IS IN-CONTEXT REINFORCEMENT LEARNING A POSTERIOR SAMPLER? AN EMPIRICAL STUDY

Anonymous authors
Paper under double-blind review

ABSTRACT

In-context reinforcement learning (ICRL) exploits the impressive in-context learning ability of large language models to zero-shot generalize to new, unseen tasks at inference time without any parameter updates. Among many approaches, the Decision-Pretrained Transformer (DPT) is particularly appealing for its theoretical grounding: under idealized assumptions such as a sufficiently expressive model and infinite pretraining data, DPT can perform Bayesian posterior sampling. Whether this characterization survives in practice remains open. Therefore, we ask: *is ICRL, specifically DPT and our proposed DPT variants, a Bayesian posterior sampler in practice?* To approach this question, we establish two necessary conditions that a Bayesian posterior sampler must satisfy: *sufficient-statistic invariance* and the *martingale property*. We then construct statistical hypothesis tests to statistically assess if practically-trained ICRL models satisfy these conditions. Applying these tests in Gaussian, Bernoulli, and linear bandit settings, we find that both conditions are consistently violated, suggesting that ICRL may not behave as Bayesian posterior samplers in practice. Beyond this negative result, our tests provide a principled toolkit for testing the Bayesian behavior of ICRL.

1 INTRODUCTION

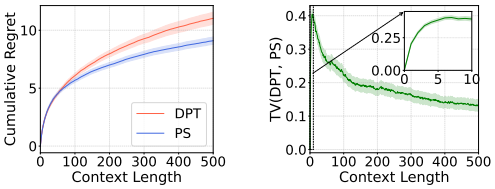
Reinforcement learning (RL) has demonstrated remarkable performance in a wide range of applications, such as strategy games (Mnih et al., 2013; 2015), robotics (Levine et al., 2016; Duan et al., 2016), and recommender systems (Afsar et al., 2023; Lin et al., 2023). However, a fundamental challenge in RL is the limited capability to few-shot or zero-shot generalize to new unseen tasks, as traditional RL methods learn a policy for a single task through an ocean of environment interactions and extensive trials-and-errors, and thus typically requires retraining a policy from scratch for each new task. In-context RL (ICRL) offers a compelling alternative attributed to the impressive capability of in-context learning by large language models (LLMs) (Dong et al., 2024; Li et al., 2023; Wei et al., 2023). Specifically, a transformer (Vaswani et al., 2017) pretrained on a diverse set of tasks/environments can adapt to and reinforcement learn in new, unseen tasks at test time, purely by conditioning on interactions with the new, unseen tasks (a.k.a. context) without any parameter updates. Several ICRL frameworks have been proposed to realize this vision. *Algorithm Distillation* (AD) pretrains a transformer on the complete learning histories of an RL algorithm (e.g., PPO (Schulman et al., 2017) or SAC (Haarnoja et al., 2018)), where the context consists of a sequence of episodes exhibiting progressive improvement and dynamics (Laskin et al., 2023). The transformer learns to extrapolate this improvement operator, effectively distilling the source RL algorithm's learning dynamics into its in-context behavior. *Decision-Pretrained Transformer* (DPT) adopts a fundamentally different approach (Lee et al., 2023). Rather than imitating an evolving algorithm, DPT pretrains a transformer via supervised learning to predict optimal actions conditioned on contexts of varying lengths. Moreover, DPT can employ arbitrary context data, such as interactions collected under a random policy (Chen & Paternain, 2024), by leveraging the optimal action labels.

Notably, a central theoretical grounding underpins DPT's appeal. Prior works (Lee et al., 2023; Lin et al., 2024) establish a theoretical understanding that "DPT is an efficient implementation of *Bayesian posterior sampling*" that achieves rate-optimal Bayesian regret (Russo & Van Roy, 2016). This, however, relies on strong assumptions such as a sufficiently expressive model and infinite pretraining data to exactly match or closely approximate the pretraining data distribution (Lee et al., 2023; Lin et al., 2024), which may not hold in practice.

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.

In addition, an empirical gap between DPT and a Bayesian posterior sampler (PS) is observed from experiments (see Figure 1), where we compare the default setting of DPT as in (Lee et al., 2023) and Thompson sampling (Thompson, 1933) (a Bayesian posterior sampler in multi-armed bandits) in Gaussian (multi-armed) bandits. Accordingly, this paper aims to explore and answer the following research question:

Is ICRL, including DPT and our proposed DPT variants, a Bayesian posterior sampler in practice?



(a) Cumulative Regret (b) TV Distance

Figure 1: The empirical gap between DPT and PS. "TV Distance" refers to the total variation distance between DPT's and PS's action distribution.

Furthermore, it is worth highlighting that the research question above is also motivated by an empirical study (Falck et al., 2024), where the in-context learning of common LLMs like GPT-3.5 (OpenAI, n.d.), GPT-4 (OpenAI, 2023), Llama-2 (Touvron et al., 2023), and Mistral (Jiang et al., 2023) is demonstrated non-Bayesian from a martingale perspective (a necessary condition for Bayesian systems). Inspired by (Falck et al., 2024), we take a statistical hypothesis testing approach to the research question above. Our **main contributions** are as follows.

- **Necessary conditions.** We provide a behavioral characterization of a transformer as a Bayesian posterior sampler in terms of its action distribution, and derive two necessary conditions from it, namely *sufficient-statistic invariance* and the *martingale property*, which we instantiate for Gaussian, Bernoulli, and linear bandits.
- **Testing framework and DPT variants.** We turn each condition into a hypothesis test on action probabilities. Since DPT predicts only actions, we propose DPT′ and GDPT, which augment DPT with a reward head so that the martingale property becomes testable.
- **Empirical findings.** Across all bandit settings, deployment modes, context transformations, and context lengths, DPT, DPT′, and GDPT consistently violate sufficient-statistic invariance, and DPT′ and GDPT additionally violate the martingale property, even when their decision performance closely tracks Thompson Sampling. Ablation studies that enforce permutation invariance do not remove the violations. Beyond this negative result, our statistical tests provide a principled toolkit for auditing Bayesian claims about ICRL.

