance Bayesian optimization. In *International Conference on Learning Representations*, 2024. 3

[30] Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-task Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 26, 2013. 3

[31] Matthias Feurer, Benjamin Letham, and Eytan Bakshy. Scalable meta-learning for Bayesian optimization using ranking-weighted Gaussian process ensembles. In *AutoML Workshop at ICML*, 2018. 3

[32] Martin Wistuba, Nicolas Schilling, and Lars Schmidt-Thieme. Scalable Gaussian process-based transfer surrogates for hyperparameter optimization. *Machine Learning*, 107(1):43–78, 2018. 3

[33] Valerio Perrone, Rodolphe Jenatton, Matthias W Seeger, and Cédric Archambeau. Scalable hyperparameter transfer learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018. 3

[34] Katarzyna Kobalczyk and Mihaela van der Schaar. Towards automated knowledge integration from human-interpretable representations. In *International Conference on Learning Representations*, 2025. 3

[35] Paul Edmund Chang, Nasrulloh Ratu Bagus Satrio Loka, Daolang Huang, Ulpu Remes, Samuel Kaski, and Luigi Acerbi. Amortized probabilistic conditioning for optimization, simulation and inference. In *International Conference on Artificial Intelligence and Statistics*, pages 703–711, 2025. 3, 6, 19

[36] George Whittle, Juliusz Ziomek, Jacob Rawling, and Michael A Osborne. Distribution transformers: Fast approximate Bayesian inference with on-the-fly prior adaptation. In *International Conference on Machine Learning*, pages 134877–134896. PMLR, 2026. 3

[37] Sarthak Mittal, Niels Leif Bracher, Guillaume Lajoie, Priyank Jaini, and Marcus Brubaker. Amortized in-context Bayesian posterior estimation. *arXiv preprint arXiv:2502.06601*, 2025. 3

[38] Yang Yang, Severi Rissanen, Paul Edmund Chang, Nasrulloh Ratu Bagus Satrio Loka, Daolang Huang, Arno Solin, Markus Heinonen, and Luigi Acerbi. PriorGuide: Test-time prior adaptation for simulation-based inference. In *International Conference on Learning Representations*, 2026. 3

[39] Donggyun Kim, Seongwoong Cho, Wonkwang Lee, and Seunghoon Hong. Multi-task processes. In *International Conference on Learning Representations*, 2022. 3

[40] Matthew Ashman, Cristiana Diaconu, Adrian Weller, and Richard E. Turner. In-context in-context learning with Transformer neural processes. In *Proceedings of the 6th Symposium on Advances in Approximate Bayesian Inference*, volume 253, pages 1–29. PMLR, 2024. 3, 4

[41] Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom Rainforth. Prediction-oriented Bayesian active learning. In *International Conference on Artificial Intelligence and Statistics*, pages 7331–7348. PMLR, 2023. 3

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 5

[43] David Barber and Felix Agakov. The IM algorithm: a variational approach to information maximization. In *Advances in Neural Information Processing Systems*, volume 16, 2004. 5

[44] K. J. Arrow, H. B. Chenery, B. S. Minhas, and R. M. Solow. Capital-labor substitution and economic efficiency. *The Review of Economics and Statistics*, 43(3):225–250, 1961. 7, 23

[45] Xiaohong Sheng and Yu-Hen Hu. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Transactions on Signal Processing*, 53(1):44–53, 2005. 7, 22

[46] Nicolaas Prins. The psi-marginal adaptive method: How to give nuisance parameters the attention they deserve (no more, no less). *Journal of Vision*, 13(7):3–3, 2013. 10, 28

[47] Wanggang Shen, Jiayuan Dong, and Xun Huan. Variational sequential optimal experimental design using reinforcement learning. *Computer Methods in Applied Mechanics and Engineering*, 444:118068, 2025. 10

[48] Xinyue Chen, Che Wang, Zijian Zhou, and Keith W. Ross. Randomized ensembled double Q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021. 23

[49] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992. 24

[50] Edwin V Bonilla, Kian Chai, and Christopher Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems*, volume 20, 2007. 26, 38

[51] Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. In *International Conference on Machine Learning*, pages 1081–1088, 2006. 27, 31

[52] Leonid L Kontsevich and Christopher W Tyler. Bayesian adaptive estimation of psychometric slope and threshold. *Vision Research*, 39(16):2729–2737, 1999. 28

[53] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 38

[54] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 38

Appendix

The appendix is organized as follows.

- Appendix A summarizes the notation used throughout the paper.
- Appendix B derives the Barber–Agakov bound stated in Equation (8).
- Appendix C provides architecture and training details, including the construction of prior and instance tokens, the supervision of the reliability filter, and the training algorithm.
- Appendix D describes the four experimental setups, including simulator specifications, baseline implementations, and hyperparameters.
- Appendix E reports additional results for different tasks, ablation and sensitivity studies on the reliability filter, deployment shifts absent from training, the sensitivity to the candidate design pool, and the computational overhead of in-context conditioning.
- Appendix F reports the computational resources and software dependencies used in this work.

A Notation

Table 3 summarizes the notation used in the main text.

B Derivation of the Barber–Agakov bound

Proof of Equation (8). Fix the in-context knowledge C. Under the design policy π_ψ, let $p_\psi(Z, \mathcal{D}_T \mid C)$ denote the induced joint distribution over the target and the experimental trajectory, that is, the distribution of $(Z, \mathcal{D}_T)$ obtained by running π_ψ for T rounds under C. For $Z = \theta$,

$$p_\psi(\theta, \mathcal{D}_T \mid C) = p(\theta \mid C) \prod_{t=1}^{T} \pi_\psi(x_t \mid \mathcal{D}_{t-1}, C)\, p(y_t \mid \theta, x_t), \tag{13}$$

where $p(\theta \mid C)$ is the distribution of the parameters given C (Algorithm 2 specifies it during training); for $Z = y^\star$ the analogous form additionally draws $x^\star \sim p^\star(x^\star)$ and $y^\star \sim p(y^\star \mid \theta, x^\star)$. The conditional information gain equals

$$\mathcal{U}(\psi; C) = H(Z \mid C) - H_{p_\psi}(Z \mid \mathcal{D}_T, C),$$

where $H(Z \mid C)$ depends only on the prior $p(Z \mid C)$ and $H_{p_\psi}(Z \mid \mathcal{D}_T, C)$ is taken under the policy-induced joint $p_\psi(Z, \mathcal{D}_T \mid C)$. Writing the conditional entropy as an expectation,

$$H_{p_\psi}(Z \mid \mathcal{D}_T, C) = -\mathbb{E}_{p_\psi(Z, \mathcal{D}_T \mid C)}\big[\log p_\psi(Z \mid \mathcal{D}_T, C)\big],$$

we obtain

$$\mathcal{U}(\psi; C) = H(Z \mid C) + \mathbb{E}_{p_\psi(Z, \mathcal{D}_T \mid C)}\big[\log p_\psi(Z \mid \mathcal{D}_T, C)\big].$$

For any variational distribution $q_\phi(Z \mid \mathcal{D}_T, C)$, the expected Kullback–Leibler divergence is non-negative,

$$\begin{aligned} 0 &\le \mathbb{E}_{p_\psi(\mathcal{D}_T \mid C)}\Big[\mathrm{KL}\big(p_\psi(Z \mid \mathcal{D}_T, C) \,\|\, q_\phi(Z \mid \mathcal{D}_T, C)\big)\Big] \\ &= \mathbb{E}_{p_\psi(Z, \mathcal{D}_T \mid C)}\big[\log p_\psi(Z \mid \mathcal{D}_T, C) - \log q_\phi(Z \mid \mathcal{D}_T, C)\big], \end{aligned}$$

so that

$$\mathbb{E}_{p_\psi(Z, \mathcal{D}_T \mid C)}\big[\log p_\psi(Z \mid \mathcal{D}_T, C)\big] \ge \mathbb{E}_{p_\psi(Z, \mathcal{D}_T \mid C)}\big[\log q_\phi(Z \mid \mathcal{D}_T, C)\big].$$

Substituting this inequality into the mutual-information identity yields

$$\mathcal{U}(\psi; C) \ge H(Z \mid C) + \mathbb{E}_{p_\psi(Z, \mathcal{D}_T \mid C)}\big[\log q_\phi(Z \mid \mathcal{D}_T, C)\big],$$

which is the bound in Equation (8).

Table 3: Notation used in the main text.

Symbol	Meaning
Problem setup (Section 3.1)	
$\theta \in \Theta \subseteq \mathbb{R}^L$	Latent parameters of the system; L is the parameter dimension
$\theta^\star$	Ground-truth parameters of a training rollout, drawn from the prior; unknown at deployment
$x \in \mathcal{X}$, y	Design and outcome; $p(y \mid \theta, x)$ is the likelihood
T, $\mathcal{D}_t$	Horizon of the experiment, and history $\{(x_i, y_i)\}_{i=1}^t$ after t rounds with $\mathcal{D}_0 = \emptyset$
Z	Inference target: $Z = \theta$ (parameter-oriented) or $Z = y^\star$ at test inputs $x^\star \sim p^\star$ (prediction-oriented)
Rollout	One T-round sequential experiment, simulated during training
$\pi_\psi(x_t \mid \mathcal{D}_{t-1}, C)$	Design policy with parameters ψ
$q_\phi(Z \mid \mathcal{D}_t, C)$	Variational posterior (inference network) with parameters ϕ
$p_\psi(Z, \mathcal{D}_T \mid C)$	Joint distribution of target and trajectory induced by running π_ψ under C (Equation (13))
$\mathcal{U}(\psi; C)$, $\mathcal{J}(\psi, \phi; C)$	Information objective (Equation (1)) and the variational term of its lower bound (Equation (8))
In-context knowledge (Section 3.2)	
$C = (C_{\text{prior}}, C_{\text{obs}})$, $\lvert C \rvert$	In-context knowledge, consisting of the prior channel and the previous-instance channel (either may be empty), and its number of in-context tokens (L prior tokens or K instance tokens)
ρ_l, $\boldsymbol{\rho}_l \in \mathbb{R}^{N_B}$	User-specified marginal belief over θ_l, and its normalized histogram over N_B bins (Equation (2))
f_{prior}, $\boldsymbol{\zeta}_l$, $\mathbf{e}_l^{\text{prior}}$	Histogram MLP, parameter-specific embedding, and prior token (Equation (3))
$\mathcal{A}_k$, K	Observations $\{(x_i^{(k)}, y_i^{(k)})\}_{i=1}^T$ of previous instance k, and the number of instances
$\theta^{(k)}$	Parameters that generated instance k
g_{aux}, n_s, $\mathbf{r}_k$	Per-instance encoder, its number of summary slots, and the instance token (Equation (4))
$\mathbf{e}_j$, D	An in-context token (prior or instance), and the model dimension
Reliability filter (Section 3.3)	
$\mathbf{E}_t$	Embeddings of the experimental history $\mathcal{D}_t$
f_ω, $\alpha_j^{(t)}$, τ	Filter with parameters ω, reliability score of token j (Equation (5)), and keep-or-discard threshold (0.5)
r_j^{prior}, r_j^{inst}, c_j	Oracle reliability labels (Equations (6) and (7)) and their binary version $c_j = \mathbb{I}[r_j > 1/2]$
$b^\star$, β, κ	Histogram bin containing $\theta_l^\star$, and the prior and instance label sharpness
$[r_l^{\min}, r_l^{\max}]$	Support (or finite simulation range) of θ_l used for normalization
p_{rel}, Σ_{rel}	Probability of the relevant regime (0.7), and covariance of the relevant-regime perturbation (standard deviation 0.1 of the support per dimension)
p_{good}, p_{uniform}	Mixture probabilities of the prior regimes during training ($p_{\text{good}} = 0.7$, $p_{\text{uniform}} = 0.2$; Algorithm 2)
Training (Section 3.4)	
$\mathcal{L}_{\text{policy}}$, $\mathcal{L}_{\text{inf}}$, $\mathcal{L}_{\text{filter}}$	Policy, inference, and filter losses, with weights $\lambda_1, \lambda_2, \lambda_3$ (Equation (9))
R_t, γ	Per-step reward, and discount factor
N_q	Size of the candidate design pool
E, E_{warm}	Total and warm-up training epochs

C Further details

C.1 Training algorithm

Algorithm 1 summarizes the joint training procedure for the design policy π_ψ, the inference network q_ϕ, and the reliability filter f_ω described in Section 3.4. Algorithm 2 details the sampling of the in-context knowledge C and the ground-truth $\theta^\star$ used in each rollout. The pseudocode presents

the empty-history case. For GP active learning, initialize $\mathcal{D}_0$ with one randomly sampled target observation instead.

Algorithm 1 Training procedure for IMBUE.

