BEYOND TERMINAL OBJECTIVES: BAYESIAN EXPERIMENTAL DESIGN WITH MULTISTEP ACTIONS & LOSSES

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing loss-driven Bayesian experimental design (BED) methods learn policies that target a terminal loss evaluated at a fixed end point. We argue that this terminal formulation is ill-suited to cases where we can take intermediary actions or our experiment endpoint is unknown. To address this, we introduce MA-BED (Multistep Action-BED), which extends the loss-driven BED formulation to include intermediary actions and losses throughout the acquisition trajectory. This assignment of value to incomplete trajectories allows us to encourage the data to be as informative as possible at every stage of the experiment, such that we are encouraged to learn quickly. In turn, this allows us to support problems where separate actions must be taken during acquisition, and problems with unknown, stochastic, or adaptively chosen experiment endpoints. The intermediate signal additionally shortens credit assignment, stabilizing optimization over long rollouts. Indeed, because of the practical training benefits this provides, we find MA-BED not only improves intermediate decisions compared with terminal-loss baselines on target tracking, adaptive stopping, and classical design problems, it can also improve the terminal decisions for longer rollouts.

1 INTRODUCTION

Collecting data requires committing to design decisions before knowing what the resulting experiments will reveal. Bayesian experimental design (BED) targets designs that maximize the value of the information they are expected to provide for a downstream inferential, prediction, or decision problem (Chaloner and Verdinelli, 1995; Huan et al., 2024; Lindley, 1956; Rainforth et al., 2024). In sequential settings, this choice must itself adapt to what has already been observed, and recent policy-based approaches therefore learn entire design strategies that enable fast, adaptive, and non-myopic decision making at deployment (Foster et al., 2021; Huan and Marzouk, 2016; Ivanova et al., 2021). In parallel, recent developments have begun to move BED beyond generic information-acquisition criteria, allowing designs to be optimized directly for the downstream actions and loss that ultimately matter (Filstroff et al., 2021; Go et al., 2026; Huang et al., 2024; 2026b; Rossa et al., 2026).

Across this otherwise diverse literature, one structural assumption is almost invariably shared: the loss is evaluated at a single point, with actions only taken at the end of a trajectory whose length is fixed in advance. We highlight that many sequential decision problems are not naturally expressed in this form. In some, actions and losses accrue throughout. For example, in tracking, the goal is to maintain an estimate of an object's state as noisy observations arrive, and every intermediate estimate carries its own loss (Hurtado et al., 2008; Kreucher et al., 2007). In others, the endpoint of the experiment is not known when the policy is trained, because costs or data availability are uncertain, or we adaptively stop once sufficient evidence has accrued. Such settings arise, for example, in adaptive clinical trials and biostatistics (Pallmann et al., 2018), environmental and epidemiological monitoring (Case et al., 2024; Kang et al., 2016) and active sensing and robotics (Hero and Cochran, 2011; Jones et al., 2013).

Terminal evaluation also makes policies harder to learn: sparse, delayed reward gives long credit-assignment paths and weak gradient signal, a difficulty that grows with the horizon and which is well documented in reinforcement learning (RL) (Andrychowicz et al., 2017; Ng et al., 1999). Consequently, existing policy-based BED methods have rarely been successfully applied to problems with more than a few dozen experiment steps (Rossa et al., 2026), even when using telescoping tricks and RL techniques to aid training (Blau et al., 2022; Huang et al., 2024; 2026a; 2025; Iqbal et al., 2024).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address these limitations, we introduce MA-BED (Multistep Action-BED), which extends the loss-driven BED formulation to include intermediary actions and losses throughout the acquisition trajectory, thereby explicitly assigning value to intermediate decisions. Specifically, rather than using only a single loss at a fixed trajectory endpoint, MA-BED pairs every acquisition step with a separate action and evaluates a corresponding intermediary loss, with the overall loss then formed by the accumulation of the per-step losses. Building on the terminal loss-driven framework Action-BED (Rossa et al., 2026), MA-BED then formulates a joint policy for designs and *all* actions that maps from previous data to the next design decision and the intermediary action to be taken at that step. This then allows the MA-BED policy to be trained using fully end-to-end gradients of the expected overall loss, while providing supervision to this policy at each step of the rollout.

The upshot of this setup is first that MA-BED provides a natural framework for expressing experimental design problems with intermediary actions, such as tracking problems. Second, it allows us to work with dynamic endpoints and learn policies that provide anytime learning. Finally, it can improve the training signal for policy training even in cases where we only care about the terminal reward, allowing better policies to be learned in practice. In particular, we show theoretically that for any given terminal-only problem, there always exists a set of intermediate losses whose resulting multistep objective has the same Bayes-optimal policies as the original terminal-only problem, while allowing design decisions to be made myopically without backpropagating gradients through the trajectory.

We empirically confirm these benefits across three problems: target tracking, preference elicitation, and location-finding. In each case we find that, not only does MA-BED provide improved performance for intermediary actions and early stopping, but it also often improves performance on the terminal loss for long-horizon experiments without any notable increase in training costs.

2 Background

Bayesian experimental design. BED (Lindley, 1956; Rainforth et al., 2024) provides a principled framework for selecting experiments to learn about an unknown quantity of interest or to inform some other downstream task. Let $\theta \sim p(\theta)$ denote latent variables of interest and let $\xi \in \Xi$ denote an experimental design. Given θ and ξ, an observation $y \in \mathcal{Y}$ is generated according to a probabilistic model $y \sim p(\cdot \mid \theta, \xi)$, which, together with the prior $p(\theta)$, defines the data-generating process. Classical BED has largely been formulated as a problem of reducing the expected uncertainty from the prior to the posterior for suitable uncertainty functions (Dawid, 1998; DeGroot, 1962). A canonical example is the expected information gain (EIG) (Lindley, 1956), which uses Shannon entropy.

Sequential and policy-based BED. In sequential BED, experiments are selected adaptively as observations are collected. At step t, the experimenter has access to the history $h_{t-1} := \{(\xi_1, y_1), \ldots, (\xi_{t-1}, y_{t-1})\}$, $h_0 := \emptyset$, and selects the next design according to a data-acquisition policy $\xi_t = \pi_d(\cdot \mid h_{t-1})$. The corresponding observation is then generated from $y_t \sim p(\cdot \mid \theta, \xi_t, h_{t-1})$, and the history is augmented as $h_t = (h_{t-1}, (\xi_t, y_t))$. Rather than solving a new design optimization problem after every observation, *policy-based BED* amortizes the adaptive experimental strategy into a parameterized policy π_d trained over simulated trajectories (Blau et al., 2022; Bracher et al., 2025; Dong et al., 2025; Foster et al., 2021; Huang et al., 2026a; 2025; Iqbal et al., 2024; Ivanova et al., 2021; Lim et al., 2022). Compared with traditional iterative approaches, this dramatically reduces computation at deployment and often produces better designs by learning policies that plan ahead and avoid direct posterior inference. For a prescribed horizon T, the policy is optimized according to the expected utility of the complete experimental trajectory, where the expectation is taken over the latent state, observations, and possibly stochastic design decisions induced by π_d. In classical formulations, the utility is most commonly information-theoretic, so that the policy is trained to collect observations that maximally reduce uncertainty, typically entropy, about θ.

Loss-driven BED. Many data-acquisition problems are ultimately driven by the need to make an explicit downstream decision (Dawid, 1998; Huang et al., 2024; Rossa et al., 2026). Let $\mathcal{A}$ denote an action space and let $\ell : \mathcal{A} \times \Theta \times \mathcal{H}_T \to \mathbb{R}$ be a task-specific loss encoding the cost of taking action a under latent state θ after observing history h_T. Given a design policy π_d, the Bayes-optimal terminal decision minimizes the posterior risk (Berger, 1985), and its expected value before data acquisition defines the expected future loss (EFL),

$$\mathrm{EFL}_\ell^{B,T}(\pi_d) := \mathbb{E}_{p(h_T;\pi_d)}\left[\inf_{a\in\mathcal{A}} \mathbb{E}_{p(\theta|h_T)}\left[\ell(a,\theta,h_T)\right]\right], \tag{1}$$

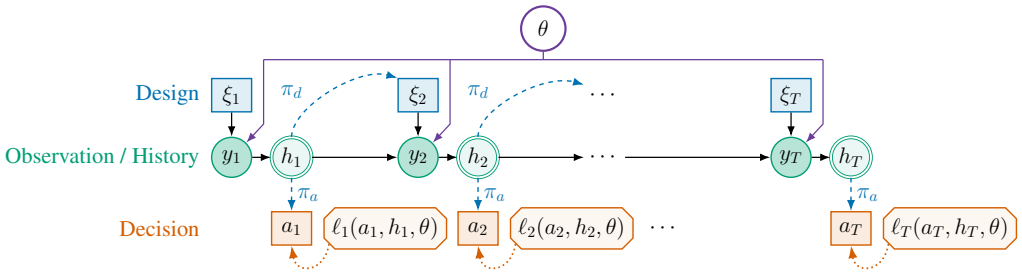

Figure 1: **MA-BED dependency structure.** *Solid arrows* describe the construction of the experimental trajectory: a design ξ_t generates an observation y_t, then appended together to the current history to form h_t. *Dashed arrows* represent decisions produced from the current history by the joint policy: h_t determines both the intermediate action a_t and the next design ξ_{t+1}. Each action is evaluated through the stepwise loss ℓ_t, and *dotted arrows* indicate the learning signal propagated from this loss back through the rollout. The multistep objective aggregates these intermediate losses across the trajectory.

which can then be optimized to learn the policy. This, though, can be computationally demanding due to the nested minimisation, and methods have generally relied on using losses that yield analytic (but typically themselves intractable) optima (Huang et al., 2026b), often coupled with learning amortized artifacts that help with the resulting nested estimation that must be performed.

Action-BED, introduced in Rossa et al. (2026), instead lifts this pointwise Bayesian decision problem to one of joint policy optimization by introducing an action policy $\pi_a \in \Pi_a := \{\pi \mid \pi : \mathcal{H}_T \to \mathcal{A}\}$. Under the usual conditions ensuring the existence of Bayes decision rules, this yields

$$\mathrm{EFL}_\ell^{B,T}(\pi_d) = \inf_{\pi_a \in \Pi_a} \mathbb{E}_{p(\theta)p(h_T|\theta;\pi_d)} \left[\ell(\pi_a(h_T), \theta, h_T)\right],$$

so the design and action policies can be optimized jointly to yield loss-driven design policies. Note here that the downstream action is terminal and only evaluated after all data is collected.

3 Multistep Action-BED

To address this restriction to terminal actions, we introduce MA-BED, which extends loss-driven BED to settings where task-relevant decisions are taken *throughout* data acquisition, rather than only once it is complete. The key idea will be to first allow for actions between each step of the data acquisition, with corresponding intermediate losses, and then to jointly learn a policy that both dictates how designs are chosen and maps from the data gathered so far to the intermediate action to take.

3.1 Decision-making Along the Acquisition Trajectory

In the formulation of Bayes-optimal experimental policies as minimizers of an EFL, the loss captures the downstream consequences of data acquisition through the quality of the action taken to solve the task. However, the resulting class of data-gathering problems is all based on terminal actions.

Many sequential decision problems also require task-relevant actions to be taken *during* data acquisition. In addition to the sequence of experimental designs, we therefore consider a sequence of actions $a_{1:T} \in \prod_{t=1}^T \mathcal{A}_t$, where a_t is taken after observing the outcome of the t-th experiment, based on the data available at this stage, $h_t \in \mathcal{H}_t$. The quality of the data then depends on the decisions it supports at multiple stages, rather than exclusively at the terminal step. This motivates replacing a single terminal loss by a trajectory-valued objective that assigns value to the sequence of decisions.

Definition 1 (Multistep trajectory loss)**.** *Let $\ell := (\ell_t)_{t=1}^T$, $\ell_t : \mathcal{A}_t \times \Theta \times \mathcal{H}_t \to \mathbb{R}$, be a family of measurable stage-wise losses, and let $\lambda := (\lambda_1, \ldots, \lambda_T) \in \mathbb{R}_+^T$ be a vector of non-negative temporal weights. The associated multistep trajectory loss is defined by*

$$L_{\ell,\lambda}(a_{1:T}, \theta, h_T) := \sum_{t=1}^T \lambda_t \, \ell_t(a_t, \theta, h_t). \tag{2}$$

Here, $\ell_t(a_t, \theta, h_t)$ measures the quality of the decision taken at step t, while λ_t controls the relative importance of that stage; the terminal formulation is recovered by setting $\lambda_t = 0$ for $t < T$. The

latent state θ should be understood broadly: different losses may depend on different latent aspects of the underlying problem, which may coincide with model parameters or represent only task-relevant latent quantities. The dependence on h_t further allows the loss to vary with the current information state. Figure 1 summarizes the sequential decision structure adopted by MA-BED, with the explicit dependencies between the components of the experimental rollout.

Following Bayesian decision theory, the resulting objective for the design, or acquisition, policy, π_d, is the expected multistep loss attained when, at each stage, a rational agent takes the Bayes-optimal action given the information acquired up to that point, yielding the *multistep* EFL:

$$\mathrm{EFL}_{\ell,\lambda}^{B,\mathrm{MS}}(\pi_d) := \sum_{t=1}^{T} \lambda_t \, \mathbb{E}_{p(h_t;\pi_d)} \left[\inf_{a_t \in \mathcal{A}_t} \mathbb{E}_{p(\theta|h_t)} \left[\ell_t(a_t, \theta, h_t) \right] \right]. \tag{3}$$

Thus, the design policy is assessed by the quality of the Bayes-optimal decisions supported by the information it provides throughout the trajectory.

3.2 Joint Optimization of the Multistep EFL

The decision problem defined in (3) involves a collection of nested pointwise optimizations, one at each step of the acquisition. Following the policy-lifting principle of Rossa et al. (2026), we instead reformulate these optimizations as learning a set of measurable action policies acting directly on the gathered data. Specifically, we introduce a sequence of decision rules, or action policies, $\pi_a = (\pi_{a,t})_{t=1}^{T} \in \Pi_a^{1:T}$, such that $\xi_t = \pi_d(h_{t-1})$, $a_t = \pi_{a,t}(h_t)$, and each design and action depends only on the information available when it is taken. The co-optimisation of the resulting lifted multistep objective with respect to all policies now admits a Bayes-optimal design strategy for our problem, as formalized in the following result.

Proposition 1 (Multistep policy lifting)**.** *Under standard measurability and integrability assumptions detailed in Appendix A, for any $\pi_d \in \Pi_d$ and any weights $\lambda \in \mathbb{R}_+^T$,*

$$\mathrm{EFL}_{\ell,\lambda}^{B,\mathrm{MS}}(\pi_d) = \inf_{\pi_a \in \Pi_a^{1:T}} \mathrm{EFL}_{\ell,\lambda}^{\mathrm{MS}}(\pi_d, \pi_a), \tag{4}$$

$$\textit{where} \quad \mathrm{EFL}_{\ell,\lambda}^{\mathrm{MS}}(\pi_d, \pi_a) := \mathbb{E}_{p(\theta)p(h_T|\theta;\pi_d)} \left[\sum_{t=1}^{T} \lambda_t \ell_t \left(\pi_{a,t}(h_t), \theta, h_t \right) \right]. \tag{5}$$

Consequently, the joint optimization problem is $(\pi_d^B, \pi_a^B) \in \operatorname{argmin}_{\pi_d \in \Pi_d,\, \pi_a \in \Pi_a^{1:T}} \mathrm{EFL}_{\ell,\lambda}^{\mathrm{MS}}(\pi_d, \pi_a)$.

We thus see that the resulting multistep objective is a weighted sum of horizon-specific EFLs, each evaluating the decision enabled by a successive prefix h_t of the same acquisition trajectory. The proof, together with the decomposition into horizon-specific EFLs, is given in Appendix A.

In practice, we parameterize the policies as neural networks π_d^ϕ and π_a^ψ, with respective parameters ϕ and ψ. In principle, one could use a separate action network at each stage, since each $\pi_{a,t}$ solves a stage-wise decision problem from the information available in h_t. Typically, and in our experiments, however, the same task is solved repeatedly as information accumulates, with successive histories forming extensions of the same acquisition trajectory. We therefore share action-policy parameters across horizons and use a single network that maps each h_t to the corresponding action a_t. Other architectures are equally compatible with the framework, including stage-specific action networks or shared representations between π_d^ϕ and π_a^ψ themselves. Architectural details are given in Appendix D.

We optimize the multistep objective with stochastic gradient-based methods, which require an unbiased estimate of its gradient. Assuming the observation model is differentiably reparameterisable (Kingma and Welling, 2014) and that the usual regularity conditions hold (Mohamed et al., 2020), we can use the pathwise estimator

$$\widehat{\nabla}_{\phi,\psi} = \frac{1}{B} \sum_{b=1}^{B} \nabla_{\phi,\psi} \left[\sum_{t=1}^{T} \lambda_t \, \ell_t \Big(\pi_{a,t}^{\psi}(h_t^{(b)}), \theta^{(b)}, h_t^{(b)} \Big) \right], \qquad h_t^{(b)} = h_t(\theta^{(b)}, \epsilon_{1:t}^{(b)}; \phi), \tag{6}$$

where $(\theta^{(b)}, \epsilon_{1:T}^{(b)})_{b=1}^{B}$ are independent samples from $p(\theta)q(\epsilon_{1:T})$ and $q(\epsilon_{1:T})$ is a noise outsourcing distribution. Pathwise gradients propagate through the entire acquisition trajectory, and a single rollout therefore provides supervision at all horizons while preserving the non-myopic dependence of future designs on past observations; see Appendix C.2 for full derivation of the pathwise gradient. When reparameterisation is not possible, score-function (REINFORCE) estimators (Williams, 1992) can be used instead.

4 BEYOND TERMINAL OBJECTIVES

Multistep objectives affect both *what* sequential BED problems can represent and *how* policies are optimized. We now discuss important aspects of each, and provide some associated theoretical insights.