2 In-Context Reinforcement Learning Under DPT

This section describes the three main phases of ICRL under the DPT framework (Lee et al., 2023): pretraining dataset collection, pretraining, and deployment. We consider a finite-horizon decision-making setting where each environment (or task) $\tau = \langle \mathcal{A}, \mathcal{R}, H \rangle$ consists of an action space $\mathcal{A}$, reward kernel $\mathcal{R} : \mathcal{A} \to \Delta(\mathbb{R})$ evaluating the quality of the decision, and the horizon H.

2.1 Pretraining Dataset Collection

The pretraining dataset is collected by repeatedly sampling environments, collecting context data, and labeling optimal actions. More concretely, for each instance $i \in \{1, \dots, N\}$:

$i)$ **Sample an environment:** Draw $\tau_i \sim \mathcal{T}_{\text{env}}$ from a distribution $\mathcal{T}_{\text{env}}$ over environments.

$ii)$ **Collect context data:** Generate context data with n transitions $D_n^{(i)} = \big((A_1^{(i)}, R_1^{(i)}), \dots, (A_n^{(i)}, R_n^{(i)})\big)$ by deploying a context-collection policy in environment τ_i, i.e., $D_n^{(i)} \sim \mathcal{D}_{\text{context}}(\cdot\,; \tau_i) \in \Delta((\mathcal{A} \times \mathbb{R})^n)$. The context-collection policy may be arbitrary, ranging from a simple random policy to more sophisticated approaches such as PPO or posterior sampling. Throughout the paper, a context is an *ordered* sequence of action–reward pairs; uppercase (A_t, R_t) denote the random action and reward at position t, and lowercase (a, r) denote their realizations.

iii) **Label the optimal action and construct the pretraining dataset:** Let $a^{*,(i)} \sim \pi^*_{\tau_i}$, where $\pi^*_{\tau_i}$ denotes the optimal policy for the environment τ_i. Select the optimal action $a^{*,(i)}$ as the label, and add tuple $(D_n^{(i)}, a^{*,(i)})$ to the pretraining dataset $\mathcal{B}$.

Together, the joint pretraining distribution is given by

$$\mathcal{P}_{\mathrm{pre}}(\tau, D, a^*) = \mathcal{T}_{\mathrm{env}}(\tau) \cdot \mathcal{D}_{\mathrm{context}}(D; \tau) \cdot \pi^*_\tau(a^*). \tag{1}$$

2.2 Pretraining

Having collected the pretraining dataset $\mathcal{B}$, we use it to pretrain a decoder-only transformer such as GPT-2 (Radford et al., 2019), with L layers, H_{attn} attention heads, and embedding dimension d_{emb}.

Architecture. Consider a generic instance $(D_n, a^*) \in \mathcal{B}$, where we drop the index i for brevity. Following Lee et al. (2023), each transition in the context $D_n = \big((A_1, R_1), \ldots, (A_n, R_n)\big)$ is stacked into a single token $\xi_t = (A_t, R_t) \in \mathbb{R}^{d_\xi}$ with $d_\xi = d_{\mathcal{A}} + 1$, where the action A_t is identified with its encoding in $\mathbb{R}^{d_{\mathcal{A}}}$. A fixed start token $\xi_0 = \mathbf{1} \in \mathbb{R}^{d_\xi}$, which plays the role of the query-state token in (Lee et al., 2023) (a constant in the multi-armed bandit setting), is prepended, giving the input sequence $\xi_{0:n} = (\xi_0, \xi_1, \ldots, \xi_n) \in \mathbb{R}^{d_\xi \times (n+1)}$. Each token of $\xi_{0:n}$ is first mapped to a d_{emb}-dimensional embedding by a linear layer, and a learned positional encoding is added, yielding $h_t^{(0)} \in \mathbb{R}^{d_{\mathrm{emb}}}$ for position $t = 0, \ldots, n$. The embedded sequence then passes through L standard transformer blocks. Each block performs two operations, each wrapped in a residual connection: (i) causal multi-head self-attention, which updates the representation at position t by aggregating information from positions $\leq t$, and (ii) a position-wise feed-forward network, e.g., an MLP applied independently to each position with weights shared across positions. A linear head W_{out} and a Softmax operation map the final-layer representation $h_t^{(L)}$ at each position to a distribution over actions

$$\pi_{\mathrm{TF}}(\cdot \mid D_t) := \mathrm{Softmax}\big(W_{\mathrm{out}}\, h_t^{(L)}\big) \in \Delta(\mathcal{A}), \qquad t = 0, \ldots, n, \tag{2}$$

where $D_t := \big((A_1, R_1), \ldots, (A_t, R_t)\big)$ is the length-t prefix of the context and D_0 is the empty context. Due to the causal attention, $\pi_{\mathrm{TF}}(\cdot \mid D_t)$ depends only on the first t transitions, not on future context tokens. In particular, $\pi_{\mathrm{TF}}(\cdot \mid D_0)$ acts without any information about the environment, and can be interpreted as the model's prior.

Loss function. The transformer is trained to minimize the expected cross-entropy loss over the pretraining distribution

$$\min_\psi \ \mathbb{E}_{\mathcal{P}_{\mathrm{pre}}}\left[\sum_{t=1}^{n} \ell\big(\pi_{\mathrm{TF}}(\cdot \mid D_t), a^*\big)\right], \quad \ell\big(\pi_{\mathrm{TF}}(\cdot \mid D_t), a^*\big) := -\log \pi_{\mathrm{TF}}(a^* \mid D_t), \tag{3}$$

where ψ denotes the transformer's parameters.