Require: prior $p(\theta)$; simulator $p(y \mid \theta, x)$; candidate pool size N_q^{train}; rollout length T; warm-up epochs E_{warm}; total epochs E; loss coefficients $\lambda_1, \lambda_2, \lambda_3$
Ensure: trained parameters (ψ, ϕ, ω)

for epoch $= 1, \ldots, E$ **do**
  for each minibatch **do**
    Sample in-context knowledge C and ground-truth $\theta^\star$ following Algorithm 2
    Compute reliability labels $\{c_j\}$ for tokens in C (Equations (6) and (7))
    Initialize $\mathcal{D}_0 \leftarrow \emptyset$ and $\log q_\phi^{(0)} \leftarrow \log q_\phi(Z^\star \mid \mathcal{D}_0, C)$
    for $t = 0, 1, \ldots, T-1$ **do**
      if epoch $\le E_{\text{warm}}$ **then**
        Sample x_{t+1} uniformly from a pool of size N_q^{train}
      else
        Sample $x_{t+1} \sim \pi_\psi(\cdot \mid \mathcal{D}_t, C)$ from a pool of size N_q^{train}
      end if
      Observe $y_{t+1} \sim p(y \mid \theta^\star, x_{t+1})$
      Update $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x_{t+1}, y_{t+1})\}$
      Compute in-context token embeddings and reliability scores $\{\alpha_j^{(t+1)}\}$ (Equation (5))
      Exclude in-context tokens with $\alpha_j^{(t+1)} \le \tau$
      Compute $\log q_\phi^{(t+1)} \leftarrow \log q_\phi(Z^\star \mid \mathcal{D}_{t+1}, C)$
      Compute reward $R_{t+1} \leftarrow \log q_\phi^{(t+1)} - \log q_\phi^{(t)}$
    end for
    Compute $\mathcal{L}_{\text{inf}}$ from $\{\log q_\phi^{(t)}\}_{t=0}^{T}$ (Equation (10))
    Compute $\mathcal{L}_{\text{filter}}$ from $\{\alpha_j^{(t)}\}_{t=1}^{T}$ and $\{c_j\}$ (Equation (12))
    if epoch $\le E_{\text{warm}}$ **then**
      $\mathcal{L} \leftarrow \lambda_2 \mathcal{L}_{\text{inf}} + \lambda_3 \mathcal{L}_{\text{filter}}$
    else
      Compute $\mathcal{L}_{\text{policy}}$ from $\{R_t\}_{t=1}^{T}$ and $\{\log \pi_\psi(x_t \mid \mathcal{D}_{t-1}, C)\}_{t=1}^{T}$ (Equation (11))
      $\mathcal{L} \leftarrow \lambda_1 \mathcal{L}_{\text{policy}} + \lambda_2 \mathcal{L}_{\text{inf}} + \lambda_3 \mathcal{L}_{\text{filter}}$ (Equation (9))
    end if
    Update (ψ, ϕ, ω) by an AdamW step on $\nabla \mathcal{L}$
  end for
end for

C.2 Architecture and training details

Token embedders. Designs x are mapped to the model dimension D by a two-layer MLP f_x with feedforward dimension 128 and output dimension $D = 32$, and observations y by an analogous MLP f_y. Each context token combines the design and outcome embeddings, $\mathbf{e}_n^{\text{ctx}} = f_x(x_n) + f_y(y_n)$, and each query token uses the design embedder alone, $\mathbf{e}_m^{\text{qry}} = f_x(x_m^q)$. The construction of target tokens depends on the target type. For parameter targets $Z = \theta$, the L target tokens $\{\boldsymbol{\zeta}_l\}_{l=1}^{L}$ are learnable embeddings, one per parameter dimension. These are the same $\boldsymbol{\zeta}_l$ that bind prior tokens to their parameters in Equation (3). For predictive targets $Z = y^\star$ at test inputs $\{x_m^\star\}_{m=1}^{M}$, each target token is the embedding of the corresponding test input, $\mathbf{e}_m^{\text{tgt}} = f_x(x_m^\star)$, sharing the design embedder with the context and query tokens. In-context tokens are constructed as described in Appendix C.3 for prior tokens and in Appendix C.4 for instance tokens. All four token groups share the embedding dimension $D = 32$.

Shared encoder. The shared encoder is a 3-layer Transformer with 4 attention heads per layer and feedforward dimension 128. Information flow among the four token groups is controlled by the structured attention mask described in Section 3.4 (Figure 7), together with a per-example key-

Algorithm 2 Sampling in-context knowledge C and ground-truth $\theta^\star$ during training.

Require: prior $p(\theta)$; mode $\in$ {prior, previous instances}; number of previous instances $K \sim \mathrm{Unif}\{0, \dots, K_{\max}\}$

if mode is prior **then**
  Draw the prior regime: uniform with probability p_{uniform}, Gaussian otherwise
  Sample a marginal ρ_l for each parameter l under the drawn regime (Appendix C.3)
  Discretize each ρ_l into a histogram $\boldsymbol{\rho}_l$ over N_B bins (Equation (2))
  for each parameter $l = 1, \dots, L$ **do**
    Sample $\theta^\star_l \sim \boldsymbol{\rho}_l$ (calibrated) with probability p_{good}; otherwise displace the draw away from ρ_l (miscalibrated) (Equation (15))
  end for
  Encode each $\boldsymbol{\rho}_l$ into a prior token $\mathbf{e}_l^{\text{prior}}$ (Equation (3))
  Set $C \leftarrow \{\mathbf{e}_l^{\text{prior}}\}_{l=1}^{L}$
else
  Sample $\theta^\star \sim p(\theta)$
  for $k = 1, \dots, K$ **do**
    Draw the regime, relevant with probability p_{rel} and irrelevant otherwise
    if relevant **then**
      Sample $\theta^{(k)} = \theta^\star + \eta_k$ with $\eta_k \sim \mathcal{N}(0, \Sigma_{\text{rel}})$
    else
      Sample $\theta^{(k)} \sim p(\theta)$ independently of $\theta^\star$
    end if
    Sample designs $x_1^{(k)}, \dots, x_T^{(k)}$ uniformly from $\mathcal{X}$
    Sample observations $y_i^{(k)} \sim p(y \mid \theta^{(k)}, x_i^{(k)})$
    Encode $\mathcal{A}_k = \{(x_i^{(k)}, y_i^{(k)})\}_{i=1}^{T}$ into an instance token $\mathbf{r}_k$ (Equation (4))
  end for
  Set $C \leftarrow \{\mathbf{r}_k\}_{k=1}^{K}$
end if
return $C, \theta^\star$

padding mask produced by the reliability filter (Section 3.3) that excludes in-context tokens with $\alpha_j^{(t)} \le \tau = 0.5$.



Figure 7: Attention masks of IMBUE's shared Transformer layers for the previous-instance channel (left, in-context tokens $\mathbf{r}_1, \dots, \mathbf{r}_K$) and the prior channel (right, in-context tokens $\mathbf{e}_1^{\text{prior}}, \dots, \mathbf{e}_L^{\text{prior}}$). Rows index queries and columns index keys, ordered as in-context tokens, context tokens (x_i, y_i), query tokens x_m^{q}, and target tokens $\boldsymbol{\zeta}_l$. Shaded squares indicate allowed attention.

Reliability filter. The filter f_ω takes an in-context token as the query and the context tokens as keys and values, all detached from the main computation graph (stop-gradient) so that gradients from the filter loss do not propagate into the embedders shared with the rest of the network. Query and key-value pairs are processed by a Transformer-style cross-attention block consisting of multi-head cross-attention with 4 heads, an Add&LayerNorm, a position-wise feedforward network (linear, ReLU, dropout, linear), and a second Add&LayerNorm. The resulting representation is passed through a two-layer scoring MLP with hidden dimension 16 to produce the scalar logit, which is passed through a sigmoid to give the reliability score α_j in Equation (5).

Output heads. The inference head parameterizes a Gaussian mixture with 10 components, produced by two-layer MLPs each of hidden dimension 128. For each component, the MLP outputs a raw weight, mean, and standard deviation. Weights are normalized through a softmax, standard deviations are passed through softplus with a floor $\sigma_{\text{min}} = 10^{-4}$, and means are unconstrained. The design head is a two-layer MLP with hidden dimension 128 that maps each query position to a scalar logit, followed by a softmax over the N_q candidates to form π_ψ.

Optimization. We use AdamW with initial learning rate 10^{-3} and the default weight decay 10^{-2}. The learning rate follows a cosine annealing schedule that decays to zero over the full training budget. Training follows the two-stage schedule of Section 3.4. The first stage runs for E_{warm} epochs with the policy gradient disabled and designs drawn uniformly from a candidate pool of size N_q^{train}. The second stage activates the full joint objective in Equation (9) on the same pool. Loss coefficients are fixed at $\lambda_1 = \lambda_2 = \lambda_3 = 1$ across all benchmarks. Benchmark-specific values of E_{warm}, N_q^{train}, T, and other task-dependent hyperparameters are listed in Table 4.

Table 4: Benchmark-specific hyperparameters for IMBUE.

	Location Finding	CES	GP active learning	Psychometric
Experiment setup				
Rollout length T	20	10	15	30
Parameter dimension L	4	5	—	4
Target type Z	θ	θ	$y^\star$	θ
Training schedule				
Training-time candidate pool size N_q^{train}	200	200	200	200
Total epochs E	70,000	250,000	250,000	120,000
Warm-up epochs E_{warm}	15,000	30,000	30,000	15,000
Batch size	200	200	200	200
Discount factor γ	1	1	1	1
Evaluation schedule				
Evaluation-time candidate pool size N_q^{eval}	2,000	2,000	500	200
Contrastive samples	10^6	10^7	—	—
Evaluation-time target set size	4	5	100	4
In-context knowledge				
Number of histogram bins N_B	100	—	—	—
Number of summary slots n_s	—	2	4	4
Maximum $K_{\max}$ at training	—	5	5	5
Reliability filter				
Threshold τ	0.5	0.5	0.5	0.5
Instance label sharpness κ	—	5	—	4

C.3 Prior tokens

This section describes how prior tokens are generated, embedded, and labeled. The histogram representation follows ACE [35].

Histogram representation and embedding. For each parameter θ_l, the user-supplied marginal ρ_l is discretized into a histogram over $N_B = 100$ uniformly spaced bins covering the support of θ_l, normalized as in Equation (2). The histogram is mapped to the model dimension by a two-layer MLP

$f_{\text{prior}} : \mathbb{R}^{N_B} \to \mathbb{R}^D$ and added to the parameter-specific embedding $\boldsymbol{\zeta}_l$ shared with the target token,

$$\mathbf{e}_l^{\text{prior}} = f_{\text{prior}}(\boldsymbol{\rho}_l) + \boldsymbol{\zeta}_l \in \mathbb{R}^D. \tag{14}$$

The choice of $N_B = 100$ was not tuned in our experiments.

Prior generation during training. Each rollout draws one of two regimes.

Uniform regime (probability $p_{\text{uniform}} = 0.2$). Every marginal is flat on its support, $\rho_l = \mathcal{U}(r_l^{\min}, r_l^{\max})$. The tokens carry no information about $\theta^\star$, and Equation (6) assigns them $r^{\text{prior}} = 1/2$ regardless of where $\theta^\star$ falls.

Gaussian regime (probability $1 - p_{\text{uniform}} = 0.8$). Each marginal is a Gaussian $\rho_l = \mathcal{N}(\mu_l, \sigma_l^2)$ with a random location $\mu_l \sim \mathcal{U}(r_l^{\min}, r_l^{\max})$ and a random width $\sigma_l \sim \mathcal{U}(\sigma^{\min}, \sigma^{\max}) \cdot (r_l^{\max} - r_l^{\min})$. For the Location Finding task, we use $\sigma^{\min} = 0.05$ and $\sigma^{\max} = 0.3$.

In both regimes, $\theta^\star$ is then drawn parameter-wise from the histograms. With probability p_{good} a parameter is *calibrated*: $\theta_l^\star \sim \boldsymbol{\rho}_l$, drawn as a bin from the histogram followed by a uniform draw within that bin, so that the prior places mass near the truth. Otherwise it is *miscalibrated*: the same draw θ_l^{cal} is displaced by a random fraction of the support toward its far side,

$$\theta_l^\star = \text{clip}\big(\theta_l^{\text{cal}} + d_l s_l,\ r_l^{\min},\ r_l^{\max}\big), \qquad s_l \sim \mathcal{U}(0.3,\ 0.5) \cdot (r_l^{\max} - r_l^{\min}), \tag{15}$$

where $d_l = +1$ if θ_l^{cal} lies below the midpoint of $[r_l^{\min}, r_l^{\max}]$ and $d_l = -1$ otherwise. Figure 8 shows examples of sampled priors.