4.1 BEYOND FIXED-HORIZON DECISION PROBLEMS

MA-BED allows us to generalize from the standard experimental design setting to a more general notion of *online decision-making*, where actions must be taken repeatedly from the information available so far, as in adaptive tracking, sequential monitoring, or online prediction. An important special case of this is *anytime decision-making*, where a single policy should perform well across a range of possible horizons rather than being specialised to one fixed budget. The weights λ then encode the desired trade-off between performance at different stages of the rollout.

Multistep objectives also arise naturally when the deployment horizon is uncertain. At deployment, if a decision is required at a single, but random, stopping time $\tau \in \{1, \dots, T\}$, then we set each λ_t to the stopping probabilities $\mathbb{P}(\tau = t)$. Alternatively, we can use $\lambda_t = \mathbb{P}(\tau \geq t)$ if decisions incur losses continuously until a random interruption time.

4.2 EXISTENCE OF TERMINAL-PROBLEM-PRESERVING INTERMEDIATE LOSSES

We now show that any terminal decision problem admits a series of non-trivial intermediary losses whose multistep objective $\mathrm{EFL}^{B,\mathrm{MS}}_{\ell,\lambda}(\pi_d)$ preserves the minima of the original terminal problem regardless of what temporal weights we use. In other words, dense intermediate supervision can, in principle, leave the terminal problem exactly unchanged, as formalized in the following result.

Theorem 1 (Bellman-consistent intermediate losses)**.** *Let (π_d^B, π_a^B) be Bayes-optimal for the terminal-only problem. Retain the original terminal loss ℓ_T at T. For $t < T$ use the next design ξ_{t+1} as the local action and define the Bellman-consistent intermediate losses for $0 < t < T-1$ (with $h_0 = \emptyset$):*

$$\ell_t^{\mathrm{BC}}(\xi_{t+1}, \theta, h_t) := \mathbb{E}_{p(h_T|\theta,h_t,\xi_{t+1};\pi_d^B)}\left[\ell_T\big(\pi_a^B(h_T), \theta, h_T\big)\right] - V_t^\star(h_t), \tag{7}$$

$$\text{where} \quad V_t^\star(h_t) := \inf_{\xi_{t+1}\in\Xi_{t+1}} \mathbb{E}_{p(\theta|h_t)p(h_T|\theta,h_t,\xi_{t+1};\pi_d^B)}\left[\ell_T\big(\pi_a^B(h_T), \theta, h_T\big)\right], \tag{8}$$

is the optimal continuation value, and $p(h_T|\theta, h_t, \xi_{t+1}; \pi_d^B)$ indicates the distribution on data if we choose design ξ_{t+1} now and then follow the Bayes-optimal design policy of our original terminal-only problem thereafter. Then, for any non-negative temporal weights with $\lambda_T > 0$,

$$\operatorname*{argmin}_{\pi_d\in\Pi_d}\left\{\sum_{t=0}^{T-1}\lambda_t\,\mathbb{E}_{p(\theta,h_t;\pi_d)}\left[\ell_t^{\mathrm{BC}}(\pi_d(h_t),\theta,h_t)\right] + \lambda_T\ \mathrm{EFL}^{B,T}_{\ell_T}(\pi_d)\right\} = \operatorname*{argmin}_{\pi_d\in\Pi_d}\ \mathrm{EFL}^{B,T}_{\ell_T}(\pi_d).$$

Furthermore, the Bayes optimal design policy can be retrieved by greedily minimising the expected Bellman-consistent intermediate loss at each step,

$$\pi_d^B(h_t) \in \operatorname*{argmin}_{\xi_{t+1}\in\Xi_{t+1}} \mathbb{E}_{p(\theta|h_t)}\left[\ell_t^{\mathrm{BC}}(\xi_{t+1},\theta,h_t)\right], \quad \forall t \in \{0, \cdots, T-1\}. \tag{9}$$

A more comprehensive and formal version of this theorem, including the required regularity constraints and proof, is provided in Appendix B. The upshot of the result is that not only can, in principle, dense intermediate supervision leave the terminal problem exactly unchanged under arbitrary weighting of the intermediate losses, it can also produce a variant of the problem where decisions can be made greedily while maintaining Bayes-optimality of the original problem (which is equivalent to setting each $\lambda_t/\lambda_{t+1} \to \infty$ in the multistep formulation, as the next loss then always dominates future losses), thereby forgoing the need to backpropagate gradients between steps.

This approach is not usually directly followed in practice as the required Bellman-consistent intermediate losses are generally intractable. However, introducing multistep losses when we only really care about the terminal action can now be viewed as tractable, decision-aware surrogates for this inaccessible stage-wise notion of terminal progress. They will generally alter the optimization problem unless they match the Bellman-consistent intermediate losses, but they can provide local supervision at intermediate horizons while remaining aligned with the downstream task if chosen appropriately.

4.3 Dense Supervision for Long-Horizon Terminal Objectives

A recurring empirical observation in policy-based BED is that performance deteriorates as the acquisition horizon increases (Blau et al., 2022; Foster et al., 2021). Similar issues arise across machine learning whenever supervision must propagate through many dependent transformations, motivating approaches based on denser or more local learning signals (Andrychowicz et al., 2017; Ng et al., 1999). Our multistep formulation brings this principle to BED: intermediate losses shorten credit-assignment paths and reduce reliance on long products of rollout Jacobians, limiting the multiplicative attenuation or amplification of sensitivities that can lead to vanishing, exploding, or high-variance gradients. This can ease policy optimization and, as we show empirically, even improve terminal performance relative to methods that directly optimize the terminal loss. Appendix C formalizes this path-length effect.

In short, introducing intermediary losses if we only care about terminal performance introduces a trade-off between improving the signal-to-noise ratio of our gradients through providing shorter paths, and biasing the gradients away from the true objective. Let $g_T = \nabla_\phi \mathcal{J}_T$, $g_A = \nabla_\phi \mathcal{J}_A$, $g_\beta = g_T + \beta g_A$, where $\mathcal{J}_T$ denotes the terminal EFL and $\mathcal{J}_A$ only collects the intermediate objectives. Further, let G_T and G_A be stochastic estimators of g_T and g_A, and $u = g_T/\|g_T\|$. For sufficiently small $\beta > 0$, the signal-to-noise ratio along the terminal-gradient direction improves whenever

$$\frac{u^\top g_A}{\|g_T\|} > \frac{\mathrm{Cov}(u^\top G_T, u^\top G_A)}{\mathrm{Var}(u^\top G_T)}, \tag{10}$$

with the ratio of these terms indicating the level of improvement. The left-hand side measures the relative gain in terminally aligned signal, while the right-hand side captures the first-order increase in terminal-direction variance due to correlated gradient noise. Hence, multistep supervision is most useful when shorter credit-assignment paths contribute sufficiently aligned signal without introducing commensurate noise. See Appendices C.4–C.5 for full derivations.

The multistep update remains a strict first-order descent direction for the terminal objective when $\|g_T\|^2 + \beta\langle g_T, g_A\rangle > 0$. Thus, aligned auxiliary gradients preserve terminal descent for any $\beta \geq 0$, whereas conflicting gradients require a sufficiently small auxiliary weight.

5 Related Work

Policy-based BED is typically cast as a finite-horizon decision problem in which a design policy is learned by optimizing a terminal objective (Foster et al., 2021; Hedman et al., 2025; Huan and Marzouk, 2016; Ivanova et al., 2021). This objective is usually information-theoretic, though recent work targets downstream decision losses instead (Filstroff et al., 2021; Go et al., 2026; Huang et al., 2024; 2026b; Rossa et al., 2026). Either way, only the terminal history is valued. MA-BED changes the objective itself: by pairing every design with an intermediate action and loss, it covers problems where decisions are made as data arrive, or where there is no fixed endpoint.

Several methods ease long-horizon training by telescoping a terminal information-theoretic objective into per-step increments used as dense rewards (Blau et al., 2022; Cheng and Huan, 2025; Iqbal et al., 2024; Lim et al., 2022; Shen et al., 2025; Shen and Huan, 2023), akin to reward shaping in RL (Ng et al., 1999). The increments sum back to the terminal quantity, so the optimal policy is unchanged, and benefits are only seen when using RL-based policy training, not end-to-end gradients. Moreover, the construction does not extend to general decision losses and no intermediate actions are produced. Nor can this be remedied post hoc: fitting an action predictor on top of a terminally trained policy leaves the designs unaware of the intermediate losses, which we show is sub-optimal in Section 6.1. In MA-BED, a dense signal arises from the objective itself and is available for any per-step loss.

Optimal stopping in sequential design has a long history (Haggstrom, 1966), and existing methods either decide only when to stop a fixed sampling process (Rossell and Müller, 2013), derive stopping rules for a specific criterion such as cost-adjusted simple regret (Xie et al., 2026), or jointly learn design and stopping policies for information-theoretic objectives (Cheng and Huan, 2025). Unlike Cheng and Huan (2025), however, MA-BED accommodates arbitrary decisions and losses. In addition, because the policy is trained to make a good decision at every step, the stopping criterion need not be fixed in advance, and can be an exhausted budget, interruption, or user-specified threshold on the confidence of the current action, chosen at deployment without retraining (Section 6.3).

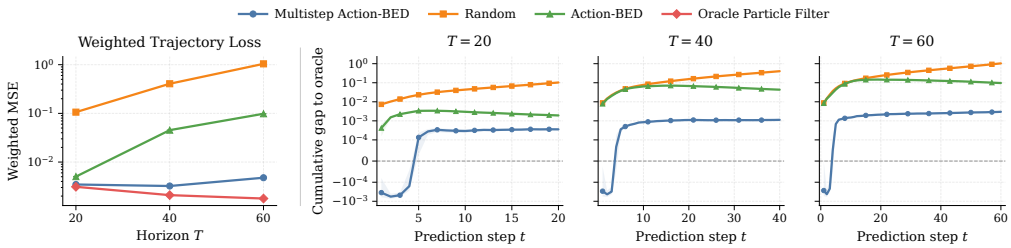

Figure 2: **Dynamic tracking performance across horizons.** Results are averaged over 10,000 simulated trajectories. **Left:** Trajectory-level tracking loss, ℓ^{MS}_{track}, obtained by uniformly averaging the prediction loss over all steps of the rollout. **Right:** Cumulative excess tracking error relative to the oracle, i.e., the average prediction-error gap accumulated up to each step t. The particle filter is a privileged-information oracle that uses the true dynamics and observation model.

6 Experiments

We evaluate MA-BED across a range of complementary BED tasks that highlight the main benefits of multistep supervision. We first consider *anytime* scenarios where decisions are needed during acquisition, including intrinsically online tasks, and stochastic deployment horizons. We then examine two secondary effects: whether better intermediate predictions reduce the number of acquisitions needed to reach a target accuracy, and whether multistep supervision can also improve terminal performance.

Baselines and Experimental Protocol. We compare MA-BED against competitive policy-based BED baselines, including ACTION-BED and DAD (Foster et al., 2021), together with random design. Each method is trained on its native objective, with auxiliary action networks additionally trained when needed to evaluate downstream predictive performance. To ensure a fair comparison, we match training and evaluation protocols whenever possible, use conservative choices for MA-BED when exact matching is not feasible, and keep the policy backbone architecture fixed across methods so that performance differences cannot be attributed to architectural or protocol discrepancies.

6.1 Dynamic Tracking

We consider a dynamic tracking problem in which predictions must be made *during* data acquisition. Each episode is governed by latent parameters θ specifying the initial state and target dynamics, under which the target evolves stochastically in $\mathbb{R}^2$ according to a regime-switching motion model alternating between straight, left-turn, and right-turn dynamics. At each step t, the design policy places a sensor at $\xi_t \in \mathbb{R}^2$ and observes a noisy distance-dependent signal $y_t \in \mathbb{R}$. From the history available at that step, the action policy predicts the current target position, $\hat{x}_t = \pi_a(h_t)$. Since tracking naturally requires accurate predictions throughout the trajectory, we evaluate policies using $\ell^{MS}_{\text{track}}(\hat{x}, x) = \frac{1}{T}\sum_{t=1}^{T} ||\hat{x}_t - x_t||_2^2$. The task is therefore intrinsically online as the relevant quantity is the evolving state x_t, and a prediction must be made from every prefix h_t. This also makes EIG-based methods such as DAD, which target episode-level latent parameters θ, less directly aligned with the deployment objective; full details are provided in Appendix E.1.

Figure 2 shows that MA-BED achieves lower trajectory-averaged tracking loss ℓ^{MS}_{track} than all baselines at every horizon, closely matching the oracle policy in weighted MSE. The cumulative excess-error curves further show that MA-BED remains consistently closer to the oracle throughout acquisition, accumulating substantially less prediction error than competing policies as the rollout progresses. Together, these results show that multistep supervision enables the design policy to adapt its acquisition strategy to support accurate predictions throughout the trajectory.

6.2 Decision-Oriented CES with Stochastic Participation

We next consider the Constant Elasticity of Substitution (CES) model, a standard BED benchmark (Blau et al., 2022; Foster et al., 2021; 2019; Hedman et al., 2025; Huang et al., 2025; Kerrigan et al., 2025), originally introduced in behavioural economics to model substitution between goods (Arrow et al., 1961). A participant has hidden preferences θ over baskets of three goods. At each step, the

Table 1: **Stochastic-horizon CES ($T = 10$).** Expected normalized recommendation (10^{-2}) regret under the stopping-time distribution induced by each participant dropout rate q.

Method	$q = 0.05$	$q = 0.1$	$q = 0.2$	$q = 0.3$	$q = 0.4$
RANDOM	12.15 ± 0.08	13.80 ± 0.09	16.91 ± 0.10	18.97 ± 0.12	20.35 ± 0.13
DAD	8.69 ± 0.05	12.95 ± 0.06	19.42 ± 0.09	23.90 ± 0.12	26.54 ± 0.14
ACTION-BED	1.66 ± 0.01	3.17 ± 0.02	6.21 ± 0.04	8.84 ± 0.05	11.46 ± 0.07
MA-BED	$\mathbf{1.24 \pm 0.01}$	$\mathbf{2.36 \pm 0.02}$	$\mathbf{4.43 \pm 0.03}$	$\mathbf{6.50 \pm 0.05}$	$\mathbf{8.26 \pm 0.06}$

design policy presents a pair of baskets and observes the participant's relative preference, with the downstream goal of recommending a high-utility basket under a fixed budget B. We modify the standard CES setting so that the participant may leave after any step with constant dropout rate q; the departure time therefore determines a stochastic recommendation horizon. The objective is therefore the expected normalized regret at departure: as q increases, useful recommendations must be obtained earlier because the participant is less likely to remain for later experiments. Full task and training details are given in Appendix E.2.

Table 1 shows that MA-BED achieves the lowest expected normalized regret for every hazard rate considered. This indicates that accounting explicitly for when a recommendation may be required changes the optimal acquisition strategy: the policy must trade off refining the eventual recommendation against obtaining useful recommendations early enough to be presented before the participant leaves. This adaptation is visible as q increases: MA-BED becomes increasingly better compared to ACTION-BED and shifts information gathering toward earlier decisions when early departure is more likely.

6.3 SOURCE LOCATION FINDING

Source location finding is a standard BED benchmark (Blau et al., 2022; Foster et al., 2021; Huang et al., 2025; Ivanova et al., 2021; Phillips et al., 2026; Rossa et al., 2026), based on the acoustic energy attenuation model of Sheng and Hu (2005). The unknown parameter $\theta \in \mathbb{R}^{2\times d}$ gives the locations of two sources in a d-dimensional space. At each step, the policy places a sensor at ξ_t and observes a noisy aggregate signal whose intensity decays with distance to the sources. The objective is to sequentially select informative sensor locations so as to accurately predict both source positions, evaluated with the permutation-invariant MSE, ℓ^{MSE}; full details are provided in Appendix E.3.

Anytime and Stochastic-Horizon Performance. We consider deployment at an exogenous stochastic horizon τ, with $p(\tau = t) = \lambda_t \propto 0.95^{T-t}$, and evaluate the corresponding *Random-Horizon MSE* (RH-MSE), $\mathbb{E}_\tau[\ell_\tau^{MSE}]$. Figure 3 shows its evolution for varying horizons T across dimensions d and methods. MA-BED achieves substantially lower RH-MSE throughout the considered settings, showing that multistep supervision learns policies that support effective predictions when the available experimental budget is uncertain. This comparison isolates the design strategy because each baseline is equipped with its own multistep action network, retrained on the trajectories induced by that policy with the same optimization budget used for policy training. The per-step losses reported in Appendix Figures 6 and 7 complement this analysis by showing that MA-BED yields lower prediction error than the closest policy baseline, ACTION-BED, at nearly every step of the trajectory.

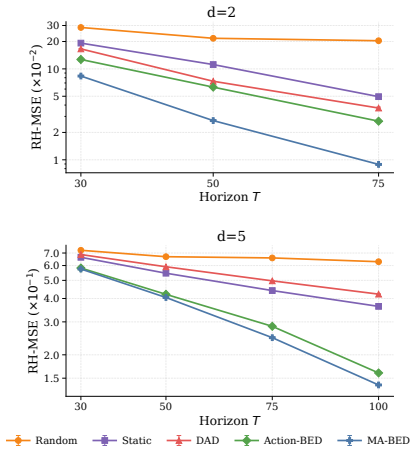

Figure 3: **Random-horizon MSE, $\mathbb{E}_\tau[\ell_\tau^{MSE}]$.** Evaluates source-location prediction at a stochastic deployment horizon, averaged over 50,000 rollouts per setting.