2.3 Deployment

After pretraining, the model is deployed in a new environment $\tau \sim \mathcal{T}_{\mathrm{test}}$ without any parameter updates. Given a context D_t, the model is queried via a forward pass on $\xi_{0:t}$, and the action distribution $\pi_{\mathrm{TF}}(\cdot \mid D_t)$ at the *last position* is read out. DPT supports two types of deployment, offline and online, which differ in how the input context is constructed.

Offline deployment. The model does not collect the context itself. Instead, a static context $D_n \sim \mathcal{D}_{\mathrm{test}}(\cdot\,; \tau)$ of n transitions is provided, and the model acts according to $\pi_{\mathrm{TF}}(\cdot \mid D_n)$.

Online deployment. The context is initialized as the empty context D_0, and the model collects its own context. At each round $t \in [n]$, the model samples an action from its predicted distribution, $A_t \sim \pi_{\mathrm{TF}}(\cdot \mid D_{t-1})$, observes a reward R_t from the test environment, and appends the transition to the context, i.e., $D_t = \big(D_{t-1}, (A_t, R_t)\big)$.

3 Are DPT and Its Variants Bayesian Posterior Samplers?

Having introduced the DPT framework, we emphasize that the idealized assumptions under which DPT is expected to behave as a Bayesian posterior sampler, such as a sufficiently expressive model

and infinite pretraining data, may not hold in practice. To investigate whether DPT exhibits Bayesian posterior sampling behavior in practice, this section establishes two necessary conditions that any Bayesian posterior sampler must satisfy and develops a statistical testing framework to quantitatively assess if DPT and its variants satisfies these conditions. All proofs in this section are provided in Appendix B.1. We begin by characterizing the transformer as a Bayesian posterior sampler, motivated by its role as the common decision-making component across most ICRL algorithms, including DPT.

3.1 Transformer as a Bayesian Posterior Sampler

We first formalize what it means for the transformer to behave as a Bayesian posterior sampler.

Definition 1 (Transformer as a Bayesian Posterior Sampler)**.** *Consider a decision-making problem with some unknown latent parameter $\theta \in \Theta$. Let $D_n = \big((A_1, R_1), \ldots, (A_n, R_n)\big)$ denote a context, and let $\pi_{\mathrm{TF}}(a \mid D_n)$ represent the transformer's action distribution given the context D_n. We say that the transformer behaves as a Bayesian posterior sampler if there exist a prior distribution $\pi_0(\theta)$ and a context-independent decision kernel $\mathcal{K}(a \mid \theta)$ such that*

$$\pi_{\mathrm{TF}}(a \mid D_n) = \int_{\Theta} \mathcal{K}(a \mid \theta)\, p_{\pi_0}(\theta \mid D_n)\, d\theta, \tag{4}$$

where $p_{\pi_0}(\theta \mid D_n)$ is the Bayesian posterior induced by the prior $\pi_0(\theta)$ and the likelihood model $p(\cdot \mid \theta)$, namely $p_{\pi_0}(\theta \mid D_n) \propto \pi_0(\theta)\, p(D_n \mid \theta)$.

Definition 1 provides a purely behavioral characterization in the sense that it only requires the transformer policy to "behave" as a Bayesian posterior sampler in terms of the action distribution. In particular, the definition does not require the transformer to explicitly infer a latent parameter θ or maintain an explicit posterior distribution over θ. Building on Definition 1, we establish the two necessary conditions that any Bayesian posterior sampler must satisfy: sufficient-statistic invariance and the martingale property.

3.1.1 Sufficient-Statistic Invariance

Let $\mathcal{X} = \{x_1, \ldots, x_K\}$ be the known action features and $\theta \in \Theta$ be the unknown parameter for the decision-making problem in Definition 1. Recall that the finite-dimensional exponential-family reward likelihoods take the form

$$p(r \mid x, \theta) = h(r, x) \exp\left\{\eta(\theta)^\top T(r, x) - \lambda(\theta)^\top \phi(x)\right\}, \tag{5}$$

where $h(r, x)$ is independent of θ, $T(r, x) \in \mathbb{R}^q$ and $\phi(x) \in \mathbb{R}^p$ are fixed functions, and $\eta(\theta)$ and $\lambda(\theta)$ are parameter-dependent vectors. For a context $D_n = \big((A_1, R_1), \ldots, (A_n, R_n)\big)$, define

$$S_n^{(1)} = \sum_{t=1}^{n} T(R_t, x_{A_t}), \quad S_n^{(2)} = \sum_{t=1}^{n} \phi(x_{A_t}), \quad \text{and} \quad S(D_n) = \left(S_n^{(1)}, S_n^{(2)}\right).$$

A statistic $S(D_n)$ is sufficient for θ if the distribution of the context D_n is independent of θ conditional on $S(D_n)$. Consequently, the posterior depends on the context only through the sufficient statistic.