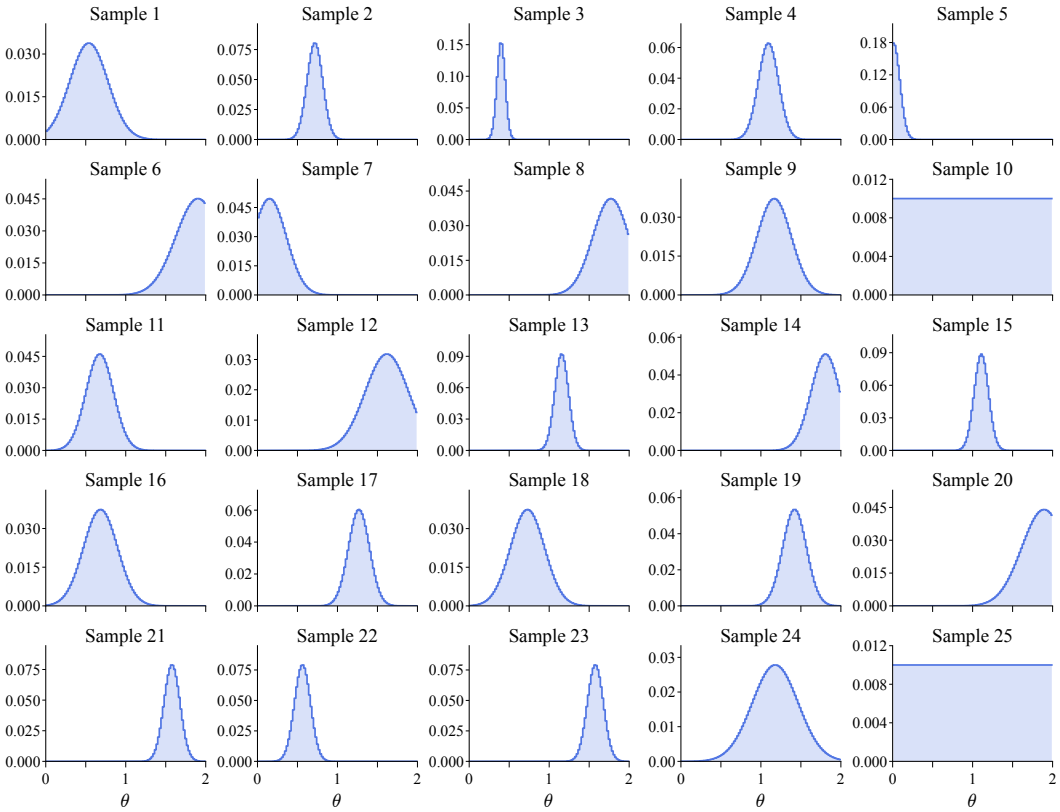


Figure 8: Illustrative examples of randomly sampled prior histograms, shown on a generic support $[0, 2]$; the Location Finding experiments use the support $[0, 1]$.

Evaluation-time prior conditions. At evaluation time we probe IMBUE's behavior under four prior conditions that span the reliability spectrum: three Gaussian priors, each with a concentration chosen from within the training range, and a uniform prior.

- *Strong prior.* A Gaussian with μ_l centered near $\theta^\star$, with $\sigma_l = 0.1 \cdot (r_l^{\max} - r_l^{\min})$, modeling a confident and accurate expert. On the Location Finding support this corresponds to $\sigma = 0.1$.
- *Weak prior.* The same form as the strong prior but with a wider $\sigma_l = 0.25 \cdot (r_l^{\max} - r_l^{\min})$, modeling a less confident expert who still places probability mass near $\theta^\star$ but spreads it more broadly. On the Location Finding support this corresponds to $\sigma = 0.25$.
- *Miscalibrated prior.* A shifted Gaussian following the training miscalibrated sub-regime, with concentration matched to the strong prior so that the comparison isolates the effect of an incorrect mean from the level of confidence.
- *Uniform prior.* A flat histogram, modeling an expert who provides no information about $\theta^\star$ beyond its support. Under this condition the reliability filter should mask out the prior tokens, so that IMBUE approximates the no-external baseline.

All four evaluation conditions fall within the training-time mixture described above. The main text reports results under the strong and miscalibrated regimes, and Figure 14 reports weak and uniform prior results.

Reliability label distribution. Figure 9 compares the reliability labels r_j^{prior} produced during training with those induced by the four evaluation conditions on the Location Finding task. These are the scalar labels of Equation (6) with $\beta = 2$; β only sharpens them toward 0 and 1 and leaves the binary targets c_j unchanged. The training distribution covers both reliable and unreliable prior tokens, and the evaluation conditions concentrate in the expected regions. Strong priors concentrate near label 1, miscalibrated priors near label 0, weak priors at intermediate values, and uniform priors near $1/2$. The evaluation reliability is therefore in distribution for the filter.

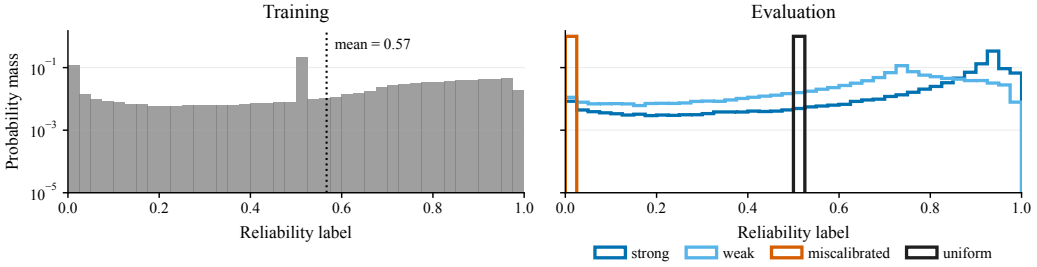


Figure 9: Distribution of prior reliability labels on the Location Finding task. Training (left) versus the four evaluation regimes (right), shown as normalized densities on a log y-axis. The training distribution covers the full range, while each evaluation regime concentrates in the expected region.

C.4 Instance tokens

This section describes how instance tokens are generated and how their reliability labels are computed.

Representation. The previous-instance channel is a collection of K instances,

$$C_{\text{obs}} = \{\mathcal{A}_k\}_{k=1}^{K}, \qquad \mathcal{A}_k = \{(x_i^{(k)}, y_i^{(k)})\}_{i=1}^{T}, \tag{16}$$

each generated from the same simulator as the target experiment but under a different parameter $\theta^{(k)}$. The horizon T matches the target experiment.

Generation during training. For each rollout, $\theta^\star$ is sampled from $p(\theta)$ and each of the K instances $\mathcal{A}_k$ is independently assigned to one of two regimes.

Relevant regime. The instance parameter is a perturbation of $\theta^\star$,

$$\theta^{(k)} = \theta^\star + \eta_k, \qquad \eta_k \sim \mathcal{N}(0, \Sigma_{\text{rel}}), \tag{17}$$

with $\Sigma_{\text{rel}} = \text{diag}(\sigma_{\text{rel},l}^2)$ and $\sigma_{\text{rel},l} = 0.1\,(r_l^{\max} - r_l^{\min})$.

Irrelevant regime. The instance parameter is drawn independently of $\theta^\star$, $\theta^{(k)} \sim p(\theta)$.

Conditional on $\theta^{(k)}$, designs $x_1^{(k)}, \ldots, x_T^{(k)}$ are sampled uniformly from $\mathcal{X}$ and outcomes are simulated from $p(y \mid \theta^{(k)}, x_i^{(k)})$. Random designs reflect the assumption that previous instances were collected by experimenters whose design choices need not align with ours.

Per-instance encoder. We detail the implementation of the per-instance encoder g_{aux} defined in Equation (4). Each design–outcome pair is embedded as $\mathbf{u}_i^{(k)} = f_x(x_i^{(k)}) + f_y(y_i^{(k)}) \in \mathbb{R}^D$, using the same MLPs f_x and f_y as the context tokens (Appendix C.2). We prepend n_s learnable summary slots $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_{n_s})$ to these embeddings and pass the joint sequence through an n_{aux}-layer Transformer. Let $\mathbf{h}_s^{(k)} \in \mathbb{R}^D$ denote its output at summary slot s for the k-th instance. The complete encoder mapping includes mean pooling over these slot outputs,

$$\mathbf{r}_k = g_{\text{aux}}(\mathcal{A}_k) = \frac{1}{n_s} \sum_{s=1}^{n_s} \mathbf{h}_s^{(k)} \in \mathbb{R}^D. \tag{18}$$

We use $n_{\text{aux}} = 3$ layers in all experiments. The number of summary slots for each task is given in Table 4. The slots interact with the design–outcome tokens through self-attention, and the final mean keeps the main-encoder input length at one token per instance.

Reliability labels. For parameter-oriented BED, an instance token $\mathbf{e}_j = \mathbf{r}_k$ representing instance $\mathcal{A}_k$ receives the label r_j^{inst} of Equation (7), thresholded at $1/2$ to give the binary label c_j. Table 5 lists the normalization ranges $[r_l^{\min}, r_l^{\max}]$ for CES and Psychometric, the two parameter-oriented benchmarks that use the instance channel, and Table 4 gives their sharpness κ. For prediction-oriented BED, the regime-based binary labels are described in Appendix D.3.

Table 5: Normalization ranges $[r_l^{\min}, r_l^{\max}]$ used to construct reliability labels for previous instances.

Task	Parameter	$r_l^{\min}$	$r_l^{\max}$
CES	χ	0.05	1
	w_1	0	1
	w_2	0	1
	w_3	0	1
	$\log u$	-8	8
Psychometric	$\theta_{\text{threshold}}$	-3	3
	θ_{slope}	0.1	2
	θ_{guess}	0	0.15
	θ_{lapse}	0	0.06

Evaluation-time generation. We evaluate IMBUE under two configurations that vary the number of instances K while fixing their regime. K *relevant* draws all K instances from the relevant regime, and K *irrelevant* draws all K instances from the irrelevant regime, testing how performance scales with the amount of supplied information. Mixtures of relevant and irrelevant instances are evaluated on GP active learning, whose robustness sweep is described in Appendix D.3. The covariance Σ_{rel} used at evaluation matches the training value.

D Experimental details

Unless stated otherwise, $\pm$ values and shaded bands denote 95% confidence intervals of the mean over evaluation rollouts, computed as ± 1.96 standard errors.

D.1 BED tasks

Location Finding. We adopt the source location task of Sheng and Hu [45] as formulated by Foster et al. [13]. The goal is to localize N_{src} hidden sources from intensity measurements at sensor locations $\mathbf{x} \in \mathbb{R}^d$. The expected signal at $\mathbf{x}$ follows an inverse-square attenuation law,

$$\mu(\boldsymbol{\theta}, \mathbf{x}) = b_{\text{bg}} + \sum_{s=1}^{N_{\text{src}}} \frac{\xi_s}{m + \|\boldsymbol{\theta}_s - \mathbf{x}\|^2}, \tag{19}$$

where $\xi_s > 0$ is the strength of source s, $b_{\text{bg}} > 0$ is a background term, and $m > 0$ is a small damping constant. The observation is log-normal,

$$\log y \mid \boldsymbol{\theta}, \mathbf{x} \sim \mathcal{N}\big(\log \mu(\boldsymbol{\theta}, \mathbf{x}),\, \sigma^2\big). \tag{20}$$

The prior on each source location $\boldsymbol{\theta}_s$ is uniform on $[0,1]^d$. We use $N_{\text{src}} = 2$, $d = 2$, $\xi_1 = \xi_2 = 1$, $b_{\text{bg}} = 0.1$, $m = 10^{-4}$, and $\sigma = 0.5$.

Constant Elasticity of Substitution (CES). The CES task originates with Arrow et al. [44] and was adapted to the BED setting by Foster et al. [23]. The experimenter infers preference parameters by presenting pairs of bundles to a synthetic participant whose utility for a bundle of N_{goods} goods, $\mathbf{z} \in [0, 100]^{N_{\text{goods}}}$, is

$$U(\mathbf{z}) = \Big(\sum_{i=1}^{N_{\text{goods}}} w_i z_i^{\chi} \Big)^{1/\chi}. \tag{21}$$

The latent parameters are $\boldsymbol{\theta} = (\chi, \boldsymbol{w}, u)$ with priors

$$\chi \sim \text{Beta}(1, 1), \tag{22}$$

$$\boldsymbol{w} \sim \text{Dirichlet}(\mathbf{1}_{N_{\text{goods}}}), \tag{23}$$

$$\log u \sim \mathcal{N}(1, 3^2). \tag{24}$$

A design is a pair $\mathbf{x} = (\mathbf{z}, \mathbf{z}')$, and the response is generated by adding Gaussian noise to the utility difference and passing the result through a clipped sigmoid,

$$\eta \sim \mathcal{N}\Big(u\,(U(\mathbf{z}) - U(\mathbf{z}')),\; u^2 \sigma_{\text{CES}}^2 (1 + \|\mathbf{z} - \mathbf{z}'\|)^2 \Big), \tag{25}$$

$$y = \text{clip}\big(\sigma(\eta),\, \epsilon,\, 1 - \epsilon\big), \tag{26}$$

where σ here denotes the logistic sigmoid. We use $N_{\text{goods}} = 3$, $\sigma_{\text{CES}} = 0.005$, and $\epsilon = 2^{-22}$, giving $\boldsymbol{\theta} \in \mathbb{R}^5$ and a six-dimensional design space.

D.2 BED baselines

This subsection describes the baselines used on Location Finding and CES. For the random-design reference, we draw each x_t uniformly from the design space.