Adaptive Time-Stopping for Cheaper Source-Location Identification. The previous results show that MA-BED performs better on average at intermediate horizons, suggesting that multistep supervision puts greater pressure on the policy to reach accurate predictions earlier. We test this

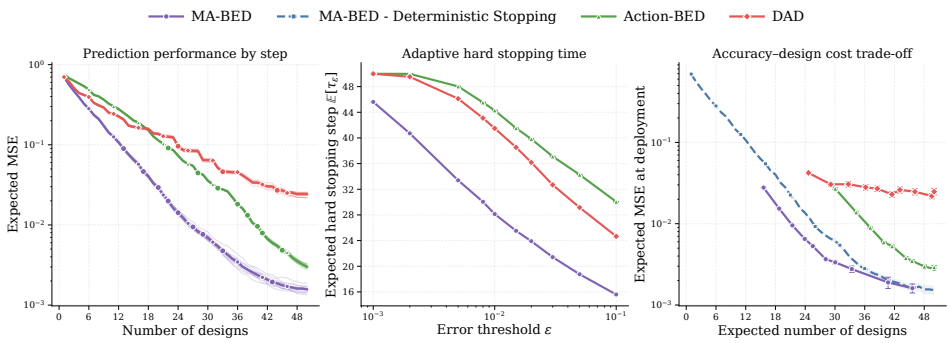

Figure 4: **Adaptive stopping for source location finding ($T = 50,\ d = 2$). Left:** expected prediction MSE versus acquisition step t; markers indicate, for each method and threshold ϵ, the first fixed design count at which its mean MSE falls below ϵ, without adaptive stopping. **Middle:** expected stopping time $\mathbb{E}[\tau_{\epsilon,\delta}]$ of the adaptive stopping rule, shown as a function of the error threshold ϵ, for a fixed confidence level of 0.95. **Right:** accuracy–design-cost trade-off, reporting the expected deployment MSE versus expected number of designs under adaptive stopping. All methods use their learned confidence head; we additionally show MA-BED without adaptive stopping as a reference.

through adaptive stopping while keeping each design policy and its multistep action network frozen. On top of the resulting rollout trajectories, we train a confidence head to estimate $\Pr(\ell_t^{MSE} \leq \epsilon)$, and stop acquisition once this probability exceeds $1 - \delta$. As a non-adaptive reference, we also evaluate MA-BED at the first fixed acquisition step whose average prediction loss falls below each ϵ. Across a range of target accuracies ϵ, Figure 4 shows that MA-BED satisfies this stopping criterion substantially earlier than competing policies. This indicates that, within the diversity of trajectories induced by a fixed policy, prediction quality can be assessed adaptively and exploited to stop acquisition earlier on easier instances while allocating additional measurements to harder ones. Thus, the improved intermediate predictions learned by MA-BED translate directly into a lower effective acquisition cost.

Long-Horizon Terminal Performance. As a secondary benefit, we examine whether MA-BED also improves the final decision, despite not being designed specifically for terminal performance. Across increasingly challenging settings, with varying horizons T and dimensions d, we evaluate the terminal performance with ℓ_T^{MSE}. The terminal-MSE results reported in Appendix Tables 12 and 13 show that MA-BED outperforms every baseline and, in particular, ACTION-BED, which directly optimizes the terminal objective, at all horizons for $d = 2$ and at the longest horizons $T \in 75, 100$ for $d = 5$. Thus, beyond improving intermediate predictions, multistep supervision can also lead optimization to better terminal solutions, with the benefit becoming most apparent for longer and more challenging rollouts.

Robustness and Computational Cost. For a subset of settings, constrained by computational cost, we repeat training over five independent seeds; the gains remain consistent across runs (Appendix Table 14). They also come at little to no additional training cost relative to the fastest learned policy baseline under a conservative comparison that matches optimization budgets (Appendix Table 15).

7 CONCLUSION

We introduced MA-BED, a loss-driven framework that extends policy-based BED from fixed terminal decisions to trajectory-valued objectives defined throughout data acquisition. This broader formulation enables adaptive design in settings where useful decisions must be made during the rollout or where the deployment horizon is uncertain. Across our experiments, MA-BED supports accurate online decisions and adapts acquisition to uncertain deployment horizons, illustrating how multistep objectives broaden the range of sequential decision problems accessible to policy-based BED. As an additional empirical benefit, intermediate supervision can also facilitate optimization when the underlying task remains purely terminal. In our long-horizon experiments, MA-BED yields stronger intermediate predictions and, in challenging regimes, improved terminal performance, suggesting that multistep supervision can also facilitate the training of long adaptive acquisition policies.

AI USE STATEMENT

Generative AI tools were used during the preparation of this work to assist with improving the clarity of the manuscript, to double-check the correctness of the mathematical derivations and proofs, and to assist with the implementation and debugging of parts of the experimental code. All AI-assisted code was reviewed and tested against the intended experimental procedures, while the final versions of all mathematical derivations and proofs were directly written and checked by the authors. The authors take full responsibility for the complete content of the paper, including all text, mathematical claims, and experimental results.

REPRODUCIBILITY STATEMENT

We provide the information required to reproduce both the theoretical and empirical results of this work in the appendices. Appendices A–C state the assumptions used in our theoretical analysis and provide the complete derivations and proofs of the results presented in the main text. Appendix D specifies the neural architectures used for the design and action policies. Appendix E provides the full experimental setup for each task, including the data-generating processes, loss functions, baseline definitions, model architectures, exact environment parameters, optimization hyperparameters, training procedures, and evaluation protocols. We also report the computational resources used for the experiments and additional robustness and computational-cost results. The code implementing MA-BED, the baselines, and all experiments will be made publicly available upon publication to facilitate reproduction of the reported results.

REFERENCES

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. (2017). Hindsight experience replay. In *Advances in neural information processing systems*, volume 30.

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

Bellman, R. (1952). On the theory of dynamic programming. *Proceedings of the National Academy of Sciences*, 38(8):716–719.

Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer New York, 2nd edition.

Blau, T., Bonilla, E. V., Chades, I., and Dezfouli, A. (2022). Optimizing sequential experimental design with deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2107–2128.

Bracher, N., Kühmichel, L., Ivanova, D. R., Intes, X., Bürkner, P.-C., and Radev, S. T. (2025). JADAI: Jointly Amortizing Adaptive Design and Bayesian Inference. arXiv:2512.22999 [stat].

Case, B., Dye-Braumuller, K. C., Evans, C., Li, H., Rustin, L., and Nolan, M. S. (2024). Adapting vector surveillance using bayesian experimental design: An application to an ongoing tick monitoring program in the southeastern united states. *Ticks and Tick-borne Diseases*, 15(3):102329.

Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304.

Cheng, C. and Huan, X. (2025). Optimal stopping for sequential bayesian experimental design.

Dawid, A. P. (1998). Coherent measures of discrepancy, uncertainty and dependence, with applications to Bayesian predictive experimental design. Technical Report 139, University College London.

DeGroot, M. H. (1962). Uncertainty, information, and sequential experiments. *The Annals of Mathematical Statistics*, 33(2):404–419.

Dong, J., Jacobsen, C., Khalloufi, M., Akram, M., Liu, W., Duraisamy, K., and Huan, X. (2025). Variational Bayesian optimal experimental design with normalizing flows. *Computer Methods in Applied Mechanics and Engineering*, 433:117457.

Filstroff, L., Sundin, I., Mikkola, P., Tiulpin, A., Kylmäoja, J., and Kaski, S. (2021). Targeted active learning for Bayesian decision-making. *arXiv preprint arXiv:2106.04193*.

Foster, A., Ivanova, D. R., Malik, I., and Rainforth, T. (2021). Deep adaptive design: Amortizing sequential bayesian experimental design. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3384–3395.

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

Go, J., Qian, X., and Yoon, B.-J. (2026). Goal-driven Bayesian optimal experimental design for robust decision-making under model uncertainty. *arXiv preprint arXiv:2605.26093*.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113.

Haggstrom, G. W. (1966). Optimal stopping and experimental design. *The Annals of Mathematical Statistics*, 37(1):7–29.

Hedman, M., Ivanova, D. R., Guan, C., and Rainforth, T. (2025). Step-DAD: Semi-amortized policy-based bayesian experimental design. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 22904–22923.

Hero, A. O. and Cochran, D. (2011). Sensor management: Past, present, and future. *IEEE Sensors Journal*, 11(12):3064–3075.

Huan, X., Jagalur, J., and Marzouk, Y. (2024). Optimal experimental design: Formulations and computations. *Acta Numerica*, 33:715–840.

Huan, X. and Marzouk, Y. M. (2016). Sequential Bayesian optimal experimental design via approximate dynamic programming.

Huang, D., Guo, Y., Acerbi, L., and Kaski, S. (2024). Amortized Bayesian experimental design for decision-making. In *Advances in Neural Information Processing Systems*.

Huang, D., Huang, Z., Hassan, C., Acerbi, L., Kaski, S., and Rainforth, T. (2026a). Efficient adaptive data acquisition via pretrained belief representations.

Huang, D., Wen, X., Bharti, A., Kaski, S., and Acerbi, L. (2025). ALINE: Joint amortization for bayesian inference and active data acquisition. In *Advances in Neural Information Processing Systems*, volume 38.

Huang, Z., Bickford Smith, F., and Rainforth, T. (2026b). Loss-driven Bayesian active learning. In *International Conference on Artificial Intelligence and Statistics*.

Hurtado, M., Zhao, T., and Nehorai, A. (2008). Adaptive polarized waveform design for target tracking based on sequential bayesian inference. *IEEE Transactions on Signal Processing*, 56(3):1120–1133.

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

Ivanova, D. R., Foster, A., Kleinegesse, S., Gutmann, M. U., and Rainforth, T. (2021). Implicit deep adaptive design: Policy-based experimental design without likelihoods. In *Advances in Neural Information Processing Systems*, volume 34, pages 25785–25798.

Jones, A., Schwager, M., and Belta, C. (2013). A receding horizon algorithm for informative path planning with temporal logic constraints. In *2013 IEEE International Conference on Robotics and Automation*, pages 5019–5024. IEEE.

Kang, S. Y., McGree, J. M., Drovandi, C. C., Caley, M. J., and Mengersen, K. L. (2016). Bayesian adaptive design: improving the effectiveness of monitoring of the great barrier reef. *Ecological Applications*, 26(8):2637–2648.

Kerrigan, G., Naesseth, C. A., and Rainforth, T. (2025). A geometric approach to optimal experimental design.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *International Conference on Learning Representations*.

Kreucher, C. M., Hero, A. O., Kastella, K. D., and Morelande, M. R. (2007). An information-based approach to sensor management in large dynamic networks. *Proceedings of the IEEE*, 95(5):978–999.

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

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

Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. (2020). Monte Carlo Gradient Estimation in Machine Learning. *Journal of Machine Learning Research*, 21(132):1–62.

Ng, A. Y., Harada, D., and Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*.

Pallmann, P., Bedding, A. W., Choodari-Oskooei, B., Dimairo, M., Flight, L., Hampson, L. V., Holmes, J., Mander, A. P., Odondi, L., Sydes, M. R., Villar, S. S., Wason, J. M. S., Weir, C. J., Wheeler, G. M., Yap, C., and Jaki, T. (2018). Adaptive designs in clinical trials: Why use them, and how to run and report them. *BMC Medicine*, 16:29.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 8024–8035.

Phillips, A., Kerrigan, G., and Rainforth, T. (2026). Bayesian experimental design via score matching. In Perković, E. and Malinsky, D., editors, *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337 of *Proceedings of Machine Learning Research*, pages 5393–5426. PMLR.

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

Rossa, T., Phillips, A., and Rainforth, T. (2026). Action-BED: Task-driven bayesian experimental design with singly intractable objectives. *arXiv preprint arXiv:2606.23662*.

Rossell, D. and Müller, P. (2013). Sequential stopping for high-throughput experiments. *Biostatistics*, 14(1):75–86.

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

Shen, W. and Huan, X. (2023). Bayesian sequential optimal experimental design for nonlinear models using policy gradient reinforcement learning. *Computer Methods in Applied Mechanics and Engineering*, 416:116304.

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

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256.

Xie, Q., Cai, L., Terenin, A., Frazier, P. I., and Scully, Z. (2026). Cost-aware stopping for bayesian optimization. In *Proceedings of the 43rd International Conference on Machine Learning*. PMLR.

A Decision-Theoretic Interpretation of the Multistep Objective

This section formalizes two complementary aspects of the multistep objective. We first show that the pointwise Bayes decision problem induced by each history can be lifted to an admissible action policy under standard measurable-selection conditions. We then show that the multistep expected future loss decomposes as a weighted sum of horizon-specific EFLs, and consequently of their Bayes counterparts.

A.1 Lifting Bayes-Optimal Problems to Multistep Action Policies

We first lift the collection of pointwise Bayes decision problems arising at the different acquisition horizons to an optimization over measurable multistep action policies. At any fixed horizon, this is the same measurable-selection argument used to relate Bayes-optimal actions and action policies in the Action-BED formulation (Rossa et al., 2026). We apply this result to each experimental prefix and then show that the resulting problems can be lifted jointly.

Per-horizon decision problems. Fix an admissible design policy $\pi_d \in \Pi_d$. For $h \in \mathcal{H}_t$, define the conditional risk

$$L_t(a, h) := \mathbb{E}[\ell_t(a, \theta, h) \mid h_t = h], \qquad a \in \mathcal{A}_t, \tag{11}$$

and the corresponding pointwise Bayes value

$$B_t(h) := \min_{a \in \mathcal{A}_t} L_t(a, h). \tag{12}$$

Regularity assumptions. For every $t \in \{1, \dots, T\}$, we assume that:

1. $\mathcal{H}_t$ is a standard Borel space and $\mathcal{A}_t$ is a non-empty compact metric space equipped with its Borel σ-algebra;
2. $L_t : \mathcal{A}_t \times \mathcal{H}_t \to \mathbb{R}$ admits a jointly measurable version and, for every $h \in \mathcal{H}_t$, $a \mapsto L_t(a, h)$ is lower semicontinuous;
3. the expectations below are well defined. A convenient sufficient condition is the existence of an integrable envelope m_t such that

$$|\ell_t(a, \theta, h_t)| \le m_t(\theta, h_t) \qquad \text{for all } a \in \mathcal{A}_t, \qquad \mathbb{E}[m_t(\theta, h_t)] < \infty.$$

These conditions ensure that the Bayes correspondence $h \mapsto \operatorname{argmin}_{a \in \mathcal{A}_t} L_t(a, h)$ is non-empty and admits a measurable selector. In particular, defining

$$\Pi^{\text{meas}}_{a,t} := \{\pi_{a,t} : \mathcal{H}_t \to \mathcal{A}_t \ : \ \pi_{a,t} \text{ measurable}\}, \tag{13}$$

there exists $\pi^B_{a,t} \in \Pi^{\text{meas}}_{a,t}$ satisfying

$$\pi^B_{a,t}(h) \in \operatorname*{argmin}_{a \in \mathcal{A}_t} L_t(a, h)$$

for $p(h_t; \pi_d)$-almost every h. Accordingly, the standard single-horizon lifting gives

$$\mathbb{E}_{p(h_t;\pi_d)}\left[\min_{a \in \mathcal{A}_t} L_t(a, h_t)\right] = \min_{\pi_{a,t} \in \Pi^{\text{meas}}_{a,t}} \mathbb{E}_{p(h_t;\pi_d)}\left[L_t(\pi_{a,t}(h_t), h_t)\right]. \tag{14}$$

We now lift these pointwise problems jointly across all horizons. Define the multistep action-policy class

$$\Pi^{1:T}_a := \prod_{t=1}^{T} \Pi^{\text{meas}}_{a,t}, \qquad \pi_a = (\pi_{a,1}, \dots, \pi_{a,T}). \tag{15}$$

Proposition 2 (Multistep policy lifting, full statement of Proposition 1)**.** *Under the assumptions above, for any fixed design policy $\pi_d \in \Pi_d$ and any weights $\lambda \in \mathbb{R}^T_+$,*

$$\sum_{t=1}^{T} \lambda_t \, \mathbb{E}_{p(h_t;\pi_d)}\left[\min_{a_t \in \mathcal{A}_t} L_t(a_t, h_t)\right] = \min_{\pi_a \in \Pi^{1:T}_a} \sum_{t=1}^{T} \lambda_t \, \mathbb{E}_{p(h_t;\pi_d)}\left[L_t(\pi_{a,t}(h_t), h_t)\right].$$

Equivalently,

$$\sum_{t=1}^{T} \lambda_t \, \mathbb{E}_{p(h_t;\pi_d)} \left[\min_{a_t \in \mathcal{A}_t} \mathbb{E}_{p(\theta|h_t)} \left[\ell_t(a_t, \theta, h_t) \right] \right] = \min_{\pi_a \in \Pi_a^{1:T}} \sum_{t=1}^{T} \lambda_t \, \mathbb{E}_{p(\theta)p(h_t|\theta;\pi_d)} \left[\ell_t(\pi_{a,t}(h_t), \theta, h_t) \right]. \tag{16}$$

Proof. For each horizon t, the single-horizon lifting (14) implies that every admissible $\pi_{a,t}$ satisfies

$$\mathbb{E}\left[L_t(\pi_{a,t}(h_t), h_t) \right] \geq \mathbb{E}\left[B_t(h_t) \right].$$

Since $\lambda_t \geq 0$, any $\pi_a \in \Pi_a^{1:T}$ therefore satisfies

$$\sum_{t=1}^{T} \lambda_t \mathbb{E}\left[L_t(\pi_{a,t}(h_t), h_t) \right] \geq \sum_{t=1}^{T} \lambda_t \mathbb{E}\left[B_t(h_t) \right].$$

For every t, let $\pi_{a,t}^B$ be a measurable Bayes selector guaranteed by the assumptions above. Since

$$\pi_a^B := (\pi_{a,1}^B, \ldots, \pi_{a,T}^B) \in \prod_{t=1}^{T} \Pi_{a,t}^{\text{meas}},$$

all horizon-specific lower bounds are attained simultaneously:

$$L_t(\pi_{a,t}^B(h_t), h_t) = B_t(h_t) \qquad p(h_t;\pi_d)\text{-a.s.}$$

for every t. Hence

$$\min_{\pi_a \in \Pi_a^{1:T}} \sum_{t=1}^{T} \lambda_t \mathbb{E}\left[L_t(\pi_{a,t}(h_t), h_t) \right] = \sum_{t=1}^{T} \lambda_t \mathbb{E}\left[B_t(h_t) \right],$$

which proves the result. □

Remark 1 (Shared parametric action policies)**.** *The lifting above is stated over the product class $\Pi_a^{1:T} = \prod_t \Pi_{a,t}$, for which the horizon-specific decision rules may be chosen independently. In our implementation, a single neural network is shared across horizons, coupling these rules through common parameters. This restricts the practical policy class but does not alter the underlying decision-theoretic formulation; it only introduces a possible approximation gap relative to the unrestricted Bayes optimum.*