Proposition 1 (Sufficient-Statistic Invariance)**.** *Consider the transformer in Definition 1 with known action features $\mathcal{X}$. At each time t, an action A_t is selected and a reward R_t is observed. i) If the transformer's latent reward likelihood is context-independent, namely $p(R_t = r \mid D_{t-1}, x_{A_t}, \theta) = p(R_t = r \mid x_{A_t}, \theta)$, then its action distribution remains invariant to any permutation of the context,*

$$\pi_{\mathrm{TF}}(a \mid D_n) = \pi_{\mathrm{TF}}(a \mid \sigma_1(D_n)),\ \forall a \in \mathcal{A}, \forall n \in \mathbb{N}, \tag{6}$$

where $\sigma_1(\cdot)$ denotes a permutation transformation. ii) If, in addition, the transformer's latent reward likelihood takes the exponential-family form (5)*, then its action distribution depends on the context only through $S(D_n)$. Accordingly,*

$$\pi_{\mathrm{TF}}(a \mid D_n) = \pi_{\mathrm{TF}}(a \mid \sigma_2(D_n)),\ \forall a \in \mathcal{A}, \forall n \in \mathbb{N}, \tag{7}$$

where $\sigma_2(\cdot)$ denotes a transformation of the contexts that preserves the sufficient statistic.

Note that permutation transformations preserve the sufficient statistic, so (6) is a special case of (7). Proposition 1 implies that if a transformer behaves as a Bayesian posterior sampler as in Definition 1, its action distribution given any context D should coincide with that given another context D' whenever D and D' share the same sufficient statistic. We now turn to the martingale property.

3.1.2 MARTINGALE PROPERTY

Proposition 2 (Martingale Property)**.** *Consider the decision-making problem and the transformer in Definition 1. Given a context $D_n = \big((A_1, R_1), \ldots, (A_n, R_n)\big)$ and the sigma-algebra $\mathcal{F}_n$ generated by D_n, let $P_{\mathrm{TF}}^{(m)}(\cdot \mid D_n)$ denote the distribution of length-m continuations generated autoregressively by the transformer's own future action and reward predictions. It then follows that*

$$\mathbb{E}_{D_{n+1:n+m} \sim P_{\mathrm{TF}}^{(m)}(\cdot \mid D_n)} \left[\pi_{\mathrm{TF}}(a \mid D_{n+m}) \mid \mathcal{F}_n\right] = \pi_{\mathrm{TF}}(a \mid D_n),\ \forall a \in \mathcal{A}, \forall m, n \in \mathbb{N}. \tag{8}$$

In other words, $\{\pi_{\mathrm{TF}}(a \mid D_n)\}_{n \geq 0}$ is a martingale with respect to $\{\mathcal{F}_n\}_{n \geq 0}$.

Given a context, Proposition 2 implies that if the transformer behaves as a Bayesian posterior sampler and its future continuations are generated according to its own Bayesian posterior-predictive distribution, then the action distribution conditioned on these future continuations coincides in expectation with the action distribution conditioned on the current given context. Notably, Proposition 1 relies on specific conditions about the transformer's latent Bayesian observational model, such as a context-independent and finite-dimensional exponential-family likelihood, whereas Proposition 2 requires access to the transformer's Bayesian posterior-predictive distribution. While Proposition 1 can be tested for DPT provided that its underlying observational model satisfies the required conditions, the martingale property in Proposition 2 cannot be directly tested for DPT. This is because DPT is pretrained to predict actions only and therefore cannot generate future continuations containing both actions and rewards. To address this limitation, we introduce two DPT variants in the next subsection that satisfy the conditions of Proposition 1 by construction and can generate future continuations according to their own Bayesian posterior-predictive distributions.

3.2 DPT VARIANTS

The two proposed DPT variants, DPT′ and GDPT, retain DPT's transformer backbone and incorporate an additional reward head to jointly predict actions *and* their associated rewards. DPT′ and GDPT differ only in their supervision labels, as described below.

DPT′. This variant is identical to DPT except that it extends DPT's optimal action prediction to a joint prediction of the optimal action and its associated reward. The training objective for DPT′ is the cross-entropy loss for both the action head and the reward head.

GDPT. This variant employs the same architecture as DPT′. However, rather than predicting the optimal action and its associated reward as in DPT′, GDPT is trained to autoregressively predict the next action-reward pair in the context. Consequently, GDPT does not require optimal actions and rewards as supervision labels. On the other hand, GDPT requires expert contexts for training, such as Thompson sampling trajectories, whereas DPT and DPT′ can be trained on arbitrary contexts, including both random and expert contexts (Lee et al., 2023). This distinction arises from the availability of optimal supervision labels for DPT and DPT′.

It is worth noting that both DPT′ and GDPT satisfy the conditions of Proposition 1 by construction. Specifically, both variants employ Gaussian reward heads whose structures mirror those of the action heads, which are supposed to be context-independent under Definition 1. Moreover, given a context, both DPT′ and GDPT can generate future continuations according to their respective posterior-predictive distributions, as required by Proposition 2.

3.3 STATISTICAL HYPOTHESIS TESTING FRAMEWORK

Having established the two necessary conditions for a Bayesian posterior sampler (Propositions 1 and 2) and introduced the transformer-based ICRL schemes including DPT, DPT′, and GDPT, this subsection develops a statistical hypothesis testing framework to assess whether these ICRL schemes satisfy the two conditions in practice.

3.3.1 SUFFICIENT STATISTIC

Under the conditions of Proposition 1, a Bayesian posterior sampler should produce identical action distributions for contexts with the same sufficient statistics. Fix a context length n. Let $\mathcal{T}$ denote the distribution over environments and let $P_n(\cdot \mid \tau)$ denote the distribution of length-n contexts

collected in environment τ. We sample $\tau \sim \mathcal{T}$ and $D_n \sim P_n(\cdot \mid \tau)$, and independently draw a transformation $\sigma \sim \Sigma$, where Σ is supported on transformations satisfying $S(\sigma(D_n)) = S(D_n)$ (including $\sigma_1(\cdot)$ and $\sigma_2(\cdot)$ in Proposition 1). For each action $a \in \mathcal{A}$, define the paired difference

$$X_a := \pi_{\mathrm{TF}}(a \mid D_n) - \pi_{\mathrm{TF}}(a \mid \sigma(D_n)). \tag{9}$$

Sufficient-statistic invariance requires $X_a = 0$ for every action. However, testing the exact equality may yield a trivial rejection whenever there exists a nonzero discrepancy, regardless of its magnitude. To avoid the effects of noise and assess systematic deviations, we instead test a relaxed null hypothesis

$$H_0 : \mathbb{E}_{\tau\sim\mathcal{T},\, D_n\sim P_n(\cdot|\tau),\, \sigma\sim\Sigma}\left[X_a\right] = 0,\ \forall a \in \mathcal{A}, \forall n \in \mathbb{N}. \tag{10}$$

Importantly, we must choose Σ such that D and $\sigma(D)$ induce distinct distributions. Otherwise, the expectation in (10) trivially equal to 0 by symmetry. For details on how we guarantee this, see Appendix B.2.