Deep Adaptive Design (DAD). DAD [13] learns an amortized design policy by stochastic gradient ascent on the sequential Prior Contrastive Estimation (sPCE) lower bound. For a policy π and $L_S \geq 0$ contrastive samples, the sPCE bound on the total information gain over T steps is

$$\mathcal{L}_T(\pi, L_S) = \mathbb{E}_{p(\theta_0, \mathcal{D}_T \mid \pi)} \left[\log \frac{p(\mathcal{D}_T \mid \theta_0, \pi)}{\frac{1}{L_S+1} \sum_{n=0}^{L_S} p(\mathcal{D}_T \mid \theta_n, \pi)} \right], \tag{27}$$

where the contrastive samples $\theta_{1:L_S}$ are drawn independently from the prior $p(\theta)$. The DAD policy comprises a permutation-invariant encoder that aggregates the experimental history through pooling and an emitter that outputs the next design, following the architecture of Foster et al. [13]. Training uses Adam with an initial learning rate of 5×10^{-5} and $(\beta_1, \beta_2) = (0.8, 0.998)$, together with an exponential schedule that decays the rate by a factor of 0.98 every $1{,}000$ epochs.

RL-BOED. RL-BOED [15] casts the design problem as a Markov decision process and learns the design policy by reinforcement learning, using a stepwise reward that estimates the marginal contribution of the t-th step to the sequential EIG. The policy network shares its backbone with DAD but replaces the deterministic emitter by a Tanh-Gaussian distribution. The encoder is a two-layer MLP with 128 ReLU units followed by a linear output of 64 units. We train with Randomized Ensembled Double Q-learning (REDQ) [48] using the configuration in Table 6.

ALINE. ALINE [17] jointly trains the design policy and a posterior approximation network in a shared Transformer encoder. Tokens for the experimental history, candidate designs, and target slots are processed jointly, with attention masks that route information among the three groups. A

Table 6: Hyperparameters for RL-BOED [15] on the two BED tasks.

Parameter	Location Finding	CES
Critics	2	2
Random subsets	2	2
Training epochs	10^5	10^5
Discount factor γ	0.9	0.9
Target update rate	10^{-3}	5×10^{-3}
Policy learning rate	10^{-4}	3×10^{-4}
Critic learning rate	3×10^{-4}	3×10^{-4}
Replay buffer size	10^7	10^6

design head produces a categorical distribution over the candidate designs and an inference head parameterizes a Gaussian-mixture posterior over the latent parameters. The policy is trained with REINFORCE [49] using a per-step reward defined as the one-step gain in posterior log-probability of the ground-truth target, and the inference network is trained by maximum likelihood. Training uses AdamW with learning rate 10^{-3} and weight decay 10^{-2}. The remaining hyperparameters are summarized in Table 7.

Table 7: Hyperparameters for ALINE [17].

Parameter	Location Finding	CES	GP active learning	Psychometric
Transformer layers	3	3	3	3
Attention heads	4	4	4	4
GMM components	10	10	10	10
Batch size	200	200	200	200
Candidate pool size N_q^{eval}	2,000	2,000	500	200
Total epochs E	10^5	2×10^5	2.5×10^5	1.2×10^5

ALINE-aux. ALINE-aux shares IMBUE's backbone and is retrained on the same $70/30$ relevant/irrelevant mixture of previous instances (Section 3.3), receiving the $K \cdot T$ previous-instance observations by concatenation to the experimental history during both training and deployment. It therefore has the opportunity to learn to ignore unreliable instances on its own, and the gap between ALINE-aux and IMBUE isolates the token interface and the filter from the effect of training exposure.

Retrieve-then-condition. Retrieve-then-condition is a knowledge-selection baseline built on ALINE-aux. For each previous instance $\mathcal{A}_k$ we fit a single parameter estimate $\hat{\theta}^{(k)}$ by MAP estimation under the prior. At each round t we score every instance by how well its estimate explains the data collected so far in the live experiment, $\sum_{(x,y)\in\mathcal{D}_t} \log p(y \mid \hat{\theta}^{(k)}, x)$, keep the two best-scoring instances, and pass them to ALINE-aux by concatenation.

Deployment-time concatenation. DAD*, RL-BOED*, and ALINE* receive the K previous instances without retraining by concatenating their $K \cdot T$ design–outcome pairs to the experimental history: as additional context tokens for ALINE*, and as additional elements of the pooled history for the encoders of DAD* and RL-BOED*.

D.3 Active learning for regression

Synthetic training samples. We train IMBUE on synthetic Gaussian process samples. Each training rollout draws a kernel uniformly from {RBF, Matérn 3/2, Matérn 5/2}; Matérn 1/2 is excluded from the training family because its sample paths are too rough to be informative for our active learning targets. The output scale is sampled from $\mathcal{U}(0.1, 1)$ and the lengthscale from $\mathcal{U}(0.1, 2)$. Inputs are drawn uniformly from $[-5, 5]$, and outputs are perturbed by Gaussian noise with standard deviation $\sigma_{\text{train}} = 0.01$. Examples of training samples are shown in Figure 10.

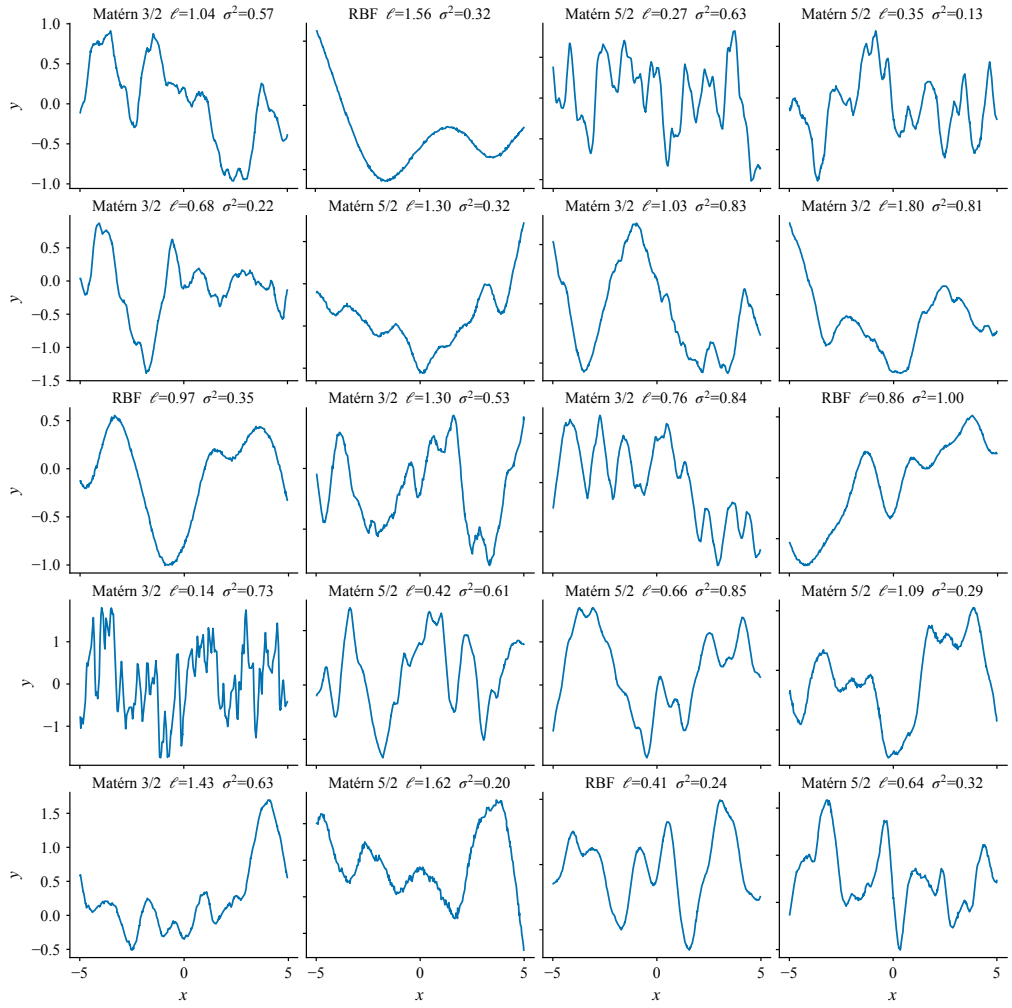


Figure 10: Examples of synthetic 1D GP functions used to train IMBUE, each drawn from the kernel and hyperparameter prior described above with noisy point evaluations.

Benchmark functions. We evaluate on four 1D benchmarks: Synthetic, Forrester, Higdon, and Gramacy & Lee. Synthetic uses fresh draws from the training GP prior on $[-5, 5]$. The three analytic functions in Figure 11 have native input domains $[0, 1]$, $[0, 20]$, and $[0.5, 2.5]$, respectively. For model inputs, each native domain is affinely mapped to $x_{\text{norm}} \in [-5, 5]$, matching the training input range. To reduce differences in output magnitude across benchmarks, we divide Forrester outputs by 5 and Gramacy & Lee outputs by 3; RMSE is reported on the resulting output scale.

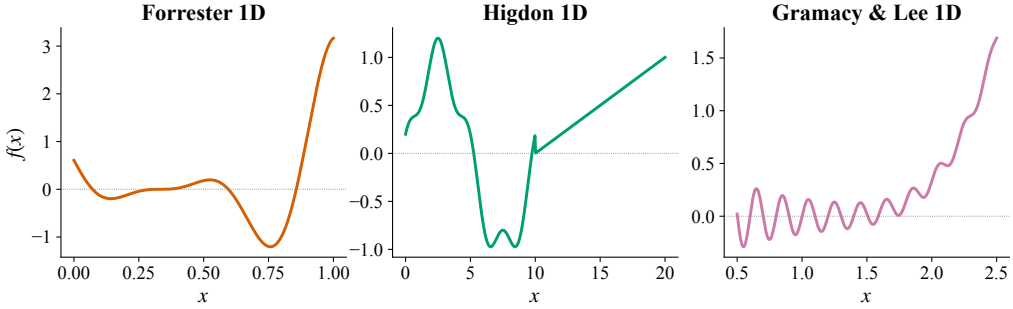


Figure 11: The three analytic 1D benchmark functions with the output scaling used at evaluation. The horizontal axes retain their native input coordinates.

Instance generation. Each rollout supplies K previous instances, of which K_{rel} are drawn from the relevant regime and $K - K_{\text{rel}}$ from the irrelevant regime. During training $K \sim \text{Unif}\{0, 1, \ldots, 5\}$; in the robustness sweep of Section 4.5, $K = 5$ is fixed and K_{rel} is swept from 0 to 5. All instances share T design points $x_1, \ldots, x_T$ drawn uniformly in x_{norm}; their outputs are generated as follows, and examples are shown in Figure 12.

Relevant regime. On the three analytic benchmarks, a relevant instance is a per-instance affine perturbation of the target function,

$$y_i^{(k)} = a_k \, f^\star(x_i) + b_k + \varepsilon_i^{(k)}, \qquad i = 1, \ldots, T, \tag{28}$$

with $a_k \sim \mathcal{U}(0.5, 1.5)$, $b_k \sim \mathcal{U}(-0.5, 0.5) \cdot \sigma_y$ where σ_y is the standard deviation of $f^\star$ over the input domain, and $\varepsilon_i^{(k)} \sim \mathcal{N}(0, \sigma^2_{\text{train}})$. The per-instance scale and offset ensure that simply pooling the instances with the target data does not recover $f^\star$ exactly. On the synthetic benchmark, where $f^\star$ is itself a GP draw, the target and the instance are drawn jointly from the same GP (same kernel and hyperparameters, noise level σ_{train}), and the resulting instance observations $y^{(k)}_{\text{base}}$ are perturbed by an independent GP function $f^{(k)}_{\text{bias}}$ drawn from the training prior, with scale $s_k \sim \mathcal{U}(0.2, 2)$ and mode $m_k \sim \text{Bern}(0.5)$:

$$y_i^{(k)} = \begin{cases} y^{(k)}_{\text{base},i} + s_k \, f^{(k)}_{\text{bias}}(x_i) & \text{if } m_k = 0 \text{ (additive)}, \\ y^{(k)}_{\text{base},i}\big(1 + s_k \tanh f^{(k)}_{\text{bias}}(x_i)\big) & \text{if } m_k = 1 \text{ (multiplicative)}. \end{cases} \tag{29}$$

Irrelevant regime. An irrelevant instance is an independent GP draw whose hyperparameters are drawn from the training prior $p(\theta)$,

$$\theta^{(k)} \sim p(\theta), \qquad y_i^{(k)} \sim p(y \mid \theta^{(k)}, x_i). \tag{30}$$

On the synthetic benchmark this is implemented as a batch permutation of the target hyperparameters.