A.2 The Multistep Objective as a Sum of Horizon-Specific EFLs

We now use the lifting result above to give a direct decision-theoretic interpretation of the multistep objective. For each horizon $t \in \{1, \ldots, T\}$, define the policy-based horizon-specific EFL as

$$\text{EFL}_{\ell_t}^{t}(\pi_d, \pi_{a,t}) := \mathbb{E}_{p(\theta)p(h_t|\theta;\pi_d)} \left[\ell_t(\pi_{a,t}(h_t), \theta, h_t) \right], \tag{17}$$

and its Bayes counterpart as

$$\text{EFL}_{\ell_t}^{B,t}(\pi_d) := \mathbb{E}_{p(h_t;\pi_d)} \left[\min_{a_t \in \mathcal{A}_t} \mathbb{E}_{p(\theta|h_t)} \left[\ell_t(a_t, \theta, h_t) \right] \right]. \tag{18}$$

Thus, $\text{EFL}_{\ell_t}^{B,t}$ is the Bayes risk of the decision supported by the information collected up to horizon t. Recall that

$$\text{EFL}_{\ell,\lambda}^{\text{MS}}(\pi_d, \pi_a) := \mathbb{E}_{p(\theta)p(h_T|\theta;\pi_d)} \left[\sum_{t=1}^{T} \lambda_t \ell_t(\pi_{a,t}(h_t), \theta, h_t) \right]. \tag{19}$$

Proposition 3 (Horizon-specific EFL decomposition)**.** *Assume the stage-wise losses are integrable. Then, for every admissible (π_d, π_a),*

$$\text{EFL}_{\ell,\lambda}^{\text{MS}}(\pi_d, \pi_a) = \sum_{t=1}^{T} \lambda_t \, \text{EFL}_{\ell_t}^{t}(\pi_d, \pi_{a,t}). \tag{20}$$

Under the regularity assumptions of Section A.1, the corresponding Bayes multistep EFL satisfies

$$\mathrm{EFL}_{\ell,\lambda}^{B,\mathrm{MS}}(\pi_d) = \sum_{t=1}^{T} \lambda_t \ \mathrm{EFL}_{\ell_t}^{B,t}(\pi_d), \tag{21}$$

where

$$\mathrm{EFL}_{\ell,\lambda}^{B,\mathrm{MS}}(\pi_d) := \min_{\pi_a \in \Pi_a^{1:T}} \mathrm{EFL}_{\ell,\lambda}^{\mathrm{MS}}(\pi_d, \pi_a).$$

Consequently,

$$\pi_d^B \in \operatorname*{argmin}_{\pi_d \in \Pi_d} \sum_{t=1}^{T} \lambda_t \ \mathrm{EFL}_{\ell_t}^{B,t}(\pi_d). \tag{22}$$

Proof. By linearity of expectation,

$$\mathrm{EFL}_{\ell,\lambda}^{\mathrm{MS}} = \sum_{t=1}^{T} \lambda_t \, \mathbb{E}_{p(\theta)p(h_T|\theta;\pi_d)} \left[\ell_t(\pi_{a,t}(h_t), \theta, h_t) \right].$$

Since the t-th loss depends on the full rollout only through its prefix h_t, marginalizing the law of h_T onto h_t gives

$$\mathbb{E}_{p(\theta)p(h_T|\theta;\pi_d)} \left[\ell_t(\pi_{a,t}(h_t), \theta, h_t) \right] = \mathrm{EFL}_{\ell_t}^{t}(\pi_d, \pi_{a,t}),$$

which proves (20). Equation (21) then follows directly from Proposition 2, and minimizing over π_d yields (22). □

Hence, the multistep objective admits a simple interpretation: it scalarizes the Bayes decision problems induced by the different experimental prefixes. The weights λ_t specify the relative decision-theoretic value assigned to performing well after each amount of acquired information.

B BELLMAN STRUCTURE OF TERMINAL AND MULTISTEP EFLS

We complement the prefix-EFL decomposition of Appendix A with a dynamic-programming interpretation of the resulting design problem Bellman (1952). Throughout this section, we reason after optimization of the downstream actions. At each horizon t, the action a_t is a readout of the current history h_t: it does not affect subsequent designs or observations, and, as in Appendix A, the action problems are not jointly constrained across horizons. They can therefore be optimized separately, yielding the prefix Bayes risk

$$B_t(h_t) := \inf_{a_t \in \mathcal{A}_t} \mathbb{E}_{p(\theta|h_t)} \left[\ell_t(a_t, \theta, h_t) \right]. \tag{23}$$

We assume throughout that the relevant conditional expectations are finite and that the required measurable minimizers exist.

B.1 BELLMAN RECURSION FOR THE MULTISTEP OBJECTIVE

By Proposition 3, the Bayes multistep EFL associated with a design policy π_d can be written as

$$\mathrm{EFL}^{B,\mathrm{MS}}_{\ell,\lambda}(\pi_d) = \mathbb{E}_{p(h_T;\pi_d)} \left[\sum_{t=1}^{T} \lambda_t B_t(h_t) \right]. \tag{24}$$

Hence, after optimizing the downstream decisions, the design problem is a finite-horizon sequential control problem with stage cost $\lambda_t B_t(h_t)$. For $t \in \{0, \dots, T\}$, define the optimal continuation value

$$V_t^{\mathrm{MS}}(h_t) := \inf_{\pi_{d,t+1:T}} \mathbb{E}_{p(h_{t+1:T}|h_t;\pi_{d,t+1:T})} \left[\sum_{s=t+1}^{T} \lambda_s B_s(h_s) \right], \tag{25}$$

where the infimum is over admissible continuation policies from steps $t+1$ to T, and $V_T^{\mathrm{MS}} \equiv 0$. Here, $p(h_{t+1:T} \mid h_t; \pi_{d,t+1:T})$ denotes the continuation law obtained by initializing the acquisition process at the current history h_t and rolling it out autoregressively under the continuation policy. Since the complete history is itself an information state, no additional Markov assumption is required. The principle of optimality gives

$$V_{t-1}^{\mathrm{MS}}(h_{t-1}) = \inf_{\xi_t \in \Xi_t} \mathbb{E}_{p(h_t|h_{t-1},\xi_t)} \left[\lambda_t B_t(h_t) + V_t^{\mathrm{MS}}(h_t) \right]. \tag{26}$$

Thus, the multistep formulation admits the usual Bellman decomposition, with the prefix Bayes risks acting as the stage-wise decision costs.

Remark 2 (Admissible design-policy class Π_d)**.** *The Bellman principle need not hold for an arbitrary parametrized class Π_d. In particular, with parameter sharing across horizons, the continuation after h_t cannot generally be optimized independently of the policy at other steps. We therefore define the value functions over admissible* continuation policies*. The standard Bellman recursion applies whenever the policy class is closed under continuation and splicing.*

B.2 TERMINAL OBJECTIVES AND SELF-CONSISTENCY

Consider now the terminal problem obtained by placing value only on the final decision,

$$J_T(\pi_d) := \mathrm{EFL}^{B,T}_{\ell_T}(\pi_d) = \mathbb{E}_{p(h_T;\pi_d)} \left[B_T(h_T) \right]. \tag{27}$$

Continuation values. The optimal continuation value is given by

$$V_t^\star(h_t) := \inf_{\pi_{d,t+1:T}} \mathbb{E}_{p(h_{t+1:T}|h_t;\pi_{d,t+1:T})} \left[B_T(h_T) \right], \qquad V_T^\star = B_T, \tag{28}$$

where the infimum is taken over admissible continuation policies. In particular, for the fixed initial history h_0,

$$V_0^\star(h_0) = J_T^\star := \inf_{\pi_d \in \Pi_d} J_T(\pi_d) \tag{29}$$

The continuation values satisfy the Bellman recursion

$$V_{t-1}^\star(h_{t-1}) = \inf_{\xi_t \in \Xi_t} \mathbb{E}_{p(h_t|h_{t-1},\xi_t)} \left[V_t^\star(h_t) \right]. \tag{30}$$

Equation (30) gives the usual self-consistency property of an optimal sequential design policy. In particular, an optimal policy may be chosen such that, at every history reached under that policy,

$$\pi_d^B(h_{t-1}) \in \operatorname*{argmin}_{\xi_t \in \Xi_t} \mathbb{E}_{p(h_t|h_{t-1},\xi_t)} \left[V_t^\star(h_t) \right]. \tag{31}$$

Thus, once the optimal continuation value is known, the current design can be selected locally: all consequences of future observations and future optimal designs have already been absorbed into $V_t^\star$.

Decomposition of the terminal objective into a sum of Bellman gaps. This also yields an exact decomposition of terminal suboptimality into local Bellman errors. Define the Bellman gap

$$\Delta_t^\star(h_{t-1}, \xi_t) := \mathbb{E}_{p(h_t|h_{t-1},\xi_t)} \left[V_t^\star(h_t) \right] - V_{t-1}^\star(h_{t-1}). \tag{32}$$

By (30),

$$\Delta_t^\star(h_{t-1}, \xi_t) \geq 0, \tag{33}$$

with equality if and only if ξ_t is Bellman-optimal at h_{t-1}.

Proposition 4 (Terminal suboptimality as accumulated Bellman gaps)**.** *For any admissible design policy π_d,*

$$J_T(\pi_d) - J_T^\star = \mathbb{E}_{p(h_T;\pi_d)} \left[\sum_{t=1}^{T} \Delta_t^\star(h_{t-1}, \xi_t) \right]. \tag{34}$$

Hence, terminal suboptimality is exactly the expected accumulation of the local Bellman suboptimalities incurred along the acquisition trajectory.

Proof. By the tower property,

$$\mathbb{E}_{p(h_T;\pi_d)} \left[\Delta_t^\star(h_{t-1}, \xi_t) \right] = \mathbb{E}_{p(h_T;\pi_d)} \left[V_t^\star(h_t) - V_{t-1}^\star(h_{t-1}) \right]. \tag{35}$$

Summing over t gives a telescoping sum,

$$\mathbb{E}_{p(h_T;\pi_d)} \left[\sum_{t=1}^{T} \Delta_t^\star(h_{t-1}, \xi_t) \right] = \mathbb{E}_{p(h_T;\pi_d)} \left[V_T^\star(h_T) \right] - V_0^\star(h_0).$$

Using $V_T^\star = B_T$ and $V_0^\star(h_0) = J_T^\star$ yields (34). □

B.3 Bellman-consistent intermediate losses

The Bellman recursion also yields a stronger way to densify the terminal objective than the telescoping construction above. In particular, one can associate each intermediate history with a loss whose local action is the *next design decision*, and whose conditional Bayes risk directly measures the excess terminal cost induced by that decision.

Let (π_d^B, π_a^B) be a Bayes-optimal design and terminal-action policy for the terminal problem, and define the realized terminal loss under the Bayes-optimal terminal decision by

$$L_T^B(\theta, h_T) := \ell_T(\pi_a^B(h_T), \theta, h_T).$$

For $t < T$, consider committing to a candidate next design ξ_{t+1} at history h_t, and following π_d^B thereafter (where π_d^B is defined to make conditionally Bayes-optimal decisions given the current beliefs at each step in a recursive fashion, noting that formally the original π_d^B is only defined "on path" where previous decisions were also Bayes-optimal, so we need to expand the definition to be future Bayes-optimal when previous decisions may not have been Bayes-optimal). Define the corresponding θ-conditional terminal cost of that design decision as

$$\bar{\ell}_t^B(\xi_{t+1}, \theta, h_t) := \mathbb{E}_{p(h_T|\theta,h_t,\xi_{t+1};\pi_d^B)} \left[L_T^B(\theta, h_T) \right]. \tag{36}$$

Averaging over the current posterior gives the usual optimal Q-function,

$$Q_t^\star(h_t, \xi_{t+1}) := \mathbb{E}_{p(\theta|h_t)} \left[\bar{\ell}_t^B(\xi_{t+1}, \theta, h_t) \right]. \tag{37}$$

By the Bellman principle,

$$V_t^\star(h_t) = \inf_{\xi_{t+1} \in \Xi_{t+1}} Q_t^\star(h_t, \xi_{t+1}), \tag{38}$$

and a Bayes-optimal policy may be chosen such that

$$\pi_d^B(h_t) \in \operatorname*{argmin}_{\xi_{t+1}} Q_t^\star(h_t, \xi_{t+1}). \tag{39}$$

We can therefore define, for $t < T$, the Bellman-consistent intermediate losses

$$\ell_t^{\mathrm{BC}}(\xi_{t+1}, \theta, h_t) := \bar{\ell}_t^B(\xi_{t+1}, \theta, h_t) - V_t^\star(h_t). \tag{40}$$

Although these losses need not be pointwise non-negative for every θ, their expectations over the posterior, i.e. conditional Bayes risks, are always non-negative:

$$\mathbb{E}_{p(\theta|h_t)}\left[\ell_t^{\mathrm{BC}}(\xi_{t+1}, \theta, h_t)\right] = Q_t^\star(h_t, \xi_{t+1}) - V_t^\star(h_t) \tag{41}$$

$$=: A_t^\star(h_t, \xi_{t+1}) \geq 0, \tag{42}$$

with equality if and only if ξ_{t+1} is Bellman-optimal at h_t. Thus, $A_t^\star$ is the Bellman advantage, or equivalently the local excess terminal risk, associated with the candidate next design.

Note here that we immediately have $\forall t \in \{1, \cdots, T-1\}$:

$$\pi_d^B(h_t) \in \operatorname*{argmin}_{\xi_{t+1}} Q_t^\star(h_t, \xi_{t+1}) = \operatorname*{argmin}_{\xi_{t+1} \in \Xi_{t+1}} \mathbb{E}_{p(\theta|h_t)}\left[\ell_t^{\mathrm{BC}}(\xi_{t+1}, \theta, h_t)\right] \tag{43}$$

because $V_t^\star(h_t)$ is independent of ξ_{t+1}, so we have proven the second part of Theorem 1 from the main paper.

We are now also ready to more formally state, and prove, the first part of Theorem 1 by proving the multistep and terminal objectives yield the same optimal policies as follows.

Theorem 2 (Bellman-consistent intermediate losses, full statement of Theorem 1)**.** *Assume the set of possible design-policies is closed under continuation and splicing (i.e. it is sufficiently expressive that the set of possible piecewise defined policies is the same as that of end-to-end policies), and that the relevant expectations and measurable minimizers exist. Let (π_d^B, π_a^B) be Bayes-optimal for the terminal problem, such that*

$$(\pi_d^B, \pi_a^B) \in \operatorname*{argmin}_{\pi_d \in \Pi_d, \pi_a \in \Pi_a} \mathbb{E}_{p(\theta, h_T; \pi_d)}\left[\ell_T(\pi_a(h_T), \theta, h_T)\right].$$

For $t = 1, \ldots, T-1$, let ℓ_t^{BC} be defined by (40), and retain the original Bayes-optimal terminal loss

$$\ell_T^{\mathrm{BC}}(\theta, h_T) := L_T^B(\theta, h_T) = \ell_T(\pi_a^B(h_T), \theta, h_T).$$

For any temporal weights $\lambda \in \mathbb{R}_+^T$ with $\lambda_T > 0$, define the corresponding design-stage multistep objective

$$J_\lambda^{\mathrm{BC}}(\pi_d) := \mathbb{E}_{p(\theta, h_T; \pi_d)}\left[\sum_{t=0}^{T-1} \lambda_t \ell_t^{\mathrm{BC}}\big(\pi_d(h_t), \theta, h_t\big) + \lambda_T L_T^B(\theta, h_T)\right]. \tag{44}$$

Then

$$\operatorname*{argmin}_{\pi_d \in \Pi_d} J_\lambda^{\mathrm{BC}}(\pi_d) = \operatorname*{argmin}_{\pi_d \in \Pi_d} J_T(\pi_d), \tag{45}$$

$$\text{where} \quad J_T(\pi_d) = \inf_{\pi_a \in \Pi_a} \mathbb{E}_{p(\theta, h_T; \pi_d)}\left[\ell_T(\pi_a(h_T), \theta, h_T)\right] = \mathbb{E}_{p(\theta, h_T; \pi_d)}\left[L_T^B(\theta, h_T)\right]. \tag{46}$$

In particular, π_d^B is Bayes-optimal for J_λ^{BC} for every such choice of temporal weights.

Proof. For any design policy π_d, conditioning on the history h_t and using (42) gives

$$\mathbb{E}_{p(\theta, h_T; \pi_d)}\left[\ell_t^{\mathrm{BC}}\big(\pi_d(h_t), \theta, h_t\big) \mid h_t\right] = Q_t^\star\big(h_t, \pi_d(h_t)\big) - V_t^\star(h_t) =: A_t^\star\big(h_t, \pi_d(h_t)\big) \geq 0.$$

Hence every intermediate term has non-negative expectation under any admissible policy. It follows that for any π_d,

$$J_\lambda^{\mathrm{BC}}(\pi_d) \geq \lambda_T J_T(\pi_d) \geq \lambda_T J_T^\star, \tag{47}$$

where $J_T^\star = \inf_{\pi_d} J_T(\pi_d)$. Now let π_d^B be any terminal-optimal design policy, such that $J_T(\pi_d^B) = J_T^\star$. By Proposition 4,

$$J_T(\pi_d^B) - J_T^\star = \mathbb{E}_{p(h_T; \pi_d^B)}\left[\sum_{s=1}^{T} \Delta_s^\star(h_{s-1}, \xi_s)\right] = 0,$$

because $J_T(\pi_d^B) = J_T^\star$. Since every Bellman gap is non-negative, each of them must have zero expectation under the terminal-optimal policy π_d^B. Moreover,

$$A_t^\star(h_t, \xi_{t+1}) = \Delta_{t+1}^\star(h_t, \xi_{t+1}),$$

so every intermediate Bellman-consistent loss in (44) also has zero expectation. Therefore,

$$J_\lambda^{\mathrm{BC}}(\pi_d^B) = \lambda_T J_T^\star$$

for every terminal-optimal π_d^B. Conversely, suppose that π_d minimizes J_λ^{BC}. Since a terminal-optimal policy achieves $\lambda_T J_T^\star$, the minimum value of J_λ^{BC} is $\lambda_T J_T^\star$. Together with (47) and $\lambda_T > 0$, equality requires

$$J_T(\pi_d) = J_T^\star.$$

Thus every minimizer of the shaped objective is also terminal-optimal, and every terminal-optimal policy minimizes the shaped objective, proving (45). □

Interpretation. The result differs from a purely telescoping potential-based construction. The intermediate losses need not sum pathwise to the terminal loss, nor do their temporal weights need to be uniform. Instead, each loss is individually Bellman-consistent: after averaging over the current posterior, it is exactly the non-negative excess terminal risk associated with the next design decision. Hence arbitrary non-negative reweighting of these auxiliary terms cannot create a policy that improves upon the terminal optimum, while every terminal-optimal policy incurs zero expected auxiliary cost.