3.3.2 Martingale Property

Given a fixed starting context D_n and a continuation length $m \geq 0$ in DPT′ and GDPT, let $P_{\mathrm{TF}}^{(m)}(\cdot \mid D_n)$ denote the distribution of length-m continuations generated autoregressively by their transformer's own future action and reward predictions. For each action $a \in \mathcal{A}$, define

$$Y_a := \pi_{\mathrm{TF}}(a \mid D_{n+m}) - \pi_{\mathrm{TF}}(a \mid D_n), \tag{11}$$

where the randomness in Y_a arises solely from the generated continuations $D_{n+1:n+m}$. Following Proposition 2, the null hypothesis is

$$H_0 : \mathbb{E}_{D_{n+1:n+m}\sim P_{\mathrm{TF}}^{(m)}(\cdot|D_n)}\left[Y_a\right] = 0,\ \forall a \in \mathcal{A}, \forall n \in \mathbb{N}. \tag{12}$$

We generate independent continuations from the same D_n to obtain conditionally i.i.d. increment vectors. Since $\sum_{a\in\mathcal{A}} Y_a = 0$, we apply the paired Hotelling test as described in Appendix C.1.

4 Experiments

This section presents experimental results for the two statistical tests developed in Section 3.3. We first describe the shared experimental setup, then present results for DPT, DPT′, and GDPT in Gaussian, Bernoulli, and linear bandits settings. Following these, we investigate the effect of the transformer's architecture through an ablation study.

4.1 Experimental Setup

Environments (Tasks). We consider three bandit problems: Gaussian, Bernoulli, and linear bandits. In Gaussian bandits, rewards satisfy $r \mid a \sim \mathcal{N}(\mu_a, \sigma^2)$ with unknown arm means $\{\mu_a\}$ and known variance σ^2. In Bernoulli bandits, rewards satisfy $r \mid a \sim \mathrm{Bernoulli}(p_a)$ with unknown success probabilities $\{p_a\}$. In linear bandits, each arm a is associated with a fixed, known feature vector $x_a \in \mathbb{R}^d$, and rewards satisfy $r \mid a \sim \mathcal{N}(x_a^\top \theta, \sigma^2)$ with an unknown shared parameter $\theta \in \mathbb{R}^d$ and known variance σ^2. Gaussian and Bernoulli bandits can be viewed as special cases of linear bandits.

Pretraining and Deployment. The dataset collection, pretraining, and deployment procedures for DPT, DPT′, and GDPT follow Section 2. For online deployment, the contexts are collected by the model itself. For offline deployment, the contexts are pre-specified by either a Dirichlet-mixture random policy or Thompson Sampling depending on the bandit setting, following (Lee et al., 2023).

Tests and Metrics. Under Proposition 1, a Bayesian posterior sampler with the assumed reward likelihood must be invariant to transformations preserving the sufficient statistic. For a context D_n, let N_a and S_a be the number of pulls and the total rewards of arm a. The sufficient statistic is $\{(N_a, S_a)\}_{a=1}^K$ for Gaussian and Bernoulli bandits, and $\left(V = \sum_a N_a x_a x_a^\top, b = \sum_a x_a S_a\right)$ for linear bandits (details in Appendix B.3). We test this invariance with two transformations: *sorting* reorders the action-reward pairs in ascending order of reward, *reward replacement* replaces each arm's rewards with its empirical mean plus centered uniform noise. Since Bernoulli rewards only take values in $\{0, 1\}$, we only use *sorting*. Each test evaluates the hypothesis in Eq. (10) on $N = 5000$ independently sampled environments and their contexts. For the martingale test, we fix one

starting context D_n, generate independent continuations of length m using the model's own future action and reward predictions, and test Eq. (12). Both tests are carried out with the paired Hotelling tests with significance level $\alpha = 0.05$ (see Appendix C.1). In addition, we compare cumulative regret, offline suboptimality, and the total variation (TV) distance of action distributions against Thompson Sampling, a canonical posterior sampling method.

Hyperparameters. Appendix C.2 provides a complete list of the hyperparameters used in our experiments, including those specifying the transformer's architecture and the experimental setup.

4.2 Experimental Results

Table 1: p-values for sufficient-statistic invariance tests on DPT at significance level $\alpha = 0.05$.