Reliability label. Because the task is non-parametric, the parameter-distance label of Equation (7) does not apply. For the token $\mathbf{e}_j = \mathbf{r}_k$ representing instance k, we instead supervise the filter with the regime indicator,

$$c_j = \begin{cases} 1 & \text{if instance } k \text{ is drawn from the relevant regime (Equations (28) and (29))}, \\ 0 & \text{if it is drawn from the irrelevant regime (Equation (30))}. \end{cases} \tag{31}$$

Baseline implementations. The three baselines on this task ingest previous-instance observations without the token interface, and none of them uses the reliability labels.

ALINE*. At deployment, the ALINE backbone receives all $K \cdot T$ previous-instance observations as additional context tokens, concatenated with the experimental observations. This is the same concatenation protocol used for ALINE* on the CES task (Section 4.3).

GP*. A Gaussian process surrogate conditioned on the experimental observations and all $K \cdot T$ previous-instance observations together, treating every point as an evaluation of the same function. The GP uses an RBF kernel with a noise term, is refit by marginal-likelihood maximization at each step, and selects queries by uncertainty sampling. This is the variant reported in Section 4.5.

MT-GP. A multi-task Gaussian process that treats each previous instance as a separate task correlated with the target. The target is task 0 and each auxiliary previous instance is task $k \in \{1, \ldots, K\}$. We use the rank-Q intrinsic coregionalization model of Bonilla et al. [50] with a zero prior mean,

$$k\big((x_i, t_i), (x_j, t_j)\big) = \big[WW^\top + \text{diag}(\boldsymbol{v})\big]_{t_i, t_j} \cdot \sigma_f^2 \, \exp\!\Big(-\tfrac{1}{2} \, \|x_i - x_j\|^2 / \ell^2\Big), \tag{32}$$

where the tasks share an RBF kernel with lengthscale ℓ and amplitude σ_f^2, while W and $\boldsymbol{v}$ determine the correlations and task-specific variation across instances. We use rank $Q = 1$, so tasks share a single latent component together with task-specific residual variation. Each task additionally has its own learned observation-noise variance. Hyperparameters are optimized by maximum marginal likelihood using Adam with learning rate 0.05, with 200 optimization steps for the initial fit and 60 steps for subsequent fits initialized from the preceding solution. At each acquisition step, the GP is fit to the currently observed target points together with all previous-instance observations, and the next target point is selected by maximizing the predictive standard deviation for task 0.

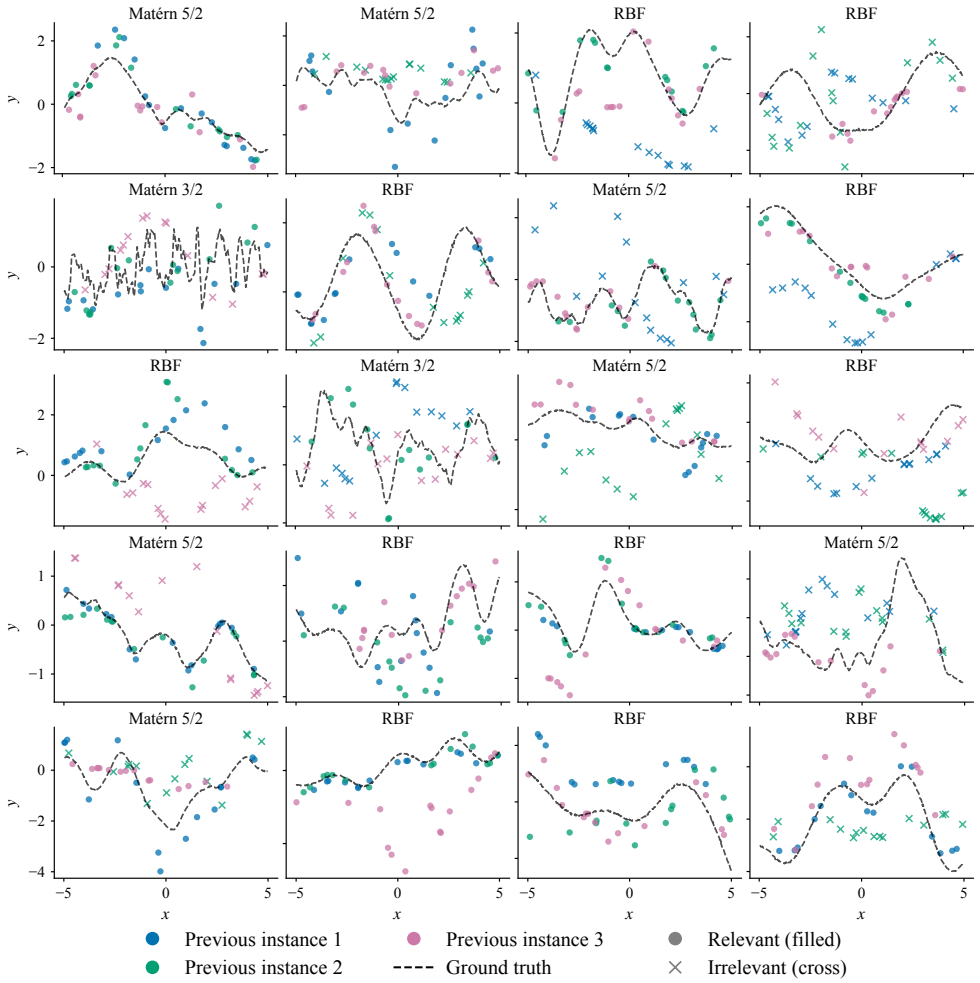


Figure 12: Examples of previous instances supplied during training. Each panel shows one target function (black dashed line) and up to three instances $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ (colored points), with filled circles for the relevant regime and crosses for the irrelevant regime.

Acquisition functions. Random sampling selects the next query from the candidate pool randomly. Uncertainty sampling selects the next query at the point of maximum predictive standard deviation,

$$\mathrm{US}(x) = \sqrt{\mathbb{V}[y \mid x, \mathcal{D}]}. \tag{33}$$

Variance reduction selects the query that maximizes the expected reduction in posterior variance over a test set $\{x_m^\star\}_{m=1}^M$ [51],

$$\mathrm{VR}(x) = \frac{\sum_{m=1}^M \left(\mathrm{Cov}_{\mathrm{post}}(x_m^\star, x)\right)^2}{\mathbb{V}[y \mid x, \mathcal{D}]}, \tag{34}$$

with the posterior covariance computed in closed form from the GP,

$$\mathrm{Cov}_{\mathrm{post}}(x^\star, x) = k(x^\star, x) - k(x^\star, X_{\mathrm{train}})\,(K_{\mathrm{train}} + \sigma_{\mathrm{GP}}^2 I)^{-1}\,k(X_{\mathrm{train}}, x), \tag{35}$$

where k is the GP kernel, $K_{\mathrm{train}} = k(X_{\mathrm{train}}, X_{\mathrm{train}})$, and σ_{GP}^2 is the noise variance.

D.4 Psychometric model

Model description. In the psychometric task [3], the experimenter infers the parameters of a participant's psychometric function from binary yes/no responses to stimulus intensities $x \in \mathbb{R}$. The

response probability is

$$\pi(x;\boldsymbol{\theta}) = \theta_{\text{guess}} + (1 - \theta_{\text{guess}} - \theta_{\text{lapse}})\, F\left(\frac{x - \theta_{\text{threshold}}}{\theta_{\text{slope}}}\right), \qquad F(z) = 1 - \exp(-10^z), \tag{36}$$

where $\theta_{\text{threshold}}$ is the threshold, θ_{slope} the slope, θ_{guess} the guess rate, and θ_{lapse} the lapse rate. The priors are

$$\theta_{\text{threshold}} \sim \mathcal{U}(-3, 3), \qquad \theta_{\text{slope}} \sim \mathcal{U}(0.1, 2), \tag{37}$$

$$\theta_{\text{guess}} \sim \mathcal{U}(0, 0.15), \qquad \theta_{\text{lapse}} \sim \mathcal{U}(0, 0.06), \tag{38}$$

where the narrow supports on θ_{guess} and θ_{lapse} reflect their physiological interpretation. The design space is $[-5, 5]$ and the response is $y \sim \text{Bernoulli}(\pi(x;\boldsymbol{\theta}))$. The metric is the joint RMSE on $(\theta_{\text{threshold}}, \theta_{\text{slope}})$.

Baselines. We compare IMBUE against three baselines.

QUEST+ [3] selects each stimulus to maximize the expected information gain about the full parameter vector on a discrete grid of candidate values. We discretize the parameter space at the resolution recommended by Watson [3] and run inference over the joint posterior on $(\theta_{\text{threshold}}, \theta_{\text{slope}}, \theta_{\text{guess}}, \theta_{\text{lapse}})$, reporting RMSE on $(\theta_{\text{threshold}}, \theta_{\text{slope}})$.

Psi-marginal [46] extends the original psi method [52] by marginalizing over nuisance parameters during stimulus selection. Since our metric targets only $(\theta_{\text{threshold}}, \theta_{\text{slope}})$, we configure Psi-marginal to optimize the information gain on these two parameters and treat $(\theta_{\text{guess}}, \theta_{\text{lapse}})$ as nuisance parameters.

ALINE [17] is configured with a parameter target $Z = (\theta_{\text{threshold}}, \theta_{\text{slope}}, \theta_{\text{guess}}, \theta_{\text{lapse}})$ in line with our setting. The architecture and training match the description in Appendix D.2.

E Additional experimental results

E.1 Location Finding

Design trajectory. In Figure 13, most queries lie in or near the regions highlighted by the supplied prior, rather than being spread across the full design space.

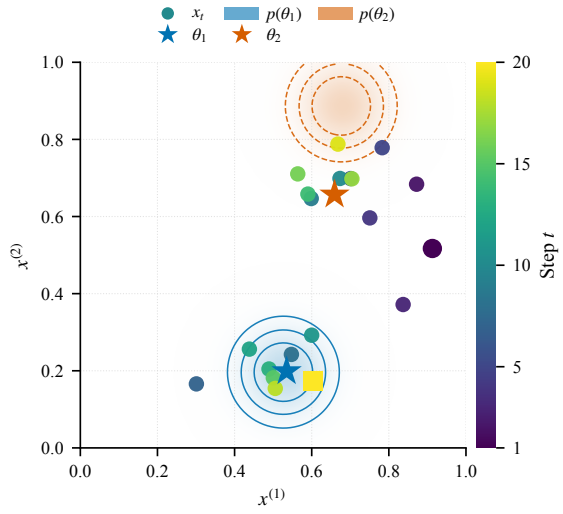


Figure 13: Location Finding trajectory: prior contours, true sources (stars), and queries colored by step; the yellow square marks the final query.

Effect of prior type. The main-text result in Figure 4 shows ALINE outperforming Random, DAD, and RL-BOED, so we restrict the prior-type comparison here to ALINE and IMBUE. Figure 14

reports sPCE under four prior conditions (strong, weak, miscalibrated, and uniform) alongside the baseline (ALINE). IMBUE accelerates early-stage information acquisition under a strong prior and to a smaller extent under a weak prior, and keeps close to the no-external baseline (ALINE) under both miscalibrated and uniform priors.

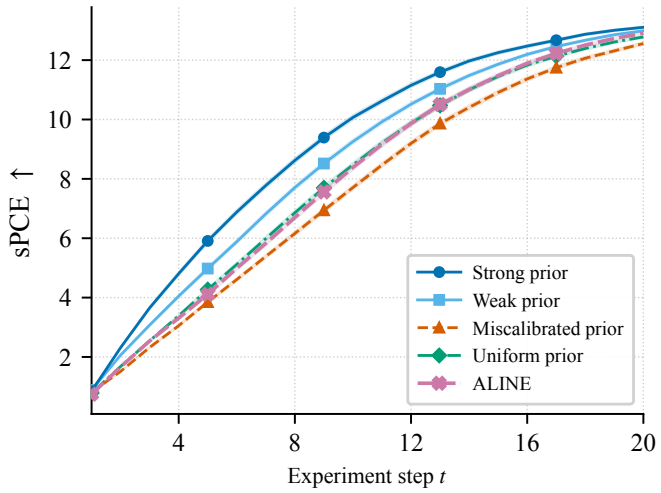


Figure 14: sPCE (↑) on Location Finding: IMBUE under strong, weak, uniform, and miscalibrated priors, and baseline ALINE (mean ± 95% CI over 2,000 rollouts).

Filter outputs over time. Figure 15 traces the per-token reliability filter outputs α_j over $T = 20$ rounds under the four prior conditions. Strong-prior scores settle close to 1, while weak-prior scores stabilize around 0.8. Miscalibrated-prior scores start at an intermediate level and decay toward 0 as observations accumulate, indicating decreasing trust in the provided prior. Uniform-prior scores stay near 0 from the first rounds. These patterns match the supervision signal described in Appendix C.3.