The construction is primarily structural rather than directly computational. Evaluating $\bar{\ell}_t^B$ requires access to an optimal continuation policy, or equivalently to the terminal-optimal Q- and value functions, which are generally intractable in the BED problems of interest. Practical multistep losses can therefore be interpreted as tractable surrogates for these ideal Bellman-consistent local objectives. Such surrogates need not preserve the original terminal problem exactly, but can provide intermediate supervision that approximates terminal progress while substantially shortening the optimization paths through long acquisition rollouts.

C Pathwise Gradients and Optimization Properties of Multistep Supervision

This appendix develops the optimization properties of multistep supervision discussed in Section 4.3. We first derive the exact pathwise-gradient decomposition of the multistep objective and make explicit the temporal paths through which each design decision receives supervision. We then characterize how the length of these paths controls sensitivity through products of rollout Jacobians, before studying when auxiliary multistep gradients improve the signal-to-noise ratio in the terminal descent direction.

Throughout this section, the true deployment objective is assumed to be terminal,

$$\mathcal{J}_T(\phi,\psi) := \mathbb{E}_{p(\theta)p(h_T\mid\theta;\pi_d^\phi)}\left[\ell_T\left(\pi_{a,T}^\psi(h_T),\theta,h_T\right)\right], \tag{48}$$

while intermediate losses are introduced only as auxiliary training signals.

C.1 Differentiable rollout representation

We assume the observation model $p(y \mid \theta,\xi)$ satisfies the reparameterization assumption, implying that, for every $\xi \in \Xi$ and $\theta \in \Theta$, there exists a function g such that,

$$y = g(\theta,\xi,\varepsilon), \qquad \varepsilon \sim p_\varepsilon, \tag{49}$$

where the distribution of ε is independent of the policy parameters. Let

$$\zeta := (\theta, \varepsilon_{1:T})$$

collect the exogenous randomness of a rollout. For the purpose of differentiation, let s_t denote a finite-dimensional differentiable representation of the information contained in h_t. The entire acquisition step can then be represented recursively as

$$s_t = F_t(s_{t-1},\theta,\varepsilon_t;\phi), \qquad t = 1,\dots,T, \tag{50}$$

where F_t includes both the design decision $\xi_t = \pi_d^\phi(h_{t-1})$ and the reparameterized observation (49), followed by the update from h_{t-1} to h_t. This notation is only introduced to make the computational graph explicit and does not impose any additional Markov assumption: s_t may simply contain the complete history h_t. We also assume that the initial representation s_0 is independent of ϕ.

Then, we assume that the acquisition-policy parameters ϕ and the action-policy parameters ψ are disjoint. We can therefore define the realized loss at horizon t by

$$c_t(s_t,\theta;\psi) := \ell_t\left(\pi_{a,t}^\psi(h_t),\theta,h_t\right). \tag{51}$$

For fixed temporal weights λ, the sample-level multistep loss is therefore

$$\widehat{\mathcal{J}}_\lambda(\phi,\psi;\zeta) := \sum_{t=1}^{T}\lambda_t c_t(s_t,\theta;\psi), \tag{52}$$

with population objective

$$\mathcal{J}_\lambda(\phi,\psi) := \mathbb{E}_\zeta\left[\widehat{\mathcal{J}}_\lambda(\phi,\psi;\zeta)\right]. \tag{53}$$

We assume throughout that the relevant maps are differentiable almost surely and satisfy sufficient integrability conditions to interchange differentiation and expectation.

C.2 Pathwise-gradient decomposition

We first make explicit how losses at different horizons propagate gradients back to earlier acquisition decisions. Define

$$J_t := \frac{\partial s_t}{\partial s_{t-1}}, \qquad B_t := \left.\frac{\partial s_t}{\partial \phi}\right|_{s_{t-1}}, \tag{54}$$

and let

$$q_t := \nabla_{s_t} c_t(s_t, \theta; \psi). \tag{55}$$

Here, B_t measures the direct sensitivity of the t-th acquisition step to the acquisition-policy parameters, whereas J_t propagates perturbations of the history through one additional acquisition step. For $i \le t$, define the rollout Jacobian from step i to step t as

$$P_{i\to t} := \begin{cases} J_t J_{t-1} \cdots J_{i+1}, & i < t, \\ I, & i = t. \end{cases} \tag{56}$$

Proposition 5 (Pathwise gradient decomposition)**.** *Under the differentiability and integrability conditions above,*

$$\nabla_\phi \mathcal{J}_\lambda = \mathbb{E}_\zeta \left[\sum_{i=1}^{T} \sum_{t=i}^{T} \lambda_t B_i^\top P_{i\to t}^\top q_t \right]. \tag{57}$$

Moreover, since the downstream actions do not affect subsequent data acquisition,

$$\nabla_\psi \mathcal{J}_\lambda = \mathbb{E}_\zeta \left[\sum_{t=1}^{T} \lambda_t \nabla_\psi c_t(s_t, \theta; \psi) \right]. \tag{58}$$

Proof. Differentiating the recursion (50) gives

$$\frac{\mathrm{d}s_t}{\mathrm{d}\phi} = J_t \frac{\mathrm{d}s_{t-1}}{\mathrm{d}\phi} + B_t. \tag{59}$$

Iterating this recursion yields

$$\frac{\mathrm{d}s_t}{\mathrm{d}\phi} = \sum_{i=1}^{t} P_{i\to t} B_i. \tag{60}$$

Hence,

$$\begin{aligned} \nabla_\phi c_t &= \left(\frac{\mathrm{d}s_t}{\mathrm{d}\phi} \right)^\top q_t \\ &= \sum_{i=1}^{t} B_i^\top P_{i\to t}^\top q_t. \end{aligned} \tag{61}$$

Using linearity of differentiation and expectation,

$$\begin{aligned} \nabla_\phi \mathcal{J}_\lambda &= \mathbb{E}_\zeta \left[\sum_{t=1}^{T} \lambda_t \nabla_\phi c_t \right] \\ &= \mathbb{E}_\zeta \left[\sum_{t=1}^{T} \lambda_t \sum_{i=1}^{t} B_i^\top P_{i\to t}^\top q_t \right], \end{aligned} \tag{62}$$

which is equivalent to (57) after exchanging the two finite sums. The action-policy parameters ψ do not enter the rollout recursion (50). Therefore each c_t depends on ψ only through the causal action $\pi_{a,t}^{\psi}(h_t)$, which immediately gives (58). □

Proposition 5 makes the difference between terminal and multistep supervision explicit. Under terminal-only training, $\lambda_T = 1$ and $\lambda_t = 0$ for $t < T$, so

$$\nabla_\phi \mathcal{J}_T = \mathbb{E} \left[\sum_{i=1}^{T} B_i^\top P_{i\to T}^\top q_T \right]. \tag{63}$$

Thus, an acquisition decision made at an early step i is supervised only through a path containing $T - i$ successive rollout Jacobians. By contrast, under multistep supervision its contribution is

$$\mathbb{E} \left[\sum_{t=i}^{T} \lambda_t B_i^\top P_{i\to t}^\top q_t \right], \tag{64}$$

which contains paths of lengths $\{0, 1, \ldots, T - i\}$ whenever the corresponding λ_t are non-zero. Multistep supervision therefore does not merely duplicate the terminal signal: it introduces additional supervision paths terminating at earlier prefixes of the trajectory.

C.3 PATH LENGTH AND GRADIENT SENSITIVITY

The gradient decomposition above highlights a fundamental difference between terminal and multistep supervision. The contribution of the loss at horizon t to the acquisition parameters acting at an earlier step i is

$$G_{t,i} := B_i^\top P_{i\to t}^\top q_t, \qquad P_{i\to t} = J_t J_{t-1} \cdots J_{i+1}.$$

Hence, temporal credit assignment involves products of successive rollout Jacobians. As in other long computational graphs, this multiplicative structure can make gradient propagation increasingly sensitive to path length: depending on the rollout dynamics, long paths may attenuate or amplify the backpropagated signal. The following simple regime makes this dependence explicit.

Proposition 6 (Sensitivity along temporal gradient paths)**.** *Suppose that, almost surely,*

$$m_k \le \sigma_{\min}(J_k) \le \|J_k\|_{\mathrm{op}} \le M_k, \qquad k = 1, \dots, T. \tag{65}$$

Then, for every $i < t$,

$$\prod_{k=i+1}^{t} m_k \le \sigma_{\min}(P_{i\to t}) \le \|P_{i\to t}\|_{\mathrm{op}} \le \prod_{k=i+1}^{t} M_k. \tag{66}$$

Consequently, if $\|J_k\|_{\mathrm{op}} \le \rho < 1$, the propagated signal can be attenuated as

$$\|P_{i\to t}^\top q_t\| \le \rho^{t-i} \|q_t\|.$$

Conversely, in the illustrative expansive regime $\sigma_{\min}(J_k) \ge \rho > 1$,

$$\|P_{i\to t}^\top q_t\| \ge \rho^{t-i} \|q_t\|.$$

Proof. The upper bound follows from submultiplicativity of the operator norm, $\|AB\|_{\mathrm{op}} \le \|A\|_{\mathrm{op}} \|B\|_{\mathrm{op}}$, which gives

$$\|P_{i\to t}\|_{\mathrm{op}} \le \prod_{k=i+1}^{t} M_k.$$

Similarly, for compatible matrices, $\sigma_{\min}(AB) \ge \sigma_{\min}(A)\sigma_{\min}(B)$, and iteration gives the lower bound. The uniform contraction (*respectively expansion*) statement follows by setting $M_k = \rho$ (*respectively* $m_k = \rho$). □

The proposition describes a deliberately simple and restrictive regime, intended only to isolate the multiplicative effect of composing rollout Jacobians over long temporal paths. In particular, in the illustrative contractive regime, for the directional amplification factor

$$R_{i,t} := \frac{\|P_{i\to t}^\top q_t\|}{\|q_t\|}, \qquad q_t \neq 0,$$

the previous bounds immediately imply that the adjoint signal propagated from horizon t to an earlier step i is attenuated as $R_{i,t} \le \rho^{t-i}$, whereas under the equally idealized expansive condition, it is amplified as $R_{i,t} \ge \rho^{t-i}$. Since

$$G_{t,i} = B_i^\top P_{i\to t}^\top q_t,$$

this attenuation or amplification directly affects the learning signal reaching the acquisition parameters at step i, up to the additional action of $B_i^\top$.

These uniform contraction and expansion assumptions are not intended as realistic descriptions of a general nonlinear neural rollout, nor as necessary conditions for vanishing or exploding gradients. Rather, they provide a transparent regime in which the dependence on temporal path length is explicit. In practice, the realized behavior is directional: it depends on how the particular adjoint vector q_t aligns with the singular directions of the successive Jacobians. The quantity $R_{i,t}$ captures precisely this realized amplification or attenuation and is therefore also a natural empirical diagnostic of gradient propagation along the rollout.

Multistep supervision changes the available gradient paths. For an early acquisition step i, terminal-only supervision provides

$$G_i^{\mathrm{term}} = B_i^\top P_{i\to T}^\top q_T,$$

so that its learning signal must propagate through a path of length $T-i$. By contrast, under multistep supervision,

$$G_i^\lambda = \sum_{t=i}^{T} \lambda_t B_i^\top P_{i\to t}^\top q_t,$$

which introduces gradient paths of all lengths $0, 1, \dots, T-i$. In particular, the local contribution $\lambda_i B_i^\top q_i$ crosses no subsequent rollout Jacobian.

Multistep supervision therefore does more than modify the scalar training objective: it changes the optimization problem by providing additional, shorter routes through which learning signals can reach early acquisition decisions. In a contractive regime, these paths need not inherit the ρ^{T-i} attenuation of the terminal signal. In an expansive regime, the interpretation is slightly different. Multistep supervision does not remove the potentially unstable terminal contribution, but supplements it with contributions that traverse fewer expansive Jacobians and are therefore less exposed to the same multiplicative amplification. Provided that the weighting schedule assigns sufficient mass to these shorter horizons and that the terminal contribution does not dominate the sum, the optimization signal can consequently rely less strongly on a single highly amplified long path. Thus, intermediate supervision can mitigate the dependence of gradient propagation on extreme path lengths, although it does not by itself guarantee protection against exploding gradients; strongly expansive terminal paths may still require explicit stabilization such as gradient clipping or normalization.

Importantly, this does not provide a universal stabilization guarantee. The long terminal path remains present whenever $\lambda_T > 0$, and the different horizon-specific contributions may reinforce, cancel, or interfere with one another. The effect therefore depends on the weighting schedule, the rollout dynamics, and the alignment of the corresponding gradient signals.

Gradient variability. Path length can also affect the stochastic variability of the gradient. Let

$$\widetilde{G}_{t,i} := G_{t,i} - \mathbb{E}[G_{t,i}],$$

and define the scalar variance of a vector gradient contribution by

$$\mathrm{Var}(G_{t,i}) := \mathbb{E}\|\widetilde{G}_{t,i}\|^2.$$

Since $G_{t,i}$ contains the random product $P_{i\to t}$, stochastic amplification or attenuation through a long sequence of Jacobians can also produce highly variable gradient estimates. For the multistep gradient,

$$G_i^\lambda = \sum_{t=i}^{T} \lambda_t G_{t,i},$$

its variance satisfies

$$\mathrm{Var}(G_i^\lambda) = \sum_{s,t=i}^{T} \lambda_s \lambda_t \mathbb{E}\Big[\Big\langle \widetilde{G}_{s,i}, \widetilde{G}_{t,i} \Big\rangle\Big].$$

Thus, shorter paths may reduce the multiplicative stochastic sensitivity of individual contributions, but multistep supervision does not necessarily reduce the variance of their sum: the result also depends on cross-horizon covariances and gradient alignment. We investigate these interactions separately below.

C.4 Alignment with the terminal objective

The preceding analysis concerns the length and conditioning of gradient paths with respect to the design-policy parameters ϕ, for which temporal credit assignment arises. However, shorter paths are useful only if the corresponding gradients remain relevant to the terminal task. We therefore separate the terminal and auxiliary objectives. We note

$$\mathcal{J}_A := \sum_{t<T} \lambda_t \mathbb{E}\left[\ell_t\left(\pi_{a,t}^{\psi}(h_t), \theta, h_t\right)\right], \tag{67}$$

and consider

$$\mathcal{J}_\beta := \mathcal{J}_T + \beta \mathcal{J}_A, \qquad \beta \geq 0. \tag{68}$$

We focus on gradients with respect to the design-policy parameters and let

$$g_T := \nabla_\phi \mathcal{J}_T, \qquad g_A := \nabla_\phi \mathcal{J}_A.$$

Proposition 7 (Terminal descent under auxiliary supervision)**.** *Assume $g_T \neq 0$. The mean multistep gradient with respect to ϕ,*

$$g_\beta = g_T + \beta g_A,$$

defines a strict first-order descent direction for the terminal objective if and only if

$$\|g_T\|^2 + \beta \langle g_T, g_A \rangle > 0. \tag{69}$$

In particular, if $\langle g_T, g_A \rangle \geq 0$, every $\beta \geq 0$ satisfies the first-order alignment condition. If $\langle g_T, g_A \rangle < 0$, it is sufficient and necessary that

$$\beta < \frac{\|g_T\|^2}{-\langle g_T, g_A \rangle}. \tag{70}$$

Proof. For a sufficiently small step size $\eta > 0$, a first-order Taylor expansion of the terminal objective along the design-policy update $-g_\beta$ gives

$$\mathcal{J}_T(\phi - \eta g_\beta) = \mathcal{J}_T(\phi) - \eta \langle g_T, g_\beta \rangle + o(\eta).$$

Hence $-g_\beta$ is a strict first-order descent direction exactly when $\langle g_T, g_\beta \rangle > 0$. Using $g_\beta = g_T + \beta g_A$ gives

$$\langle g_T, g_\beta \rangle = \|g_T\|^2 + \beta \langle g_T, g_A \rangle,$$

from which both cases follow. □

This proposition makes explicit why multistep supervision cannot provide a universal optimization improvement for the design policy: auxiliary gradients can conflict with the terminal objective. Their usefulness depends jointly on the conditioning benefit of shorter credit-assignment paths and on their alignment with the terminal gradient in the design-policy parameter space.