	Gaussian bandits				Bernoulli bandits	
	Offline		Online		Offline	Online
n	Sort	Replace	Sort	Replace	Sort	Sort
50	2.88×10^{-112}	6.95×10^{-230}	2.44×10^{-259}	$< 10^{-300}$	1.41×10^{-141}	4.79×10^{-80}
100	5.68×10^{-108}	$< 10^{-300}$	5.22×10^{-265}	$< 10^{-300}$	2.23×10^{-158}	5.83×10^{-133}
200	5.62×10^{-178}	$< 10^{-300}$	1.44×10^{-282}	$< 10^{-300}$	5.43×10^{-184}	4.36×10^{-161}
300	3.10×10^{-155}	$< 10^{-300}$	1.02×10^{-256}	$< 10^{-300}$	9.22×10^{-196}	1.63×10^{-167}
400	1.25×10^{-148}	$< 10^{-300}$	1.93×10^{-229}	$< 10^{-300}$	3.17×10^{-209}	4.13×10^{-177}
500	8.53×10^{-154}	$< 10^{-300}$	5.96×10^{-224}	$< 10^{-300}$	2.63×10^{-206}	1.33×10^{-187}

	Linear bandits			
	Offline		Online	
n	Sort	Replace	Sort	Replace
25	1.56×10^{-18}	7.60×10^{-7}	4.08×10^{-22}	2.31×10^{-4}
50	1.76×10^{-49}	1.65×10^{-12}	2.95×10^{-59}	1.31×10^{-8}
100	4.88×10^{-76}	1.21×10^{-21}	1.05×10^{-87}	4.79×10^{-26}
150	1.81×10^{-67}	1.74×10^{-42}	8.36×10^{-69}	7.27×10^{-33}
200	6.43×10^{-93}	6.38×10^{-45}	2.33×10^{-82}	6.94×10^{-40}

DPT. Table 1 reports the p-values of the sufficient-statistic tests for DPT on Gaussian, Bernoulli, and linear bandits, under both offline and online deployments and both the "Sort" and "Replace" transformations. Since the sufficient-statistic invariance should hold for arbitrary context lengths, we consider context lengths $n \in \{50, 100, 200, 300, 400, 500\}$ for Gaussian and Bernoulli bandits and $n \in \{25, 50, 100, 150, 200\}$ for linear bandits. Notably, all tests in Table 1 reject the null hypothesis in (10) at $\alpha = 0.05$. This provides evidence that DPT does not satisfy Proposition 1 and hence cannot be characterized as a Bayesian posterior sampler under Definition 1, provided that the latent reward likelihood of the transformer is context-independent and belongs to the exponential family. We further compare DPT with Thompson Sampling in Figures 2, 3, and 4 for Gaussian, Bernoulli, and linear bandits, respectively.

DPT′. As described in Section 3.2, DPT′ satisfies the conditions of Proposition 1 by construction. In addition, it can generate future continuations from its own action–reward predictions, so that both tests apply. For space reasons, we present the martingale tests in Table 2 and defer the sufficient-statistic tests to Table 5 in Appendix D.2. The sufficient-statistic tests follow the same protocol as for DPT, and all of them reject the null hypothesis in (10) at $\alpha = 0.05$. For the martingale tests, we fix one starting context D_n for each bandit setting and deployment mode, and generate $N = 5000$ independent continuations of length m from DPT′. Since the martingale property in (8) should hold for arbitrary n and m, we consider starting-context lengths $n \in \{50, 100, 200, 300, 400\}$ with continuation lengths m satisfying $n + m \leq 500$ for Gaussian and Bernoulli bandits, and $n \in \{25, 50, 100, 150\}$ with $n + m \leq 200$ for linear bandits. All martingale tests in Table 2 reject the null hypothesis in (12) at $\alpha = 0.05$. In other words, the current action probabilities of DPT′ differ systematically from their expected values after model-generated continuations, in violation of Proposition 2. Per-arm increment curves are provided in Appendix D.2. Performance comparisons with Thompson Sampling are presented in Figures 5, 8, and 11. In Gaussian bandits, offline DPT′ attains suboptimality close to that of Thompson Sampling while its TV distance remains positive.

Table 2: p-values for martingale tests on DPT′ at significance level $\alpha = 0.05$.

		Gaussian		Bernoulli	
n	m	Offline	Online	Offline	Online
50	100	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$
	200	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$
	300	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$
	400	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$
	450	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$
100	100	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$
	200	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$
	300	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$
	400	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$
200	100	$< 10^{-300}$	1.45×10^{-55}	$< 10^{-300}$	1.51×10^{-157}
	200	$< 10^{-300}$	1.91×10^{-82}	$< 10^{-300}$	1.29×10^{-130}
	300	$< 10^{-300}$	4.09×10^{-119}	$< 10^{-300}$	1.46×10^{-113}
300	100	$< 10^{-300}$	1.83×10^{-204}	$< 10^{-300}$	1.24×10^{-91}
	200	$< 10^{-300}$	3.75×10^{-12}	$< 10^{-300}$	1.12×10^{-76}
400	100	$< 10^{-300}$	1.67×10^{-250}	$< 10^{-300}$	$< 10^{-300}$

		Linear bandits	
n	m	Offline	Online
25	50	$< 10^{-300}$	$< 10^{-300}$
	100	$< 10^{-300}$	$< 10^{-300}$
	150	$< 10^{-300}$	$< 10^{-300}$
	175	$< 10^{-300}$	$< 10^{-300}$
50	50	$< 10^{-300}$	$< 10^{-300}$
	100	$< 10^{-300}$	$< 10^{-300}$
	150	$< 10^{-300}$	$< 10^{-300}$
100	50	$< 10^{-300}$	$< 10^{-300}$
	100	$< 10^{-300}$	$< 10^{-300}$
150	50	$< 10^{-300}$	$< 10^{-300}$

Table 3: p-values for martingale tests on GDPT at significance level $\alpha = 0.05$.