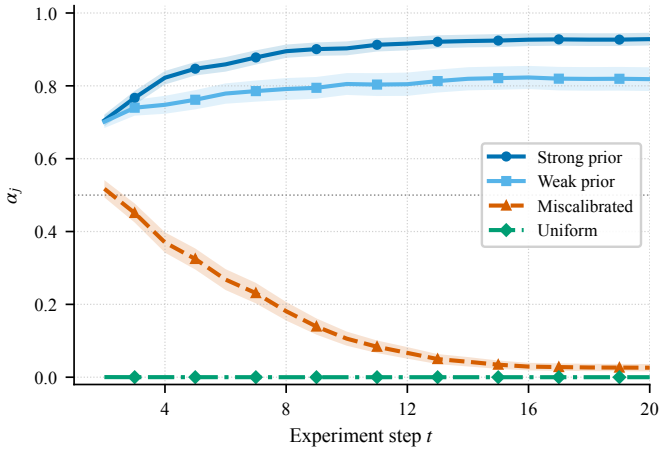


Figure 15: Reliability filter output α_j for the prior tokens on Location Finding, mean ± 95% CI over 100 rollouts, shown as a function of experiment step t. Each curve corresponds to one of the four prior conditions in Figure 14. The dotted line marks $\tau = 0.5$.

E.2 CES

Posterior visualization. Figure 16 shows the marginal posterior distributions estimated by IMBUE on a single CES rollout. The posteriors concentrate around the ground-truth values across all five parameters.

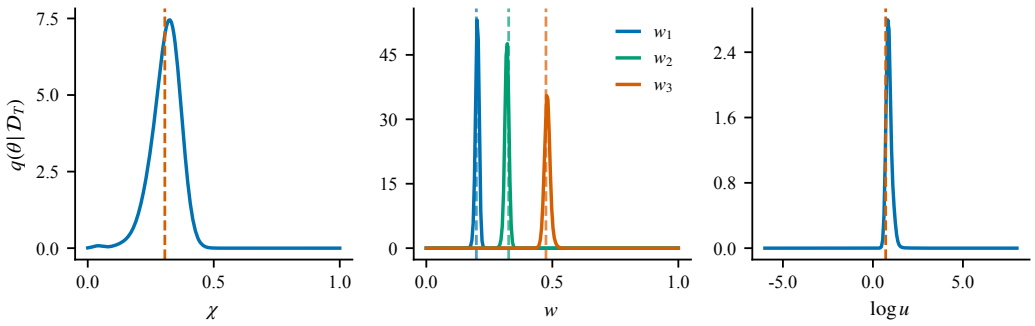


Figure 16: Marginal posterior distributions estimated by IMBUE on the CES task after $T = 10$ steps. Dashed lines mark the true parameter values.

Effect of the number of previous instances K. Figure 17 contrasts the effect of adding relevant and irrelevant previous instances. With relevant instances (left), sPCE improves initially and then levels off, suggesting diminishing returns from additional context. With irrelevant instances (right), sPCE remains close to its $K = 0$ value over the tested range.

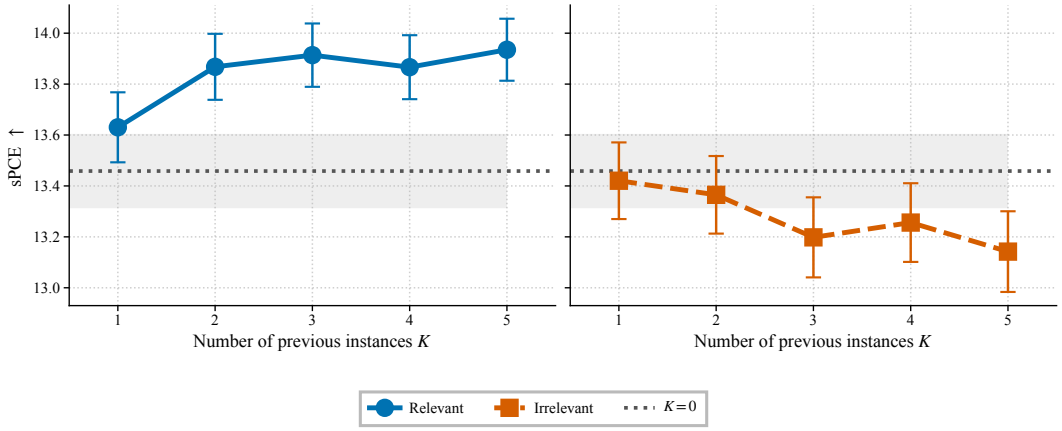


Figure 17: Effect of the number of previous instances K on CES. sPCE (mean $\pm$ 95% CI, 2,000 rollouts).

Effect of training-time exposure to previous instances. Figure 18 extends Figure 3 to the five configurations in Table 1. With auxiliary inputs, ALINE-aux improves on ALINE*, especially under irrelevant input, but remains below IMBUE. This gap supports the combined benefit of the instance-token interface and reliability filter beyond training exposure. Their individual contributions are examined in Table 2. At $K = 0$, the three methods perform similarly.

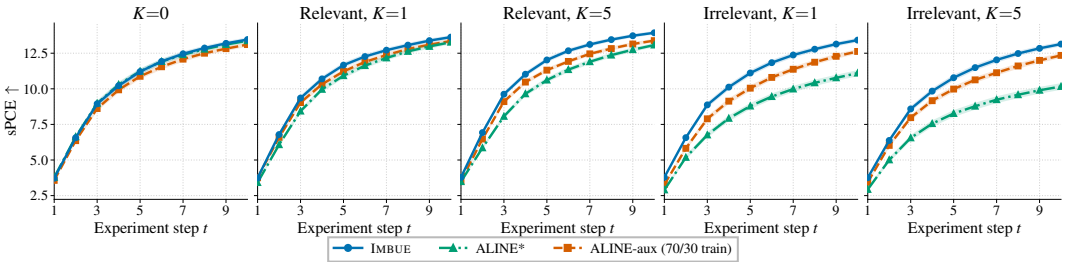


Figure 18: CES with previous-instance observations. Stepwise sPCE (mean $\pm$ 95% CI, 2,000 rollouts) over experiment steps $t = 1, \ldots, 10$. Panels vary the number of previous instances supplied at deployment, $K = 0$, $K \in \{1, 5\}$ relevant, or $K \in \{1, 5\}$ irrelevant.

Number of summary slots. We ablate the number of summary slots n_s in the per-instance encoder g_{aux} on the CES task. Table 8 shows small differences across these settings. Increasing the number of summary slots does not consistently improve performance. We use $n_s = 2$ in the CES experiments and $n_s = 4$ in the GP and Psychometric experiments.

Table 8: Effect of the number of summary slots n_s in the per-instance encoder g_{aux} on CES. sPCE (mean $\pm$ 95% CI, 2,000 rollouts).

	$K = 0$	$K = 1$		$K = 5$	
n_s	—	Rel.	Irrel.	Rel.	Irrel.
1	13.42 ± 0.15	$\mathbf{13.73 \pm 0.13}$	13.37 ± 0.16	13.92 ± 0.12	$\mathbf{13.22 \pm 0.16}$
2 (**IMBUE**)	$\mathbf{13.46 \pm 0.15}$	13.63 ± 0.14	$\mathbf{13.42 \pm 0.15}$	$\mathbf{13.93 \pm 0.12}$	13.14 ± 0.16
4	13.41 ± 0.15	13.58 ± 0.14	13.33 ± 0.16	13.82 ± 0.12	13.09 ± 0.16

E.3 GP active learning

Acquisition trajectory. Figure 19 shows one IMBUE rollout on a synthetic GP function over $T = 15$ steps. As observations accumulate, the predictive mean approaches the ground truth and uncertainty decreases near queried locations.

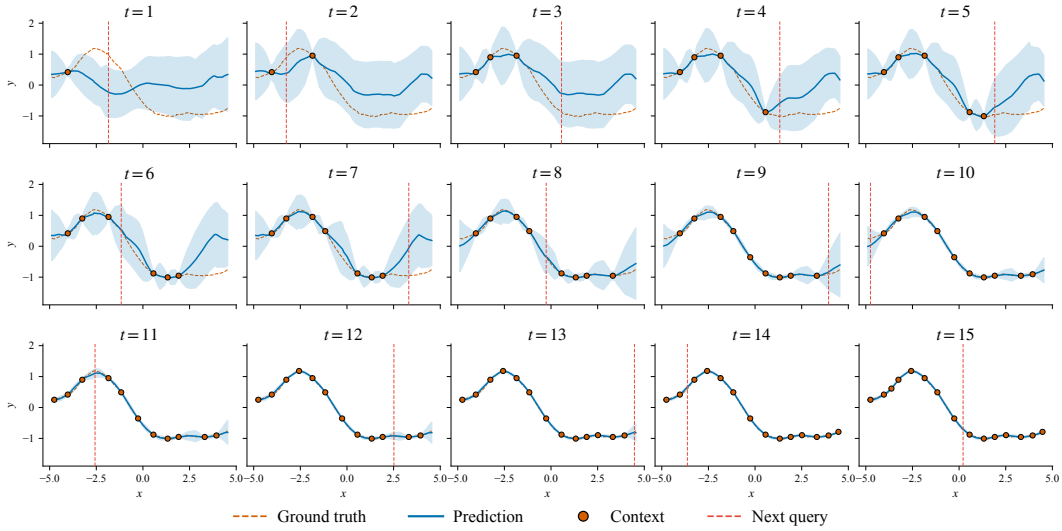


Figure 19: Sequential design strategy of IMBUE on a 1D synthetic GP function over $T = 15$ steps. Each panel shows the ground-truth function (orange dashed), the predictive mean (blue solid) with 95% predictive band (shaded), the accumulated context (orange dots), and the next queried design (red dashed vertical line).

Extended baseline comparison. Figure 20 compares IMBUE with GP and ALINE baselines. GP-RS, GP-US, and GP-VR condition on experimental observations and use random sampling, uncertainty sampling, and variance reduction [51], respectively. GP* additionally conditions on all observations from five relevant previous instances and uses uncertainty sampling. ALINE uses experimental observations alone, while ALINE* concatenates previous-instance observations at deployment. IMBUE uses no previous instances ($K = 0$) or five relevant instances ($K = 5$ rel.).

Without previous instances, IMBUE performs similarly to ALINE and is competitive with the GP baselines. Supplying relevant previous instances reduces its early-step RMSE on all four benchmarks, with smaller differences from the $K = 0$ variant later in the rollout. GP* and ALINE* have lower initial RMSE than their counterparts without previous instances, but improve less as new observations are collected. With relevant previous instances, IMBUE attains lower final RMSE than both baselines.

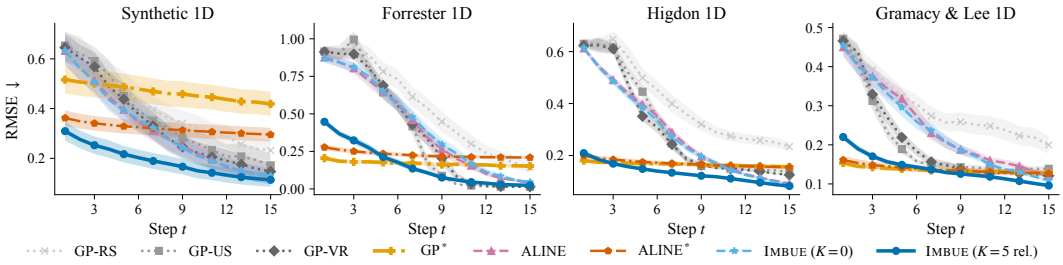


Figure 20: Extended baseline comparison on four 1D GP active learning benchmarks (RMSE ↓, mean and 95% CI over 100 rollouts).

Robustness to irrelevant previous instances. Figure 21 reports RMSE at each experimental step for different proportions of irrelevant previous instances, extending the final-step comparison in the main text (Figure 5). When all five previous instances are relevant, IMBUE starts with lower RMSE than in the $K = 0$ condition on all four benchmarks. When irrelevant instances are included, IMBUE continues to reduce RMSE as experimental observations accumulate, reaching final RMSE close to its $K = 0$ performance across the tested compositions. In comparison, GP* and ALINE* show smaller reductions in RMSE over the rollout, and their final RMSE increases with the proportion of irrelevant instances.