C.5 Terminal-direction signal-to-noise ratio

We now account for stochastic gradient noise in the design-policy parameters ϕ. Let G_T be an unbiased stochastic gradient estimator of the terminal objective and G_A an unbiased stochastic gradient estimator of the auxiliary objective,

$$\mathbb{E}[G_T] = g_T, \qquad \mathbb{E}[G_A] = g_A.$$

The two estimators may be computed from the same simulated trajectory and are therefore generally correlated. Define

$$G_\beta := G_T + \beta G_A. \tag{71}$$

Whenever $g_T \neq 0$, let

$$u := \frac{g_T}{\|g_T\|} \tag{72}$$

be the unit terminal-gradient direction, and define

$$\mathrm{SNR}_T(G) := \frac{\left(\mathbb{E}[u^\top G]\right)^2}{\mathrm{Var}(u^\top G)}. \tag{73}$$

Set

$$\mu_T := \mathbb{E}[u^\top G_T] = \|g_T\|, \tag{74}$$

$$\mu_A := \mathbb{E}[u^\top G_A] = u^\top g_A, \tag{75}$$

$$V_T := \mathrm{Var}(u^\top G_T), \tag{76}$$

$$V_A := \mathrm{Var}(u^\top G_A), \tag{77}$$

$$C := \mathrm{Cov}\left(u^\top G_T, u^\top G_A\right). \tag{78}$$

Then

$$\mathrm{SNR}_T(G_\beta) = \frac{(\mu_T + \beta \mu_A)^2}{V_T + 2\beta C + \beta^2 V_A}. \tag{79}$$

Proposition 8 (Local improvement of terminal-direction gradient SNR). *Assume* $\mu_T > 0$ *and* $V_T > 0$. *Then*

$$\frac{\mathrm{d}}{\mathrm{d}\beta} \log \mathrm{SNR}_T(G_\beta)\Big|_{\beta=0} = 2\left(\frac{\mu_A}{\mu_T} - \frac{C}{V_T}\right). \tag{80}$$

Consequently, sufficiently small positive auxiliary weight improves terminal-direction SNR whenever

$$\frac{\mu_A}{\mu_T} > \frac{C}{V_T}. \tag{81}$$

Proof. From (79),

$$\log \mathrm{SNR}_T(G_\beta) = 2\log(\mu_T + \beta\mu_A) - \log(V_T + 2\beta C + \beta^2 V_A). \tag{82}$$

Differentiating gives

$$\frac{\mathrm{d}}{\mathrm{d}\beta} \log \mathrm{SNR}_T(G_\beta) = \frac{2\mu_A}{\mu_T + \beta\mu_A} - \frac{2C + 2\beta V_A}{V_T + 2\beta C + \beta^2 V_A}. \tag{83}$$

Evaluating at $\beta = 0$ yields

$$2\left(\frac{\mu_A}{\mu_T} - \frac{C}{V_T}\right).$$

A strictly positive derivative implies that $\mathrm{SNR}_T(G_\beta) > \mathrm{SNR}_T(G_T)$ for all sufficiently small $\beta > 0$. □

The condition admits a direct interpretation. The quantity μ_A measures the mean auxiliary gradient projected onto the terminally useful direction, whereas C measures the amount of stochastic variation that the auxiliary gradient shares with the terminal gradient in that same direction. Auxiliary supervision improves local terminal-direction SNR when the additional aligned signal is sufficiently large relative to this covariance contribution.

For completeness, an exact finite-β comparison follows directly from (79). For $\beta > 0$ such that $\mathrm{Var}(u^\top G_\beta) > 0$, we have $\mathrm{SNR}_T(G_\beta) > \mathrm{SNR}_T(G_T)$ if and only if

$$2\mu_T\big(\mu_A V_T - \mu_T C\big) + \beta\big(\mu_A^2 V_T - \mu_T^2 V_A\big) > 0. \tag{84}$$

The local criterion (81) is precisely the first-order version of this condition. Notice also that when G_T and G_A are estimated independently, $C = 0$. In this special case, any auxiliary estimator satisfying $\mu_A > 0$ locally increases the terminal-direction SNR for sufficiently small β.

Importantly, terminal-direction SNR is not by itself a measure of whether the resulting gradient is useful for terminal optimization. Its numerator is squared, so a stochastic gradient whose mean projection is large but points strongly against the terminal gradient can still have a large SNR. In particular, at finite β, a meaningful optimization interpretation additionally requires the alignment condition

$$\mathbb{E}[u^\top G_\beta] = \mu_T + \beta\mu_A > 0,$$

which is equivalent to $\langle g_T, g_\beta \rangle > 0$ and therefore to the first-order terminal-descent condition of Proposition 7. The SNR should thus be interpreted jointly with gradient alignment: it quantifies the reliability of the signal along the terminal direction, but not whether that signal has the correct sign.

The preceding results isolate three complementary effects of multistep supervision for long-horizon terminal problems. First, Proposition 5 shows that a terminal loss supervises an early acquisition decision only through the full remaining rollout, whereas multistep losses introduce additional paths ending at intermediate horizons. Second, Proposition 6 shows that the sensitivity carried by these paths is governed by products of rollout Jacobians, whose conditioning can deteriorate multiplicatively with temporal depth. Intermediate losses therefore provide structurally shorter alternatives to the terminal credit-assignment path. Third, Propositions 7 and 8 show that shorter paths are beneficial only when the resulting auxiliary gradient remains sufficiently aligned with the terminal task and contributes useful signal relative to its additional stochastic variation.

D ARCHITECTURE

Throughout the paper, we parameterize the design and downstream action policies by differentiable neural networks, denoted respectively by π_d^ϕ and π_a^ψ. The specific architecture can be chosen according to the structure of the experimental problem, but a common choice is an encoder–aggregator–emitter decomposition, in which past design–outcome pairs are first embedded locally, then combined into a representation of the acquisition history, and finally mapped to the next design or downstream action. The aggregation mechanism can be adapted to the statistical structure of the task: permutation-invariant architectures such as DeepSets with symmetric pooling are natural for exchangeable histories, recurrent networks such as LSTMs can encode ordered or non-exchangeable trajectories, while Transformer-based aggregators provide a flexible attention-based alternative. The action policy can follow the same general construction, possibly with shared encoders or history representations to encourage a common representation between acquisition and decision-making, or instead use a simpler task-specific network when the downstream mapping is sufficiently direct. Our framework does not depend on any particular architectural choice, provided that the resulting computation graph remains differentiable and compatible with pathwise optimization.

Design policy architecture. For all experiments, we use the same TNP-like design-policy architecture as in Rossa et al. (2026); see Appendix E therein for full details. Briefly, each past design–outcome pair (ξ_t, y_t) is encoded using separate MLPs and fused into a latent history token. The resulting sequence is processed by a Transformer-based history aggregator augmented with learned temporal embeddings and a decision token, whose final representation summarizes the information relevant for selecting the next acquisition. An MLP emitter then maps this representation to the next design ξ_{t+1}. In our implementation, the history aggregator uses two Transformer encoder layers with four attention heads and feedforward dimension 128, while the design, outcome, fusion, and emitter modules are lightweight MLPs. This architecture is kept fixed across the long-horizon experiments, so that differences in performance arise from the training objective rather than from the design-policy parameterization.

Multistep downstream action network. To produce a downstream prediction after every acquisition step, we use a simple shared MLP acting on fixed-size representations of the observed history; see Table 2 for additional details. Given a trajectory prefix $h_t = \{(\xi_s, y_s)\}_{s=1}^t$, $t \le T$, we first concatenate each design–observation pair, $z_s = [\xi_s; y_s] \in \mathbb{R}^{d_\xi + d_y}$, and construct a fixed-length representation by padding all unobserved future entries with zeros:

$$x_t = \mathrm{vec}\,(z_1, \dots, z_t, 0, \dots, 0) \in \mathbb{R}^{T(d_\xi + d_y)}. \tag{85}$$

The same downstream network f_ψ is then applied independently to every prefix,

$$\hat{a}_t = f_\psi(x_t), \qquad t = 1, \dots, T, \tag{86}$$

yielding a sequence of predictions $(\hat{a}_1, \dots, \hat{a}_T) \in \mathbb{R}^{T \times d_a}$, where d_a denotes the dimension of the downstream action. Importantly, the network parameters are shared across all acquisition steps: the model therefore learns a single mapping from partially observed, zero-padded histories to downstream actions, rather than using a different prediction head at each horizon. During multistep training, the prediction $\hat{a}_t$ is evaluated using the downstream loss at every supervised prefix. At test time, the same architecture can either return the complete sequence of intermediate decisions or only the final action $\hat{a}_T$. This architecture intentionally remains lightweight. It does not maintain a recurrent hidden state or explicitly aggregate the history through attention; temporal position is instead encoded implicitly by the location of each design–observation pair in the flattened, zero-padded input. This makes the downstream predictor inexpensive and ensures that improvements obtained with multistep supervision are not confounded with the introduction of a substantially more expressive action architecture.

Table 2: Architecture of the multistep downstream action network.

Component	**Dimension**	**Description**
Design–observation pair	$d_\xi + d_y$	$z_t = [\xi_t; y_t]$
Padded history	$T \times (d_\xi + d_y)$	$(z_1, \dots, z_t, 0, \dots, 0)$
Flattened input	$T(d_\xi + d_y)$	Input to the shared MLP
Hidden layers	$(h_1, \dots, h_L)$	Fully connected layers
Output	d_a	Downstream action $\hat{a}_t$

Table 3 summarizes the hyperparameters of the action-policy network across all tasks considered in this paper. The hidden architecture is kept fixed across experiments, while the input/output dimensions and final output transformation are adapted to the downstream task.

Table 3: Hyperparameters of the downstream MLP used in the experiments.

Parameter	**Value**
Number of hidden layers	3
Hidden dimensions	$(512, 256, 128)$
Hidden activation	`GELU`
Output activation	Task dependent
Input dimension	$T(d_\xi + d_y)$
Output dimension	d_a
Parameter sharing across steps	Yes

Alternative architectures and representation sharing. The downstream predictor could naturally be made substantially more expressive. In particular, since the design policy π_d^ϕ already constructs learned representations of the experimental history through its encoder and aggregation modules, the action policy could reuse part or all of this backbone. For example, the two policies could share the design and observation encoders, a common pair representation, or even the complete history aggregator, while retaining separate task-specific output heads. Such weight sharing could provide a richer representation to the downstream predictor, reduce redundant computation, and strengthen the coupling between acquisition and decision-making by allowing both objectives to shape a common latent representation. More generally, the multistep action policy itself could adopt the same encoder–aggregator–emitter architecture as the design policy, using DeepSets, recurrent networks, or Transformer-based aggregation depending on the structure of the experimental history.

We deliberately do not explore these architectural extensions here. Our objective is not to maximize absolute downstream predictive performance, but to isolate the effect of the proposed multistep training objective under a simple and controlled action-policy parameterization. This choice also facilitates comparison with ACTION-BED, whose terminal downstream policy is likewise implemented using a relatively simple MLP. Using a substantially richer multistep action network or sharing the design-policy backbone would introduce an additional source of performance gain and would therefore make it harder to attribute differences specifically to multistep supervision. Investigating shared representations and more expressive multistep action architectures is left for future work.

E EXPERIMENTAL DETAILS

For each experimental task, we provide a detailed description of the experimental setting, including the data-generating process, the considered objectives and loss functions, the architectures of all baseline methods, the training hyperparameters, and all additional details required to reproduce our results. Whenever possible, we make only minimal modifications to the original baseline implementations, so as to preserve their intended training procedures and ensure a fair comparison with MA-BED. We further report additional ablations and experimental results to better characterize and contextualize the empirical findings presented in the main paper.

All experiments were implemented in PyTorch (Paszke et al., 2019), and all models were trained on the same compute server equipped with an NVIDIA H100 GPU with 80GB of memory (Hopper architecture) and 64 CPU cores. Each training run was performed on a single GPU, ensuring that comparisons of computational cost are not confounded by differences in hardware or parallelization.

E.1 DYNAMIC TRACKING

We consider a sequential tracking problem in which an adaptive sensor must localize a moving target throughout a deployment horizon of length T. In contrast to classical tasks, the latent quantity of interest is no longer static but evolving between acquisitions according to stochastic dynamics. The objective is therefore inherently sequential, as accurate decisions are required throughout the trajectory rather than only after the final observation.

Target dynamics. The target evolves in a two-dimensional domain $\mathcal{X} \subset \mathbb{R}^2$. Its latent state at time t is $s_t = (x_t, v_t, m_t)$, where $x_t \in \mathbb{R}^2$ denotes its position, $v_t \in \mathbb{R}^2$ its velocity, and $m_t \in \{1, \dots, M\}$ its current motion regime. Each episode is initialized from latent parameters

$$\theta = \big(x_0, v_0, m_0, p_{\text{switch}}, \sigma_{\text{dyn}}\big),$$

with independent priors

$$\begin{aligned} x_0 &\sim \mathcal{N}(\mu_x, \Sigma_x), & v_0 &\sim \mathcal{N}(\mu_v, \Sigma_v), \\ m_0 &\sim \text{Categorical}(\pi), & p_{\text{switch}} &\sim \text{Beta}(\alpha_s, \beta_s), \\ \sigma_{\text{dyn}} &\sim \text{LogNormal}(\mu_\sigma, \sigma_\sigma^2). \end{aligned}$$

We use a switching coordinated-turn model with three motion regimes corresponding to straight motion, left turns, and right turns,

$$\omega_1 = 0, \qquad \omega_2 = +\omega_0, \qquad \omega_3 = -\omega_0.$$

Conditionally on the current regime, the dynamics are

$$v_{t+1} = R(\omega_{m_t} \Delta t) v_t + \sigma_{\text{dyn}} c_v \varepsilon_t^v, \tag{87}$$
$$x_{t+1} = x_t + \Delta t \, v_{t+1} + \sigma_{\text{dyn}} c_x \varepsilon_t^x, \tag{88}$$

where

$$R(\varphi) = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix},$$

and $\varepsilon_t^x, \varepsilon_t^v \sim \mathcal{N}(0, I_2)$ are independent process-noise variables. The motion regime evolves according to

$$m_{t+1} = \begin{cases} m_t, & \text{with probability } 1 - p_{\text{switch}}, \\ m' \neq m_t, & \text{with probability } p_{\text{switch}}, \end{cases}$$

where, conditional on a switch, the new regime is sampled uniformly among the remaining regimes.

Observation model. At each step t, the design policy selects a sensor location $\xi_t \in \mathcal{X}$. The measurement depends on the distance between the sensor and the current target position. We define the noiseless log-intensity as

$$\mu(x_t, \xi_t) = \log\left(b + \frac{1}{m + \|\xi_t - x_t\|_2^2}\right), \tag{89}$$

Table 4: **Dynamic tracking experimental parameters.** Parameters of the target dynamics, latent prior, and observation model.

Parameter	Description	Value
Episode and state space		
T	Acquisition horizon	$\in \{20, 40, 60\}$
$\dim(\mathcal{X})$	Position/design dimension	2
Δt	Time step	1.0
M	Number of motion regimes	3
Initial-state and trajectory prior		
μ_x	Initial-position mean	$(0.5,\ 0.5)$
Σ_x	Initial-position covariance	$0.15^2 I_2$
μ_v	Initial-velocity mean	$(0,\ 0)$
Σ_v	Initial-velocity covariance	$0.08^2 I_2$
π	Initial regime probabilities	$(1/3,\ 1/3,\ 1/3)$
α_s	Beta prior shape for p_{switch}	1
β_s	Beta prior shape for p_{switch}	19
μ_σ	Log-normal location for σ_{dyn}	$\log(10^{-2})$
σ_σ	Log-normal scale for σ_{dyn}	0.5
Motion dynamics		
ω_1	Straight-motion turn rate	0
ω_2	Left-turn rate	$+0.15$
ω_3	Right-turn rate	-0.15
c_x	Position process-noise factor	1.0
c_v	Velocity process-noise factor	1.0
Observation model		
b	Background signal level	0.1
m	Inverse-distance regularization	10^{-4}
σ_y	Observation-noise standard deviation	0.25
$\dim(y_t)$	Observation dimension	1

where $b > 0$ is a background signal level and $m > 0$ prevents singular behavior near the target. The observation is then generated as

$$y_t = \mu(x_t, \xi_t) + \sigma_y \varepsilon_t^y, \qquad \varepsilon_t^y \sim \mathcal{N}(0, 1). \tag{90}$$

Hence, the sensing mechanism is closely related to the source location finding model, while the signal source itself now follows a stochastic, regime-switching trajectory. Table 4 details all the parameters used for this task.

Downstream objective. After observing h_t the downstream action policy produces an estimate of the target's current position, $\hat{x}_t = \pi_a(h_t)$. Prediction accuracy at step t is measured using the squared localization error $\ell_t = \ell_{\text{track}}(x_t, \hat{x}_t) = \|\hat{x}_t - x_t\|_2^2$. The corresponding, uniformly weighted, multistep objective is

$$\ell_{\text{track}}^{\text{MS}}(\hat{x}, x) = \frac{1}{T} \sum_{t=1}^{T} \|\hat{x}_t - x_t\|_2^2. \tag{91}$$

Unlike in static localization, the prediction target itself changes with t. Consequently, each term evaluates a distinct tracking decision using only the experimental information available up to that point. The objective therefore directly measures the ability of the acquisition policy to maintain accurate localization throughout the rollout.

Baselines and experimental protocol. We evaluate the different acquisition strategies on the task of predicting the target location throughout the entire trajectory, considering increasingly long horizons $T \in \{20, 40, 60\}$. Increasing the rollout length progressively makes the tracking problem more challenging: the policy must use a growing sequence of designs and noisy observations to maintain an accurate estimate of the target location as process uncertainty accumulates, while implicitly inferring

Table 5: **Training hyperparameters for the dynamic tracking experiment.**

Hyperparameter	Action-BED	MA-BED
Policy Training		
Batch size	2,000	2,000
Training iterations	2×10^5	2×10^5
Seed	42	42
Policy Optimization		
Optimizer	Adam	Adam
Betas, β	$(0.9, 0.999)$	$(0.9, 0.999)$
Learning rate	5×10^{-4}	5×10^{-4}
Scheduler	ExpLR	ExpLR
Decay	0.95	0.95
Decay period	2,000	2,000
Max. gradient norm (clipping)	1.0	1.0
Action-Policy Retraining		
Retraining iterations	2×10^5	–
Learning rate	5×10^{-4}	–
Betas, β	$(0.9, 0.999)$	–
Max. gradient norm (clipping)	1.0	–

the latent characteristics governing its motion. Importantly, the downstream quantity of interest is here an *evolving latent state*, x_t, rather than a fixed component of θ. Moreover, even exact knowledge of the episode-level parameters θ does not determine the future target trajectory, which additionally depends on stochastic regime transitions and process noise. Consequently, evaluating the likelihood required by standard EIG-based acquisition methods entails marginalizing over the full latent state trajectory, making such approaches particularly cumbersome and poorly matched to this setting; we therefore do not include EIG-optimizing baselines.

We compare MA-BED against a random acquisition policy RANDOM, ACTION-BED, and a particle-filtering approach paired with a learned fixed sensor rule. Each policy-based method is trained using its native objective. To obtain a meaningful random acquisition policy, we sample designs from the time-aggregated prior-predictive distribution of target locations,

$$q_{\text{rand}}(\xi) = \frac{1}{T} \sum_{t=1}^{T} p(x_t = \xi).$$

We approximate q_{rand} empirically by simulating $50{,}000$ trajectories from the prior dynamics and pooling the target positions across all trajectories and time steps. The RANDOM baseline then draws sensor locations from this fixed empirical distribution, independently of the observed history. This ensures that random designs remain concentrated in regions that are plausible under the target dynamics, rather than being sampled uniformly over an arbitrary spatial domain.