		Gaussian		Bernoulli	
n	m	Offline	Online	Offline	Online
50	100	6.85×10^{-110}	2.89×10^{-248}	2.02×10^{-5}	0.099
	200	1.33×10^{-192}	$< 10^{-300}$	0.004	0.014
	300	1.20×10^{-61}	2.04×10^{-157}	3.09×10^{-4}	0.028
	400	1.90×10^{-53}	9.77×10^{-121}	0.002	0.042
	450	5.02×10^{-57}	8.08×10^{-246}	0.001	0.013
100	100	5.86×10^{-177}	$< 10^{-300}$	8.13×10^{-11}	3.01×10^{-26}
	200	1.09×10^{-20}	$< 10^{-300}$	2.15×10^{-11}	0.021
	300	5.85×10^{-78}	2.39×10^{-30}	7.22×10^{-6}	4.49×10^{-4}
	400	2.89×10^{-79}	1.08×10^{-94}	3.91×10^{-10}	0.094
200	100	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	3.08×10^{-34}
	200	2.68×10^{-10}	$< 10^{-300}$	$< 10^{-300}$	1.27×10^{-79}
	300	2.56×10^{-46}	$< 10^{-300}$	$< 10^{-300}$	4.92×10^{-24}
300	100	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$
	200	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	2.02×10^{-106}
400	100	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$	$< 10^{-300}$

		Linear bandits	
n	m	Offline	Online
25	50	3.07×10^{-11}	4.73×10^{-4}
	100	2.37×10^{-30}	1.51×10^{-7}
	150	1.79×10^{-15}	5.65×10^{-11}
	175	3.77×10^{-35}	1.38×10^{-16}
50	50	1.27×10^{-170}	$< 10^{-300}$
	100	$< 10^{-300}$	$< 10^{-300}$
	150	$< 10^{-300}$	$< 10^{-300}$
100	50	$< 10^{-300}$	$< 10^{-300}$
	100	$< 10^{-300}$	$< 10^{-300}$
150	50	$< 10^{-300}$	$< 10^{-300}$

GDPT. GDPT, likewise, satisfies the conditions of Proposition 1 by construction, and both tests follow the same protocol as for DPT′. All sufficient-statistic tests in Table 6 in Appendix D.3 reject the null hypothesis in (10) at $\alpha = 0.05$. Hence, GDPT also cannot be characterized as a Bayesian posterior sampler under Definition 1. Table 3 reports the martingale tests. All Gaussian and linear tests, for both offline and online starting contexts, and all Bernoulli tests with offline starting contexts reject at $\alpha = 0.05$. For Bernoulli bandits with online starting contexts, 13 of the 15 tests reject the null hypothesis, with two exceptions $(n, m) = (50, 100)$ and $(100, 400)$ at p-values of 0.099 and 0.094, respectively. Thus, the null hypothesis in (12) is rejected in all but two of the evaluated settings, indicating systematic violations of Proposition 2 at the tested contexts. Per-arm increment curves are provided in Appendix D.3. Performance comparisons with Thompson Sampling are presented in Figures 14, 17, and 20. In Gaussian bandits, the cumulative regret and offline suboptimality of GDPT closely follow those of Thompson Sampling, whereas its offline TV distance remains positive. Together with the results for DPT′, this demonstrates that decision performance comparable to Thompson Sampling can coexist with violations of the necessary conditions for Bayesian posterior sampling, and therefore does not by itself imply posterior-sampling behavior.

4.3 Ablation Study

The ablation study investigates whether the observed violations result primarily from the transformer's sensitivity to the order of the context. To this end, we modify the DPT architecture to be permutation-invariant by removing the positional encoding, replacing causal attention with bidirectional attention, and applying mean pooling at the output layer. This modification applies to DPT and DPT′, whose supervision labels are the optimal action (reward) given the entire context, but not to GDPT, whose autoregressive objective requires causal attention to prevent each prediction from accessing its own target. We therefore evaluate DPT and DPT′ with the modified architecture in the Gaussian bandit under the same pretraining, deployment, and testing procedures as before. The sufficient-statistic tests are reported in Tables 7 and 8, and the martingale tests for DPT′ are in Table 9. For both models, the *sort* transformation now yields identical action-probability vectors at every evaluated context length, enforced by the modified architecture. These entries are marked as 'Equal" with no p-value reported. However, all *reward replacement* tests and all martingale tests still reject the null hypotheses. Thus, enforcing permutation invariance restores sorting invariance by construction but still fails the sufficient-statistic invariance and the martingale property, indicating that the sensitivity to the context order is not the main reason for the observed violations. Performance comparisons and per-arm martingale increment curves for the ablated models are provided in Appendix D.4.

5 Conclusion

In this paper, we investigate whether ICRL, specifically DPT and two variants DPT′ and GDPT, behave as Bayesian posterior samplers in practice, where the idealized assumptions underpinning its theoretical characterization may not hold. We establish two necessary conditions that any Bayesian posterior sampler must satisfy: sufficient-statistic invariance and the martingale property. Then we design statistical hypothesis tests for both conditions. Experiments on Gaussian, Bernoulli, and linear bandits show that DPT, DPT′, and GDPT consistently violate the sufficient-statistic invariance. Additionally, DPT′ and GDPT violate the martingale property under their own model-generated continuations, even when their decision performance is comparable to that of Thompson Sampling. The ablation study further exhibits that enforcing permutation invariance in the architecture restores invariance to the context order but recovers neither condition, indicating that the sensitivity to the context order is not the main source of these violations. Beyond these negative findings, our tests rely only on observable action predictions and thus provide a principled toolkit for examining Bayesian claims about ICRL. Extending the tests to decision-making problems with state dependence, and understanding what architectural and training choices are needed to satisfy the two conditions, are important directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with polishing writings. We did not use generative AI tools to clean and reformat datasets, implement methods, to generate synthetic datasets, develop the theoretical models or conceptual framework, formulate mathematical claims, write proofs or provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on the research methodology or experiments, support qualitative or thematic data analysis, or interpret results; translation is not applicable to this work. Additionally, we used generative AI tools to polish the language, to edit the manuscript for clarity and readability, and to check notational consistency. All AI-assisted content was reviewed by the authors, who take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

M Mehdi Afsar, Trafford Crump, and Behrouz Far. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 55(7):1–38, 2023.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://papers.nips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html.