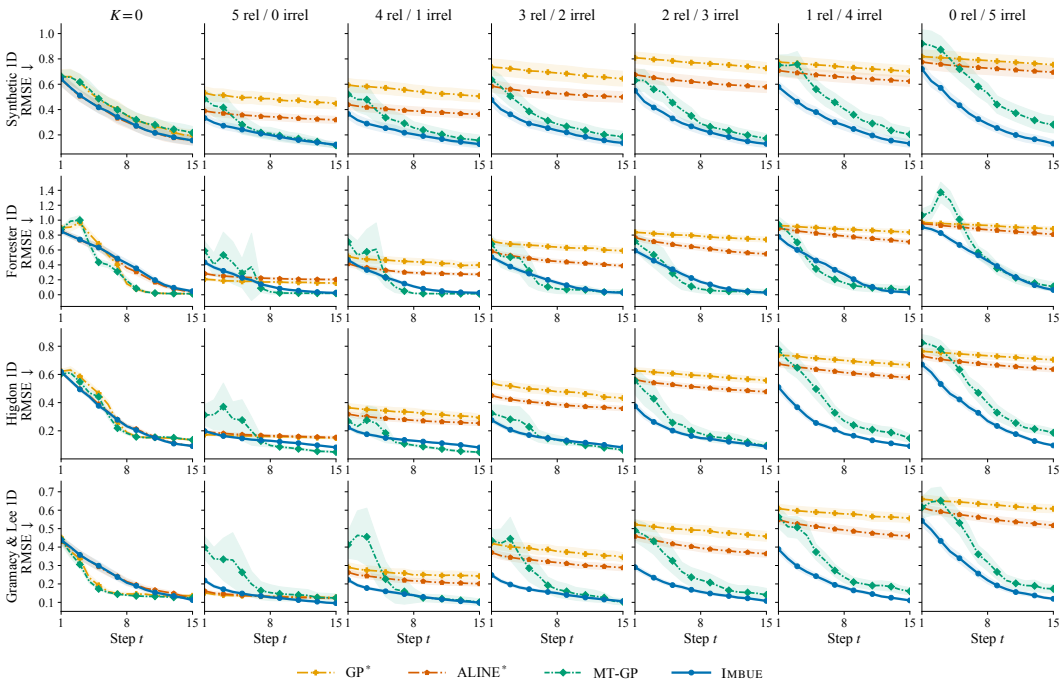


Figure 21: RMSE at each experimental step on four 1D GP benchmarks (↓, mean and 95% CI over 100 rollouts). Rows index the benchmark, and columns show the composition of the $K = 5$ previous instances as relevant/irrelevant (the leftmost column is the $K = 0$ baseline).

Inference time. Table 9 reports total inference time for rollouts of 15 steps on the four GP benchmarks. At $K = 0$, IMBUE has inference time comparable to ALINE. With five relevant previous instances, IMBUE remains substantially faster than the GP variants in the table, which refit their models at each step. Relative to ALINE, IMBUE incurs a small absolute increase in inference time for instance encoding and reliability filtering.

Table 9: Total inference time for rollouts of 15 steps on four GP benchmarks (mean $\pm$ standard deviation in seconds, 10 runs).

Method	Synthetic 1D	Forrester	Higdon	Gramacy & Lee
GP-RS	0.332 ± 0.020	0.343 ± 0.029	0.305 ± 0.024	0.300 ± 0.022
GP-US	0.408 ± 0.024	0.381 ± 0.020	0.397 ± 0.023	0.399 ± 0.027
GP-VR	1.236 ± 0.037	1.242 ± 0.020	1.240 ± 0.027	1.216 ± 0.033
GP* ($K = 5$ rel.)	0.556 ± 0.068	0.640 ± 0.036	0.648 ± 0.050	0.649 ± 0.087
ALINE	0.047 ± 0.000	0.049 ± 0.003	0.049 ± 0.003	0.048 ± 0.003
IMBUE ($K = 0$)	0.049 ± 0.004	0.047 ± 0.000	0.047 ± 0.000	0.047 ± 0.000
IMBUE ($K = 5$ **rel.**)	0.074 ± 0.000	0.075 ± 0.005	0.075 ± 0.004	0.075 ± 0.005

E.4 Psychometric model

Figure 22 extends the main-text comparison (Figure 6) to different numbers of relevant or irrelevant previous instances. Without previous instances ($K = 0$), IMBUE performs similarly to ALINE, QUEST+, and Psi-marginal. Relevant previous instances accelerate the reduction in IMBUE's RMSE, with greater benefits when more instances are supplied. ALINE*, however, does not improve over its $K = 0$ baseline even with relevant previous instances. With irrelevant previous instances, IMBUE's RMSE remains close to that obtained with $K = 0$ throughout the rollout, whereas the RMSE of ALINE* increases with K.

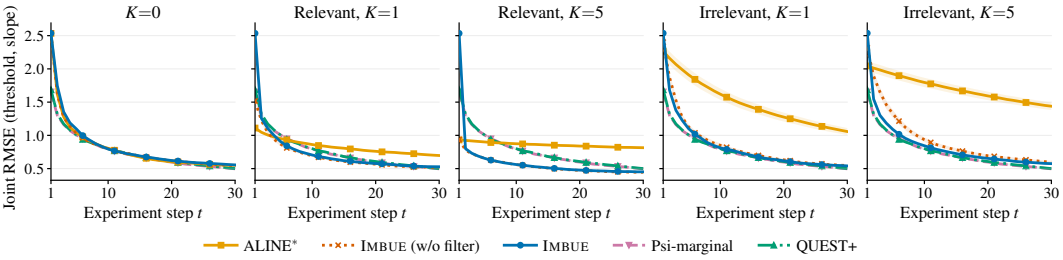


Figure 22: Psychometric model: joint RMSE on $(\theta_{\text{threshold}}, \theta_{\text{slope}})$ ($\downarrow$) over $T = 30$ steps (mean $\pm$ 95% CI, 500 rollouts). Panels compare no instances ($K = 0$) with $K \in \{1, 5\}$ relevant or irrelevant instances. QUEST+ and Psi-marginal use no previous instances; their curves are repeated across panels for reference. The IMBUE (w/o filter) curves are discussed in Appendix E.5.

E.5 Ablations and sensitivity studies

We examine the reliability filter across input channels, compare binary and soft gating, and assess sensitivity to the training mixture and filter loss weight.

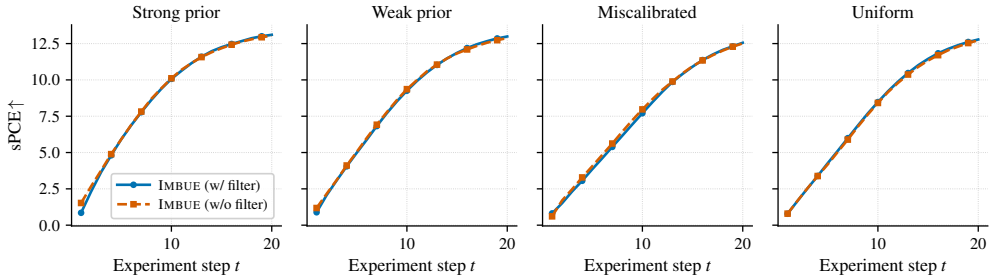
Filter behavior across channels and reliability regimes. Figure 23 compares IMBUE with and without the reliability filter on Location Finding (prior input) and CES (previous-instance input), with corresponding psychometric curves shown in Figure 22 as IMBUE (w/o filter).

Prior channel. On Location Finding (Figure 23a), the two variants are essentially indistinguishable across all four prior conditions, including the miscalibrated one.

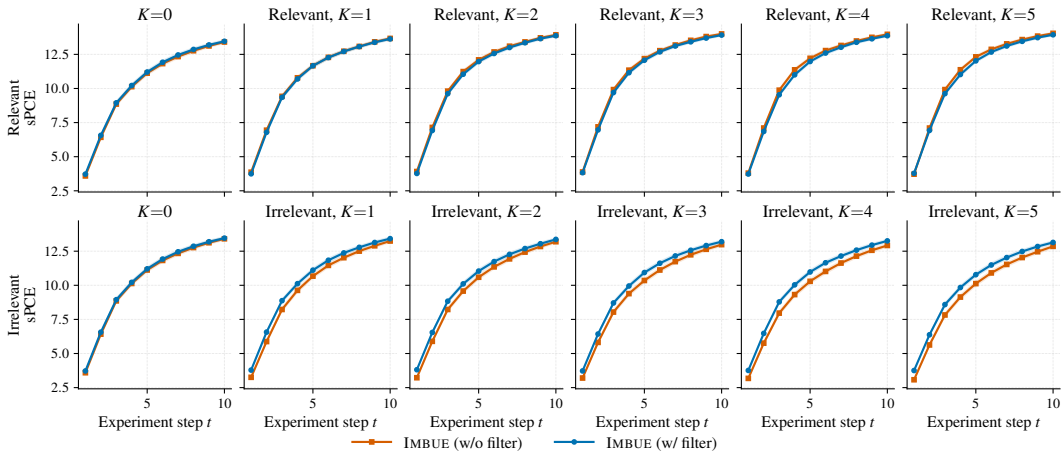
Previous-instance channel under reliable input. On CES with K relevant instances (top row of Figure 23b) and on the psychometric task with $K \in \{1, 5\}$ relevant instances (relevant panels of Figure 22), the two variants are again indistinguishable. When all instance tokens carry useful signal, removing the filter neither helps nor hurts.

Previous-instance channel under unreliable input. On CES with irrelevant previous instances (bottom row of Figure 23b), the full IMBUE achieves higher sPCE than IMBUE w/o filter, with a clearer advantage when more irrelevant instances are supplied. The gap is most visible during the middle of the rollout and narrows as experimental observations accumulate. On the psychometric task (Figure 22), the full IMBUE maintains lower joint RMSE, with a larger and more persistent gap when

more irrelevant instances are supplied. The filter can therefore provide additional protection against negative transfer.



(a) Location Finding under four prior conditions.



(b) CES under $K \in \{0, 1, \ldots, 5\}$ previous instances in both reliability regimes.

Figure 23: Filter ablation on Location Finding and CES. sPCE mean $\pm$ 95% CI over 2,000 rollouts for the full IMBUE (w/ filter) and the variant retrained without reliability filter (w/o filter).

Binary versus soft gating. Figure 24 tests whether replacing binary gating with soft gating improves sPCE on CES under the five previous-instance configurations of Table 1. With binary gating, each in-context token is kept or excluded by thresholding $\alpha_j^{(t)}$ at $\tau = 0.5$ (Section 3.3). Soft gating instead scales each token's contribution to self-attention by its reliability score $\alpha_j^{(t)}$. The two variants achieve similar sPCE with relevant and irrelevant previous instances, indicating no clear improvement from soft gating in these experiments.

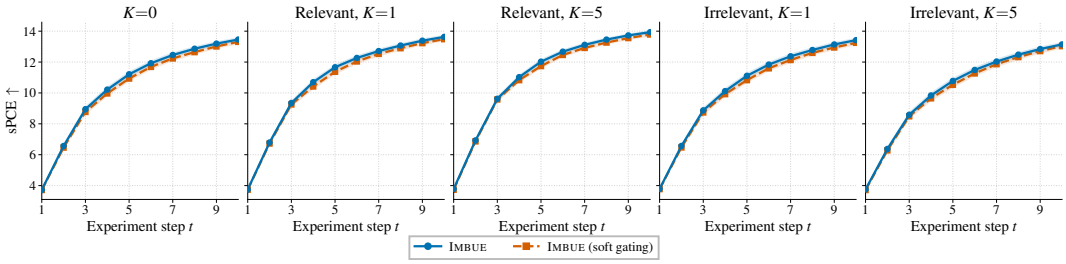


Figure 24: Binary versus soft gating on CES. sPCE (mean $\pm$ 95% CI over 2,000 rollouts, $T = 10$ steps) under the five previous-instance configurations of Table 1.

Sensitivity to training mixture and filter loss weight. Figure 25 tests sensitivity to the default training settings on Location Finding. We vary p_{good}, the probability that a prior marginal is calibrated during training (Appendix C.3), and λ_3, the weight on the filter loss $\mathcal{L}_{\text{filter}}$ in Equation (9).

Increasing p_{good} improves sPCE under strong priors but reduces it under miscalibrated priors, while weak and uniform priors are much less affected. The training mixture therefore controls a trade-off between benefiting from informative priors and robustness to misleading ones. The default $p_{\text{good}} = 0.7$ provides a compromise within the tested range.

The default $\lambda_3 = 1$ and the smaller weight $\lambda_3 = 0.1$ give nearly identical final sPCE under all four prior conditions, so the Location Finding results are insensitive to λ_3 within the tested range.

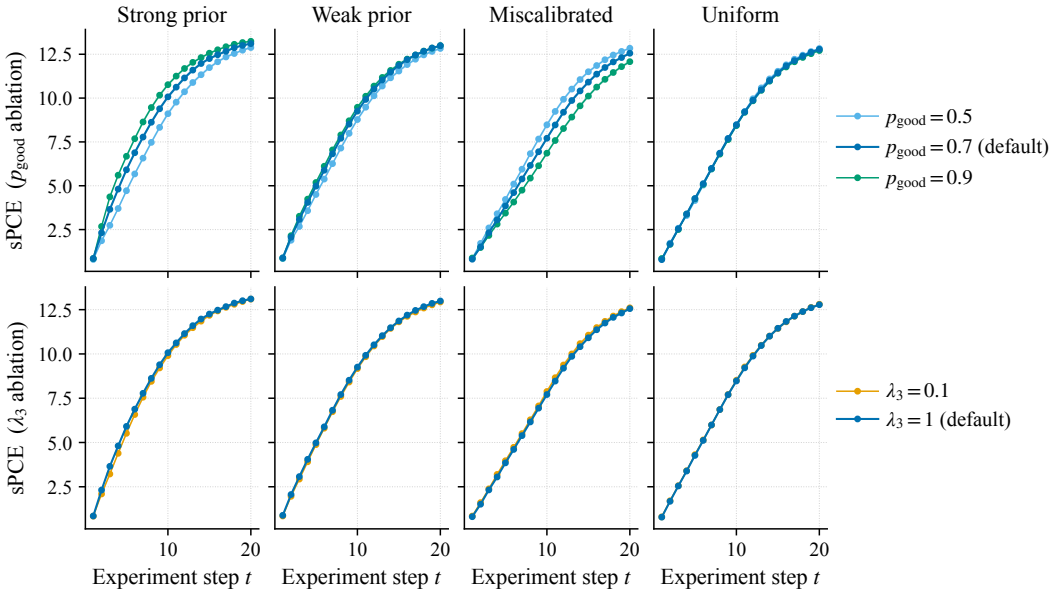


Figure 25: Sensitivity sweeps on Location Finding, sPCE (mean ± 95% CI over 2,000 rollouts). **Top row**, varying the training mixture parameter $p_{\text{good}} \in \{0.5, 0.7, 0.9\}$, with 0.7 as the default used in the main results. **Bottom row**, varying the filter loss weight $\lambda_3 \in \{0.1, 1\}$, with 1 as the default.