Because tracking is inherently online and the downstream decision is itself trajectory-valued, only MA-BED directly optimizes predictive performance across the full acquisition trajectory. To obtain a common evaluation protocol, we subsequently freeze the design policy produced by each method and train the same multistep action network on trajectories generated by that policy, using the resulting sequence of designs and observations to predict the target location at every step. The architecture and training parameters are kept the same for training MA-BED and the separate action network used for the baselines. This separates the quality of the learned acquisition strategy from differences in downstream estimators and allows all methods to be compared through the same per-step tracking loss. Training and evaluation hyperparameters are reported in Table 5.

Oracle particle filter baseline. We compare against a bootstrap particle filter with $N = 65{,}536$ particles. The filter is an "oracle" because it is given privileged information that the other methods lack: the true switching probability and trajectory-noise scale of each episode. The initial state is not revealed and is drawn from the prior. Particles are resampled systematically whenever the effective sample size drops below $N/2$, then jittered to counter degeneracy (Gordon et al., 1993) by adding

Gaussian noise to each particle's position and velocity, with standard deviation 0.3 times the posterior standard deviation. The position estimate at each step is the weighted particle mean. To choose sensor positions, we use a heuristic that circles the filter's predicted target position, advancing a quarter turn each step:

$$\xi_t = \bar{x}_{t|t-1} + \max(0.05,\ 2\,s_t)\left(\cos\tfrac{t\pi}{2},\ \sin\tfrac{t\pi}{2}\right), \tag{92}$$

where $\bar{x}_{t|t-1}$ and $\Sigma_{t|t-1}$ are the weighted mean and covariance of the predicted particle positions, and $s_t = \sqrt{\mathrm{tr}(\Sigma_{t|t-1})/2}$. The sensor therefore moves closer as the belief tightens.

E.2 Constant Elasticity of Substitution (CES)

Setting. At each experimental step $t \in \{1, \dots, T\}$, the participant is presented with a pair of baskets $\xi_t = (\xi_t^{(1)}, \xi_t^{(2)}) \in [0, 100]^6$, and reports their relative preference through a scalar response $y_t \in [0, 1]$. Values close to 0 or 1 indicate a strong preference for one of the two baskets, whereas values near $1/2$ indicate approximate indifference. Following the standard CES benchmark, the latent parameter is $\theta = (\rho, \alpha, u) \in \mathbb{R}^5$, where $\rho \in (0, 1)$ controls the elasticity of substitution between goods, $\alpha = (\alpha_1, \alpha_2, \alpha_3) \in \Delta_3$ represents the participant's relative preferences for the three goods, and $u > 0$ controls the sensitivity of the reported preference to utility differences. The utility assigned to a basket $x = (x_1, x_2, x_3)$ is

$$U_\theta(x) = \left(\sum_{i=1}^{3} \alpha_i x_i^\rho\right)^{1/\rho}.$$

We use the conventional prior

$$\rho \sim \mathrm{Beta}(1, 1), \qquad \alpha \sim \mathrm{Dirichlet}(1, 1, 1), \qquad \log u \sim \mathcal{N}(1, 3).$$

Given a design $\xi = (\xi^{(1)}, \xi^{(2)})$, the participant's response is generated from a noisy latent utility difference,

$$\eta \sim \mathcal{N}\Big(u\big[U_\theta(\xi^{(1)}) - U_\theta(\xi^{(2)})\big],\ u^2\tau^2\big(1 + \|\xi^{(1)} - \xi^{(2)}\|_2\big)^2\Big), \tag{93}$$

which is mapped to the unit interval through a logistic link,

$$y = \mathrm{clip}\left(\sigma(\eta), \varepsilon, 1 - \varepsilon\right), \qquad \sigma(z) = \frac{1}{1 + e^{-z}}$$

The heteroskedastic noise reflects the intuition that comparisons become less reliable when the two baskets are very different, while the logistic transformation converts the latent utility difference into a bounded preference score.

Task objective. Our objective differs from the one traditionally considered for this benchmark. Previous BED formulations primarily seek a sequence of designs that maximizes information about the participant's latent preference parameter θ. Here, data acquisition is ultimately decision-oriented: the collected responses should enable the model not only to identify the participant's preferences, but also to provide useful downstream basket recommendations. In particular, at the beginning of each episode, we independently sample a price vector

$$p = (p_1, p_2, p_3), \qquad p_i \sim \mathrm{Unif}(1, 2),$$

which remains fixed throughout the episode and is provided as additional context to the action policy π_a. Given a fixed budget B, the downstream decision problem is to recommend a basket maximizing the participant's latent utility subject to the budget constraint,

$$a^\star(\theta, p) \in \arg\max_{a \in \mathbb{R}^3_+} U_\theta(a) \qquad \text{s.t.} \qquad p^\top a \le B. \tag{94}$$

Given the history $h_t = \{(\xi_s, y_s)\}_{s=1}^t$ and the episode-specific prices p, the action policy therefore produces a feasible recommendation $a_t = \pi_a(h_t, p)$, under the constraint $p^\top a_t = B$. We measure its performance through the normalized utility regret

$$r_t(\theta, p) = 1 - \frac{U_\theta(a_t)}{U_\theta(a^\star(\theta, p))}.$$

Consequently, the acquisition policy is encouraged to collect information that is specifically relevant to the downstream recommendation problem, rather than information that is merely informative about every component of θ.

We additionally account for participants potentially leaving the experiment before its maximum horizon T. We model this through a constant hazard rate $q \in (0, 1]$. Conditional on remaining in the experiment until step $t < T$, a participant leaves after that interaction with probability q. The resulting effective recommendation horizon H follows a truncated geometric distribution,

$$\Pr(H = t) = \begin{cases} q(1-q)^{t-1}, & t < T, \\ (1-q)^{T-1}, & t = T, \end{cases}$$

where the final probability accounts for all participants who remain in the experiment until its maximal horizon. These probabilities sum to one and directly define the multistep weights $\lambda_t = \Pr(H = t)$. Our training objective is therefore

$$\sum_{t=1}^{T} \lambda_t \, \mathbb{E}[r_t(\theta, p)] = \mathbb{E}[r_H(\theta, p)] \,. \tag{95}$$

Thus, the multistep objective has a direct decision-theoretic interpretation as the expected recommendation regret under a stochastic participation horizon. The acquisition policy must trade off the value of collecting additional preference information against the risk that the participant leaves before subsequent recommendations can be reached, thereby encouraging high-quality recommendations throughout the interaction rather than only at a predetermined terminal horizon.

Table 6: Parameters of the CES task.

Parameter	**Value**
Number of goods	3
Maximal horizon, T	10
τ	0.005
ε	2^{-22}
Budget, B	100
Hazard rate	$q \in \{0.05, 0.1, 0.2, 0.3, 0.4\}$

Baselines and experimental protocol. We compare MA-BED against RANDOM, DAD, and ACTION-BED. Each baseline is first trained according to its native objective: DAD learns an acquisition policy by maximizing the sPCE objective, whereas ACTION-BED jointly learns acquisition and action policies to minimize recommendation regret at the terminal horizon T. RANDOM requires no training and independently samples each design as $\xi_t \sim \mathcal{U}([0, 100]^6)$.

To evaluate all methods under the same stochastic-horizon decision objective, we subsequently freeze each pretrained acquisition policy and, for every hazard rate q considered in the experiment, train a new multistep action policy using the same architecture, objective, and optimization protocol as the action policy of MA-BED. This isolates the decision-relevance of the information collected by each acquisition strategy, because all methods are evaluated through the expected recommendation regret induced by the same distribution over participant stopping times, while differing only in how their designs are acquired. This protocol is conservative in terms of compute for MA-BED as each baseline receives a full training budget for its acquisition policy and an additional budget for the hazard-specific action policy, whereas MA-BED jointly trains its acquisition and action policies within a single training budget.

Training details. All learned acquisition policies use the same design-policy network architecture. Since CES designs must lie in $[0, 100]^6$, we apply a sigmoid activation to the emitter output followed by rescaling to $[0, 100]$, ensuring that every proposed pair of baskets remains within the admissible design space. For DAD, we optimize the sPCE objective using the REINFORCE gradient estimator. A pathwise gradient through the observation model is not directly applicable because the clipped CES response distribution contains atoms at the clipping boundaries $y = \varepsilon$ and $y = 1 - \varepsilon$. The remaining methods are trained end-to-end through their respective differentiable objectives. For the action policy, the only architectural difference compared to the description in Appendix D is the input dimension. Indeed, we augment the history with the price vector p so the policy can learn to make useful recommendations under the budget constraint. Unless stated otherwise, the architectural and optimization hyperparameters are shared across methods whenever applicable.

Table 7: **Training hyperparameters for the CES experiment.**

Hyperparameter	DAD	Action-BED	MA-BED
Policy Training			
Batch size	2,000	2,000	2,000
Training iterations	10^5	10^5	10^5
Seed	42	42	42
Contrastive samples, L	2,000	–	–
Policy Optimization			
Design objective	sPCE	Terminal regret	Multistep regret
Gradient estimator	REINFORCE	Pathwise	Pathwise
Optimizer	Adam	Adam	Adam
Betas, β	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.999)$
Learning rate	5×10^{-5}	5×10^{-4}	5×10^{-4}
Scheduler	ExpLR	ExpLR	ExpLR
Decay	0.95	0.95	0.95
Decay period	2,000	2,000	2,000
Action-Policy Retraining			
Retraining iterations	10^5	10^5	–
Learning rate	5×10^{-4}	5×10^{-4}	–
Betas, β	$(0.9, 0.999)$	$(0.9, 0.999)$	–

Evaluation details. For each hazard rate q, we evaluate all policies over 10,000 independent rollouts for the expected recommendation regret induced by the stopping-time distribution. In particular, varying q controls the urgency of the recommendation problem: larger hazard rates place greater emphasis on early recommendations, whereas smaller rates allow the policy to exploit longer acquisition trajectories. For MA-BED, this trade-off directly affects the acquisition policy itself, which can adapt its designs to when accurate recommendations are most likely to be required.

As complementary diagnostics, we also report the per-step losses, sPCE and sNMC estimates of EIG for each acquisition policy. Unlike recommendation regret, these latter quantities are not target metrics for the decision problem because they measure how informative the collected designs are about the full latent preference parameter θ. They therefore allow us to distinguish general parameter identification from decision-relevant information acquisition. Figure 5 reports both the downstream regret trajectories and the corresponding EIG diagnostics for every method and hazard rate considered.

E.3 Source Location Finding

We use the source location finding benchmark, in which an agent sequentially places a sensor in $\mathbb{R}^d$ to infer the positions of two unknown signal sources. Following the setting of Foster et al. (2021), the source locations are sampled independently as

$$\theta_k \sim \mathcal{N}(0_d, I_d), \qquad k = 1, 2.$$

At step t, the agent selects a sensor location $\xi_t \in \mathbb{R}^d$. The noiseless signal intensity is

$$\mu(\theta, \xi_t) = b + \sum_{k=1}^{2} \frac{A}{m + ||\xi_t - \theta_k||_2^2},$$

and observations are generated according to

$$\log y_t = \log \mu(\theta, \xi_t) + \sigma\varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, 1).$$

This observation model is explicitly reparameterizable through ε_t. The exact experimental parameters are given in Table 8.

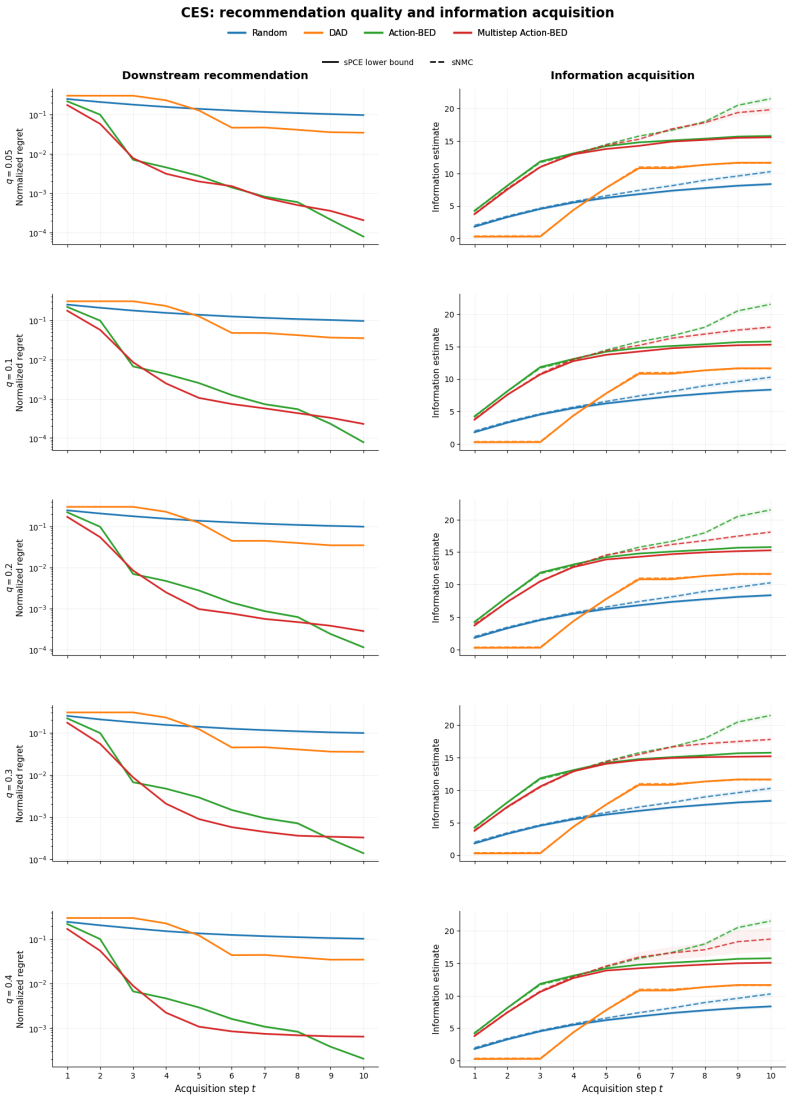

Figure 5: **Per-step recommendation performance and information acquisition on CES under stochastic participation horizons.** Each row corresponds to a different hazard rate q. *Left:* mean normalized recommendation regret at each acquisition step, evaluated over 10,000 independent rollouts; shaded regions denote standard errors. Increasing q places greater probability on early termination and therefore increases the importance of producing accurate recommendations early. *Right:* per-step sPCE and sNMC estimates evaluated with $L = 10^7$ contrastive samples across 256 rollouts.

Table 8: Parameters of the source location finding environment.

Parameter	**Value**
Number of sources, K	2
Design dimension	$d \in \{2, 5\}$
Observation dimension	1
Horizon	$T \in \{30, 50, 75, 100\}$
Base signal, b	10^{-1}
Max. signal, m	10^{-4}
A	1
Observation noise, σ	0.5

The objective of this task is therefore to sequentially collect experimental data, $h_t = \{(\xi_i, y_i)\}_{i=1}^t$ so as to enable the most accurate possible prediction of the source locations. Since MA-BED is trained with the multistep loss, the action policy π_a produces, at each step t and given the current history h_t, an estimate of the source configuration, $\hat{\theta}_t = \pi_a(h_t)$.

Downstream losses. We evaluate source-location prediction using a permutation-invariant mean squared error (MSE), since the identities of the two sources are exchangeable. At each acquisition step $t \in \{1, \dots, T\}$, the prediction $\hat{\theta}_t$ is evaluated as

$$\ell_{\mathrm{MSE}}(\theta, \hat{\theta}_t) = \min_{\sigma \in \mathcal{S}_2} \frac{1}{2d} \sum_{k=1}^{2} \left\| \hat{\theta}_{t,\sigma(k)} - \theta_k \right\|_2^2, \tag{96}$$

where $\mathcal{S}_2$ denotes the set of permutations of the two sources.

Our first evaluation criterion measures performance when the acquisition horizon is not known in advance. We define the *Random-Horizon MSE* (RH-MSE) as

$$\ell_{\mathrm{RH}}(\theta, \hat{\theta}_{1:T}) = \sum_{t=1}^{T} p_\gamma(t) \ell_{\mathrm{MSE}}(\theta, \hat{\theta}_t), \qquad p_\gamma(t) = \frac{\gamma^{T-t}}{\sum_{s=1}^{T} \gamma^{T-s}}, \tag{97}$$

with $\gamma = 0.95$. Equivalently, this is the expected downstream loss at a random stopping time $\tau \sim p_\gamma$. The same weighting is used in the multistep objective optimized by MA-BED, so that RH-MSE directly measures the quality of a policy when a downstream decision may be required at different points along the acquisition trajectory.

Random-horizon evaluation protocol. Each acquisition policy is first trained using its native objective. MA-BED jointly optimizes its design and action policies using the weighted multistep loss

$$\ell_{\mathrm{MS}}^{\lambda} = \sum_{t=1}^{T} \lambda_t \ell_{\mathrm{MSE}}\big(\theta, \pi_a(h_t)\big), \qquad \lambda_t = p_\gamma(t), \tag{98}$$

whereas ACTION-BED jointly trains its design and downstream policies using only the terminal loss $\ell_{\mathrm{MSE}}(\theta, \hat{\theta}_T)$, following Rossa et al. (2026). DAD optimizes its native information-theoretic objective, the sPCE lower bound on expected information gain. STATIC jointly optimizes a fixed sequence of T designs for the terminal source-location objective, with all designs chosen before observing any measurements and therefore without adaptation to the acquisition history. RANDOM uses an untrained acquisition policy that samples designs independently from the reference design distribution.

To compare the resulting acquisition policies independently of their native downstream parameterizations, we subsequently freeze each design policy and train a new multistep action network on top of it. The same architecture, multistep objective, and optimization budget are used for every method, with training stopped early if it exceeds five days of wall-clock time. Each method is therefore evaluated through an identical downstream predictor trained to solve the decision problem from every trajectory prefix. We report the resulting RH-MSE in (97). This protocol isolates differences in the information collected by the design policies from differences in downstream network capacity or training. See Table 9 for the exact training hyperparameters.

Long-horizon terminal evaluation. We next study how the methods scale as the acquisition problem becomes longer and higher-dimensional. We consider $d = 2$ and $T \in \{30, 50, 75\}$, as well as $d = 5$ and $T \in \{30, 50, 75, 100\}$. In this experiment, we only focus on the terminal MSE, i.e., the source-location prediction error after all T acquisitions.

The acquisition policies are again trained using their respective native objectives, under matched architectural and optimization conditions as described in Appendix D. MA-BED is evaluated using the terminal prediction of its jointly trained multistep action network, while ACTION-BED uses its native terminal action network. Since DAD, STATIC and RANDOM do not natively optimize a downstream prediction policy, their acquisition policies are frozen after training and an additional action network is fitted specifically for terminal source-location prediction. This provides each acquisition strategy with an appropriate downstream decoder while preserving the end-to-end solutions naturally learned by the two loss-driven methods.

Table 9: **Training hyperparameters for the source location finding.**

Hyperparameter	DAD	Action-BED	MA-BED
Policy Training			
Batch size	2,000	2,000	2,000
Training iterations	3×10^5	3×10^5	3×10^5
Seed	42	42	42
Contrastive samples, L	2,000	–	–
Policy Optimization			
Optimizer	Adam	Adam	Adam
Betas, β	$(0.9, 0.999)$	$(0.9, 0.999)$	$(0.9, 0.999)$
Learning rate	10^{-3}	10^{-3}	10^{-3}
Scheduler	ExpLR	ExpLR	ExpLR
Decay	0.95	0.95	0.95
Decay period	2,000	2,000	2,000
Action-Policy Retraining			
Retraining iterations	3×10^5	3×10^5	–
Learning rate	10^{-3}	10^{-3}	–
Betas, β	$(0.9, 0.999)$	$(0.9, 0.999)$	–

For all values of (T, d), policy-based methods use the same design-policy backbone and are trained under the same experimental conditions; only the acquisition horizon and source dimension are varied. In addition to terminal MSE, we report sPCE and sNMC estimates of the expected information gain as complementary measures of the informativeness of the learned acquisition trajectories. These quantities are not the primary optimization target of the comparison, but provide an information-theoretic characterization of the learned design policies and include the native objective optimized by DAD.

Early-identification of source location with learned confidence. We finally investigate whether the acquisition policies learn to identify the source locations reliably early in the rollout. For each method, we freeze both the learned design policy and its associated multistep action network, and train an additional confidence network on trajectories generated by the resulting policy for a prescribed prediction-error tolerance ϵ. The confidence network therefore receives the current history h_t and outputs a logit $c_t(h_t)$, whose sigmoid is trained to estimate

$$\sigma(c_t(h_t)) \approx \Pr\left(\ell_t \leq \epsilon \mid h_t\right),$$

where $\ell_t = \ell_t^{MSE}$ denotes the realized downstream prediction error at step t with the same permutation-invariant MSE. The confidence head is fitted using binary cross-entropy against the labels

$$z_t = \mathbb{1}\{\ell_t \leq \epsilon\},$$

while keeping both the acquisition and action policies fixed. The confidence-network architecture is reported in Table 10, and all methods use the same training hyperparameters reported in Table 11. We additionally verify on held-out trajectories that the resulting confidence estimates are approximately calibrated, by grouping predictions into probability bins and comparing the mean predicted probability in each bin with the corresponding empirical frequency of the event $\ell_t \leq \epsilon$.

For a fixed confidence level $\delta = 0.05$, we consider

$$\epsilon \in \{0.001, 0.005, 0.008, 0.01, 0.015, 0.02, 0.03, 0.05, 0.1\},$$

and record the earliest acquisition step at which the model assigns probability at least $1 - \delta$ to achieving the corresponding prediction accuracy. This provides a diagnostic of how rapidly each learned acquisition strategy accumulates enough information to localize the sources with high confidence. In particular, reaching a given accuracy level earlier indicates that the design policy has already acquired sufficient information to identify the sources from a shorter trajectory. The experiment therefore complements the per-step loss analysis by directly probing whether the multistep objective encourages MA-BED to resolve the downstream task earlier along the acquisition process.

Table 10: **Confidence-network architecture.** The same architecture is used for all acquisition policies.

Component	Specification
History element	$(\xi_t, y_t) \in \mathbb{R}^{d_\xi + d_y}$
Input representation	Zero-padded prefix h_t
Input dimension	$T(d_\xi + d_y)$
Backbone	MLP
Hidden dimensions	$(512, 256, 128)$
Activation	`GELU`
Parameter sharing	Shared across all $t = 1, \dots, T$
Output dimension	1 logit per prefix
Confidence estimate	$q_t(h_t) = \sigma(c_t(h_t))$
Training target	$\mathbb{1}\{\ell_t \leq \epsilon\}$

Table 11: **Confidence-network training hyperparameters.** The same training protocol is used for all acquisition methods.

Hyperparameter	Value
Horizon, T	50
Training objective	Binary cross-entropy with logits
Optimizer	Adam
Learning rate	10^{-3}
Batch size	2,000
Training iterations	50,000

E.4 Additional results

Tables 12 and 13 report the complete results for all methods and experimental settings, including terminal MSE, RH-MSE, sPCE, and sNMC. We additionally report the per-step downstream losses of Action-BED and MA-BED across all acquisition horizons, in Figures 6 and 7 for $d = 2$ and $d = 5$, respectively. For this comparison, each method is evaluated using its own multistep action network, allowing us to examine how downstream prediction quality evolves throughout the acquisition trajectory under the corresponding learned design policy.

Training robustness across seeds. To verify that the gains observed as the problem difficulty increases are not driven by favorable optimization runs, we repeat the complete training procedure with five independent random seeds for both Action-BED and MA-BED. We restrict this experiment to the two-dimensional setting, $d = 2$, both to illustrate the phenomenon clearly and because training becomes computationally expensive at larger dimensions. We also omit DAD from this comparison: in our experiments, it consistently achieved substantially worse performance while incurring significantly higher training costs, with particularly unfavorable scaling as the problem dimension increases. Each seed corresponds to an independent policy initialization and training run, followed by evaluation under the same protocol. Table 14 reports the terminal MSE and Random-Horizon MSE, averaged across the five trained policies. The advantage of MA-BED is preserved across independent runs for both metrics, showing that its improved terminal and Random-Horizon performance is robust to training variability rather than being attributable to a particular initialization.

Scaling of training cost with horizon. We further verify that the performance gains of MA-BED do not arise from a larger optimization budget. Across methods, we keep the policy architecture, batch size, number of optimization steps, optimizer, and training protocol fixed, so that increasing the horizon changes only the cost of processing longer acquisition trajectories. The multistep objective reuses the intermediate representations already produced during the rollout and only evaluates the downstream loss at additional prefixes. Consequently, MA-BED achieves its improved terminal and Random-Horizon performance with little to no additional training cost relative to the corresponding terminally trained policy with Action-BED as reported in Table 15. In particular, its computational scaling with T closely follows that of Action-BED, indicating that dense supervision improves how the available training signal is used rather than relying on additional optimization compute.

Table 12: **Varying-horizon source location finding ($d = 2$).** Prediction performance and information-theoretic evaluation are reported for increasing acquisition horizons T. MSE denotes the terminal MSE, ℓ_T^{MSE}. Uncertainties are reported as one standard error over $50{,}000$ rollouts. sPCE and sNMC are computed with $2{,}000$ outer samples and $1e7$ contrastive samples.

T	Method	MSE (10^{-3}) ↓	RH-MSE (10^{-2}) ↓	sPCE ↑	sNMC ↑
30	RANDOM	183.94 ± 1.61	28.54 ± 0.18	8.24 ± 0.04	8.50 ± 0.06
	STATIC	88.08 ± 1.58	19.21 ± 0.17	8.85 ± 0.04	8.85 ± 0.04
	DAD	65.89 ± 1.38	16.59 ± 0.16	$\mathbf{14.66 \pm 0.04}$	$\mathbf{16.30 \pm 0.09}$
	ACTION-BED	8.32 ± 0.24	12.69 ± 0.58	11.60 ± 0.05	11.72 ± 0.06
	MA-BED	$\mathbf{7.13 \pm 0.22}$	$\mathbf{8.34 \pm 0.06}$	12.78 ± 0.05	13.13 ± 0.06
50	RANDOM	213.18 ± 1.71	21.70 ± 0.16	9.59 ± 0.03	10.35 ± 0.06
	STATIC	49.27 ± 1.19	11.16 ± 0.14	10.28 ± 0.04	10.29 ± 0.04
	DAD	22.14 ± 0.83	7.31 ± 0.10	$\mathbf{15.41 \pm 0.03}$	$\mathbf{18.67 \pm 0.11}$
	ACTION-BED	2.95 ± 0.07	6.30 ± 0.02	13.24 ± 0.04	13.68 ± 0.06
	MA-BED	$\mathbf{1.37 \pm 0.03}$	$\mathbf{2.70 \pm 0.02}$	15.09 ± 0.03	17.99 ± 0.12
75	RANDOM	298.21 ± 1.94	20.40 ± 0.16	10.39 ± 0.03	11.98 ± 0.07
	STATIC	27.40 ± 0.87	4.95 ± 0.09	11.36 ± 0.04	11.38 ± 0.04
	DAD	22.09 ± 0.87	3.71 ± 0.09	$\mathbf{15.51 \pm 0.02}$	18.90 ± 0.10
	ACTION-BED	1.72 ± 0.10	2.66 ± 0.02	14.48 ± 0.03	15.78 ± 0.09
	MA-BED	$\mathbf{0.82 \pm 0.02}$	$\mathbf{0.89 \pm 0.01}$	15.43 ± 0.03	$\mathbf{19.16 \pm 0.12}$

Table 13: **Varying-horizon source location finding ($d = 5$).** Evaluations use the same settings as Table 12.

T	Method	MSE (10^{-1}) ↓	RH-MSE (10^{-1}) ↓	sPCE ↑	sNMC ↑
30	RANDOM	6.52 ± 0.02	7.24 ± 0.02	4.54 ± 0.05	4.56 ± 0.05
	STATIC	5.65 ± 0.02	6.64 ± 0.02	5.44 ± 0.05	5.44 ± 0.05
	DAD	6.07 ± 0.02	6.87 ± 0.02	$\mathbf{11.23 \pm 0.09}$	$\mathbf{11.76 \pm 0.10}$
	ACTION-BED	$\mathbf{4.23 \pm 0.02}$	5.82 ± 0.01	6.47 ± 0.07	6.50 ± 0.08
	MA-BED	4.30 ± 0.02	$\mathbf{5.75 \pm 0.02}$	6.62 ± 0.08	6.67 ± 0.08
50	RANDOM	6.53 ± 0.02	6.69 ± 0.02	6.37 ± 0.05	6.41 ± 0.06
	STATIC	4.60 ± 0.02	5.46 ± 0.02	7.31 ± 0.06	7.31 ± 0.06
	DAD	5.31 ± 0.02	5.91 ± 0.02	$\mathbf{14.81 \pm 0.05}$	$\mathbf{18.29 \pm 0.12}$
	ACTION-BED	$\mathbf{2.66 \pm 0.01}$	4.21 ± 0.02	10.34 ± 0.09	11.06 ± 0.12
	MA-BED	2.68 ± 0.01	$\mathbf{4.05 \pm 0.01}$	10.51 ± 0.10	11.51 ± 0.14
75	RANDOM	6.31 ± 0.01	6.59 ± 0.02	8.05 ± 0.05	8.28 ± 0.06
	STATIC	3.71 ± 0.02	4.41 ± 0.02	9.17 ± 0.06	9.19 ± 0.06
	DAD	4.58 ± 0.02	4.97 ± 0.02	$\mathbf{15.71 \pm 0.03}$	$\mathbf{21.80 \pm 0.13}$
	ACTION-BED	1.70 ± 0.01	2.84 ± 0.01	13.12 ± 0.08	15.26 ± 0.14
	MA-BED	$\mathbf{1.41 \pm 0.01}$	$\mathbf{2.47 \pm 0.01}$	14.23 ± 0.07	21.17 ± 0.23
100	RANDOM	6.28 ± 0.01	6.29 ± 0.02	9.29 ± 0.05	9.86 ± 0.07
	STATIC	3.13 ± 0.02	3.63 ± 0.02	10.43 ± 0.06	10.47 ± 0.06
	DAD	3.95 ± 0.02	4.22 ± 0.02	$\mathbf{15.87 \pm 0.02}$	23.37 ± 0.14
	ACTION-BED	0.75 ± 0.01	1.60 ± 0.01	15.73 ± 0.03	34.04 ± 0.37
	MA-BED	$\mathbf{0.72 \pm 0.01}$	$\mathbf{1.38 \pm 0.01}$	15.57 ± 0.04	$\mathbf{39.99 \pm 0.50}$

Table 14: **Robustness across independent training seeds ($d = 2$).** Mean ± standard error over five independently trained policies.

T	**Method**	**MSE** (10^{-3}) ↓	**RH-MSE**(10^{-2}) ↓	**sPCE** ↑	**sNMC** ↑
30	ACTION-BED	8.19 ± 0.34	12.72 ± 0.19	11.65 ± 0.14	11.77 ± 0.15
	MA-BED	6.73 ± 0.02	8.21 ± 0.06	13.03 ± 0.07	13.47 ± 0.10
50	ACTION-BED	3.20 ± 0.12	6.13 ± 0.11	13.15 ± 0.08	13.59 ± 0.08
	MA-BED	1.46 ± 0.08	2.82 ± 0.03	15.09 ± 0.04	17.95 ± 0.21

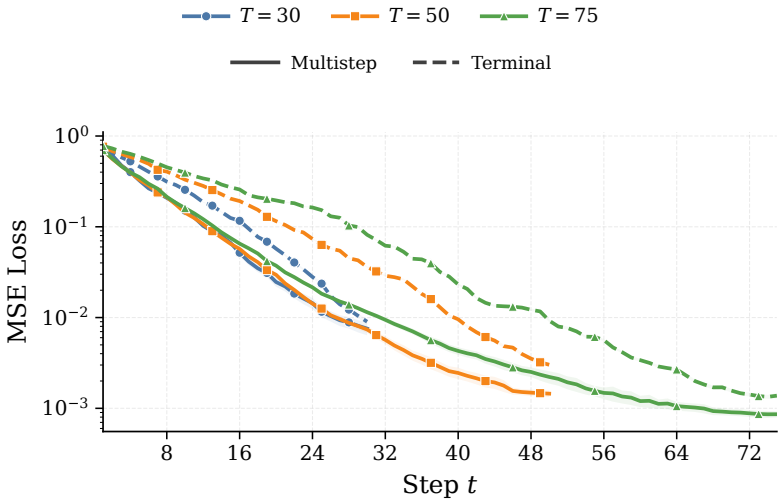

Figure 6: **Per-step loss, ℓ_t^{MSE}, across training horizons for source location finding ($d = 2$).** We compare MA-BED and ACTION-BED for policies trained with different acquisition horizons T. Each curve reports the average loss at every acquisition step over 50,000 independent rollouts, using the multistep action network associated with each learned acquisition policy.

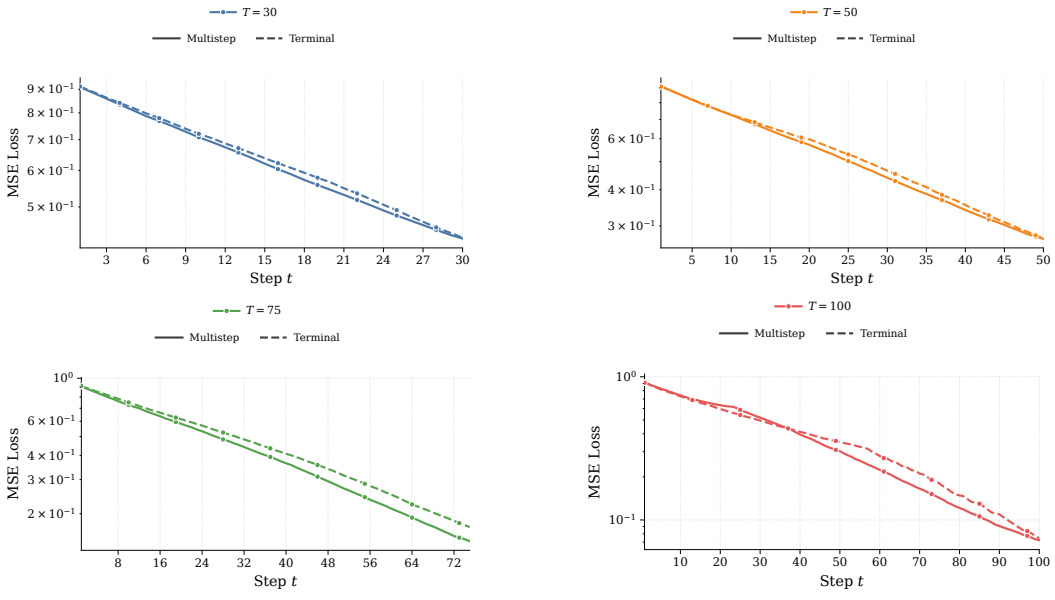

Figure 7: **Per-step loss, ℓ_t^{MSE}, across training horizons for source location finding ($d = 5$).** Same evaluation protocol as in the $d = 2$ setting.

Table 15: **Training time for source location finding.** Wall-clock training time (hours) on a single GPU under matched architectures and optimization budgets. All methods are trained for a target budget of 300k gradient steps. DAD runs exceeding the five-day wall-clock limit are stopped early; for these runs, we additionally report the fraction of the prescribed training budget completed at termination.

d	T	ACTION-BED	MA-BED	DAD	DAD completed (%)
2	30	13.88	13.96	32.4	100
	50	31.52	31.21	119.8	93.3
	75	65.86	66.00	119.7	63.5
5	30	13.79	13.95	43.3	100
	50	31.54	31.62	94.3	100
	75	66.12	66.63	119.7	77.5
	100	110.62	111.10	119.6	43.3