E.6 Deployment shifts absent from training

We evaluate the trained IMBUE models under three deployment shifts not included in the training mixture (Appendix C.3, Appendix C.4). No retraining is performed, and the remaining evaluation settings follow the corresponding CES and Location Finding experiments.

Biased but partially relevant instances (CES). All $K = 5$ instances are generated as in the relevant regime, but share a systematic shift of $0.3 \times (r_l^{\max} - r_l^{\min})$ on a randomly selected subset of the parameters. Each instance is therefore informative about the unshifted parameters and misleading about the shifted ones. IMBUE stays close to its $K = 0$ performance, while deployment-time concatenation degrades (Table 2). Biased instances receive mean reliability scores between those of relevant and irrelevant instances (Table 10). This is expected, since an instance that is right about some parameters produces weaker evidence against itself. The filter thus places a regime it has never seen between the two regimes it was trained on rather than forcing it into either.

Table 10: Mean reliability score α over instance tokens on CES with $K = 5$ instances (mean ± 95% CI, 2,000 rollouts).

Step t	Relevant	Biased (shift)	Irrelevant
2	0.818 ± 0.004	0.677 ± 0.012	0.479 ± 0.010
10	0.906 ± 0.002	0.621 ± 0.016	0.246 ± 0.008

A different noise model in the previous instances (CES). The instance parameters are generated as in the relevant regime, but the instance outcomes carry heavier-tailed Student-t noise at a larger scale in place of the Gaussian noise used during training. The live experiment is unchanged and uses the training likelihood. Under the oracle definition of Equation (7), relevant instances remain reliable because their parameters stay close to $\theta^\star$. Table 11 shows that relevant instances remain beneficial, while irrelevant instances leave performance close to the $K = 0$ condition.

Table 11: CES with $K = 5$ instances whose outcomes carry Student-t noise ($\nu = 3$, scale $3\sigma_{\mathrm{CES}}$) instead of the Gaussian noise used in training. sPCE at $T = 10$ (mean $\pm$ 95% CI, 2,000 rollouts).

Condition	sPCE
$K = 0$	13.46 ± 0.15
$K = 5$ relevant, Gaussian noise (in-dist.)	13.93 ± 0.12
$K = 5$ relevant, Student-t noise (shift)	14.00 ± 0.12
$K = 5$ irrelevant, Gaussian noise (in-dist.)	13.14 ± 0.16
$K = 5$ irrelevant, Student-t noise (shift)	13.32 ± 0.16

Priors with incorrect support (Location Finding). We test a more severe form of prior misspecification in which the supplied prior assigns exactly zero probability to the bin containing the true parameter. We first draw $\theta^\star \sim \mathcal{U}[0, 1]^4$, and only then construct the prior histogram so that it excludes the corresponding bin. We consider two variants: a *truncated* prior that concentrates its mass in a window away from $\theta^\star$, and a *boundary* prior that places its mass at the two ends of the support while $\theta^\star$ lies in the interior. This differs fundamentally from training, where $\theta^\star$ is sampled from the supplied histogram and therefore always lies within its support. We observe no sharp degradation in sPCE under either form of prior misspecification (Table 12).

Table 12: Location Finding under priors that assign zero mass to the bin containing $\theta^\star$. sPCE (mean $\pm$ 95% CI, 2,000 rollouts) at four experiment steps, alongside the three in-distribution prior types.

Prior	$t = 2$	$t = 5$	$t = 11$	$t = 20$
Strong (in-dist.)	2.26 ± 0.07	5.77 ± 0.11	10.49 ± 0.11	13.03 ± 0.06
Uniform (in-dist.)	1.72 ± 0.06	4.25 ± 0.09	9.10 ± 0.12	12.83 ± 0.07
Miscalibrated (in-dist.)	1.56 ± 0.05	3.90 ± 0.08	8.43 ± 0.11	12.61 ± 0.07
Truncated (shift)	1.46 ± 0.05	3.89 ± 0.08	8.54 ± 0.11	12.56 ± 0.07
Boundary (shift)	1.76 ± 0.06	4.27 ± 0.09	9.12 ± 0.11	12.77 ± 0.06

Taken together, these experiments probe three distinct deployment shifts absent from training: biased but partially relevant previous instances, a different noise model in the previous instances, and priors assigning zero mass to the truth. IMBUE does not degrade sharply in any of them. They are three specific, plausible shifts and do not establish robustness to arbitrary deployment mismatch.

E.7 Sensitivity to the candidate design pool

We keep the trained policies fixed and vary the candidate pool size at deployment from $N_q^{\text{eval}} = 200$ to 8,000, using i.i.d. uniform or Sobol pools on CES (Table 13) and Location Finding (Table 14). The main-text setting is $N_q^{\text{eval}} = 2{,}000$ with uniform pools.

Tables 13 and 14 show that the main qualitative conclusions hold across the tested pool sizes and constructions: relevant instances remain beneficial, irrelevant instances leave performance close to the $K = 0$ condition, and the ordering of the prior conditions is preserved. Gains from enlarging the pool diminish beyond the default size. Larger pools give denser coverage of the design space and are likely to become more useful as the design dimension grows. In very high-dimensional design spaces, enumeration is not realistic, but since the design head produces a differentiable pre-normalization score for each candidate, this score could in principle be optimized directly with respect to the design in continuous space.

Table 13: CES. sPCE at $T = 10$ as a function of the candidate design pool N_q^{eval} and construction, with the trained policy fixed (2,000 rollouts; 95% CIs are approximately ± 0.14). The main-text setting is $N_q^{\text{eval}} = 2{,}000$ with uniform pools.

Condition	Pool	$N_q^{\text{eval}} = 200$	500	1,000	2,000	4,000	8,000
$K = 0$	Uniform	13.03	13.23	13.40	13.46	13.54	13.58
	Sobol	13.15	13.35	13.48	13.58	13.69	13.70
$K = 1$ rel.	Uniform	13.25	13.48	13.61	13.63	13.75	13.81
	Sobol	13.47	13.62	13.74	13.84	13.90	13.96
$K = 5$ rel.	Uniform	13.60	13.79	13.90	13.93	13.98	14.02
	Sobol	13.59	13.73	13.90	13.93	13.98	14.00
$K = 1$ irrel.	Uniform	12.92	13.19	13.33	13.42	13.52	13.55
	Sobol	12.83	13.10	13.26	13.34	13.45	13.48
$K = 5$ irrel.	Uniform	12.83	13.07	13.20	13.14	13.34	13.37
	Sobol	12.90	13.13	13.24	13.37	13.47	13.54

Table 14: Location Finding. sPCE at $T = 20$ as a function of the candidate design pool N_q^{eval} and construction, with the trained policy fixed (2,000 rollouts; 95% CIs are approximately ± 0.06). The main-text setting is $N_q^{\text{eval}} = 2{,}000$ with uniform pools.

Prior	Pool	$N_q^{\text{eval}} = 200$	500	1,000	2,000	4,000	8,000
Strong	Uniform	11.91	12.81	13.05	13.03	13.13	13.07
	Sobol	12.19	12.93	13.12	13.13	13.10	13.09
Weak	Uniform	11.88	12.64	12.84	12.89	12.99	12.94
	Sobol	12.07	12.81	12.95	12.94	12.98	12.95
Uniform	Uniform	11.79	12.50	12.69	12.83	12.80	12.79
	Sobol	11.99	12.64	12.71	12.80	12.78	12.79
Miscalibrated	Uniform	11.67	12.31	12.49	12.61	12.55	12.55
	Sobol	11.82	12.44	12.55	12.65	12.61	12.57

E.8 Computational overhead of in-context conditioning

Table 9 compares IMBUE with GP baselines on the active learning task. Here we break the overhead of in-context conditioning down by module on CES and Location Finding, separating the in-context token pathway from the reliability filter.

Parameters. With a fixed filter architecture and embedding dimension ($D = 32$), f_ω (Equation (5)) has 9,089 parameters, independent of K, the number of histogram bins N_B, and the task. In our experimental configurations, this amounts to 6.3% of the full CES model (145,408 parameters) and 7.3% of the full Location Finding model (123,744 parameters). Beyond the filter, the additional conditioning parameters are primarily in the per-instance encoder g_{aux} for CES (38,112 parameters, plus 64 summary-token parameters), and in the histogram prior embedder for Location Finding (17,056 parameters at $N_B = 100$).

Per-step latency. Table 15 shows that on CES, encoding previous instances increases inference time, and enabling the reliability filter adds a further cost. On Location Finding, the comparison isolates the additional time required to filter prior tokens. For both tasks, the full model keeps per-step inference time in the millisecond range. The $N_q^{\text{eval}} = 2{,}000$ candidate tokens dominate the sequence, so shortening the in-context segment from $K \cdot T = 50$ concatenated observations to $K = 5$ instance tokens brings little latency benefit at this scale. The advantage of the token interface appears at larger K and T, since the in-context segment of ALINE* grows as $K \cdot T$ whereas that of IMBUE grows as K, so concatenation eventually exceeds the sequence-length budget.

Accuracy–efficiency trade-off. On CES, the instance-token interface improves sPCE over ALINE-aux under relevant, biased, and irrelevant inputs. Adding the reliability filter further improves mean sPCE under biased and irrelevant inputs while maintaining comparable performance under relevant input.

Table 15: Per-step inference latency of IMBUE by module (ms per step, mean ± standard deviation over 10 runs; NVIDIA H100, batch size 1, $N_q^{\text{eval}} = 2{,}000$). On CES the in-context input is $K = 5$ instances; on Location Finding it is the prior histogram.

Configuration	CES	Location Finding
No in-context tokens	2.55 ± 0.003	—
In-context tokens, w/o filter	3.62 ± 0.003	2.56 ± 0.002
In-context tokens + filter (full)	4.57 ± 0.005	3.28 ± 0.003

On Location Finding, the filter has little effect on sPCE under the tested prior conditions, but provides interpretable assessments of prior reliability.

F Computational resources and software

All experiments, including model development, hyperparameter tuning, baseline runs, and preliminary analyses, were conducted on NVIDIA H100 80GB GPUs. The total compute consumed across all stages is approximately 4,000 GPU hours. Training an IMBUE model for 10^5 epochs takes roughly 7 hours per benchmark. The core code base is implemented in PyTorch [53] (`https://pytorch.org/`, License: modified BSD license). The GP baselines use scikit-learn [54] (`https://scikit-learn.org/`, License: modified BSD license). For DAD [13] and RL-BOED [15], we use the PyTorch implementations at `https://github.com/e-wxy/DAD` (License: MIT) and `https://github.com/huangdaolang/RL-BOED`, respectively. MT-GP [50] is our own PyTorch implementation described in Appendix D.3. For ALINE [17], we use the official implementation (`https://github.com/huangdaolang/ALINE`, License: MIT), on which our code base also builds. We use the `questplus` package (`https://github.com/hoechenberger/questplus`, License: GPL-3.0) for QUEST+ and `Psi-staircase` (`https://github.com/NNiehof/Psi-staircase`, License: GPL-3.0) for Psi-marginal.

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: In this paper, we introduce IMBUE, an amortized BED framework that admits both forms of external knowledge at deployment within a shared Transformer architecture (Section 3). We explicitly state IMBUE's key contributions, such as a unified token interface to integrate external knowledge and a reliability filter that further reduces negative transfer. The experimental results presented in Section 4 across various tasks (Location Finding, CES, active learning, psychometrics) match and validate the claims made about IMBUE's performance and utility.

   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 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Complete proofs for the theoretical results are provided in Appendix B.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Details for all our experiments and the architecture used in our paper are available in Appendices C and D, 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: Code is publicly available at https://github.com/DralnMatthew/IMBUE under the MIT license.

   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 D.

   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 in Appendix F.

   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 F.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The only new asset is our research code, released at `https://github.com/DralnMatthew/IMBUE` under the MIT license; the repository documents the dependencies and the commands to train and evaluate models.

    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.