Weiqin Chen and Santiago Paternain. Random policy enables in-context reinforcement learning within trust horizons. *arXiv preprint arXiv:2410.19982*, 2024.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 1107–1128, 2024.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pp. 1329–1338. PMLR, 2016.

Fabian Falck, Ziyu Wang, and Christopher C. Holmes. Is in-context learning in large language models Bayesian? A martingale perspective. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12784–12805. PMLR, 2024. URL https://proceedings.mlr.press/v235/falck24a.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://papers.nips.cc/paper/2021/hash/099fe6b0b444c23836c4a5d07346082b-Abstract.html.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hy0a5MMPUv.

Jonathan N. Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8644b61a9bc87bf7844750a015feb600-Abstract-Conference.html.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International conference on machine learning*, pp. 19565–19594. PMLR, 2023.

Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.08566.

Yuanguo Lin, Yong Liu, Fan Lin, Lixin Zou, Pengcheng Wu, Wenhua Zeng, Huanhuan Chen, and Chunyan Miao. A survey on reinforcement learning for recommender systems. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. URL https://arxiv.org/abs/2303.08774.

OpenAI. GPT-3.5 Turbo model. OpenAI API documentation, n.d. URL https://developers.openai.com/api/docs/models/gpt-3.5-turbo. Accessed September 19, 2026.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of Thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. URL https://arxiv.org/abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Tomoya Wakayama and Taiji Suzuki. In-context learning is provably Bayesian inference: A generalization theory for meta-learning. *arXiv preprint arXiv:2510.10981*, 2025. URL https://arxiv.org/abs/2510.10981.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.02080.

A Related Work

In-context reinforcement learning. Transformers have been successfully applied to RL by formulating decision-making as a sequence modeling problem. Two pioneering works, Decision Transformer and Trajectory Transformer, take different approaches. Decision Transformer predicts actions conditioned on desired returns and past interactions (Chen et al., 2021), while Trajectory Transformer models trajectories autoregressively and performs planning via beam search (Janner et al., 2021). In-context reinforcement learning (ICRL) takes this paradigm a step further by studying how pretrained transformers can adapt to new, unseen tasks without any parameter updates. Laskin et al. (2023) introduced Algorithm Distillation (AD), which pretrains a causal transformer on the learning histories generated by a source RL algorithm, with the goal of distilling its policy improvement operator across episodes. Lee et al. (2023) subsequently proposed the Decision-Pretrained Transformer (DPT), which predicts optimal actions given a query state and arbitrary contexts of varying lengths. DPT typically relies on optimal policies, such as well-trained PPO policies or human-designed policies, to provide supervision labels. A central theoretical grounding for DPT is that, under idealized assumptions such as a sufficiently expressive model and infinite pretraining data, it behaves as a Bayesian posterior sampler. Building on this perspective, we investigate whether DPT and our proposed DPT variants satisfy necessary conditions for the Bayesian posterior sampler in practice.

Is in-context learning Bayesian? A prominent line of work examines whether in-context learning (ICL) can be characterized as Bayesian inference. Xie et al. (2022) model pretraining data as a mixture of hidden Markov models indexed by a latent concept and demonstrate that the pretrained predictor implicitly infers the concept shared across the prompt, becoming asymptotically Bayes-optimal as the number of in-context examples grows. Wakayama & Suzuki (2025) provide a finite-sample counterpart for meta-learning over a mixture of tasks. They decompose the ICL risk into a Bayes gap and an irreducible posterior variance, and bound the former non-asymptotically for uniform-attention transformers trained by empirical risk minimization. These results establish Bayes-optimality under idealized generative assumptions, while whether trained models actually behave in a Bayesian manner remains an empirical question. Falck et al. (2024) address it by observing that Bayesian predictive distributions over exchangeable data must satisfy a martingale property, and derive testable diagnostics from it. Applied to pretrained LLMs on synthetic tasks, these diagnostics reveal systematic violations. Following this diagnostic perspective, this work statistically tests whether ICRL behaves as a Bayesian posterior sampler.

Bayesian posterior sampling in decision-making. Bayesian posterior sampling, originating with (Thompson, 1933), selects actions by drawing an environment from the current posterior and acting optimally for that draw. Russo & Van Roy (2016) give an information-theoretic analysis of Thompson sampling, bounding Bayesian regret through the information ratio, which trades off instantaneous regret against information gained about the optimal action, and obtaining regret bounds for finite-armed and linear bandits. Osband et al. (2013) extend posterior sampling to episodic Markov decision processes (MDPs) with Bayesian regret guarantees, which samples an MDP from the posterior at the start of each episode and following its optimal policy. These works characterize the exact posterior sampling, whereas this paper explores whether ICRL schemes can be viewed as posterior samplers in practice.

B THEORY AND PROOFS

B.1 PROOFS OF PROPOSITIONS 1 AND 2

Proof of Proposition 1. Since we assumed the reward likelihood to be context independent, the joint context likelihood is proportional to the product of each reward likelihood, i.e.,

$$p(D_n \mid \theta) \propto \prod_{t=1}^{n} p(R_t \mid x_{A_t}, \theta). \tag{13}$$

Therefore, the posterior takes the form:

$$p_{\pi_0}(\theta \mid D_n) \propto \pi_0(\theta) \prod_{t=1}^{n} p(R_t \mid x_{A_t}, \theta).$$

For part $i)$, permuting the action–reward pairs leaves the product of reward likelihoods unchanged. Thus D_n and $\sigma_1(D_n)$ induce the same posterior.

For part $ii)$, substituting (5) into (13) gives

$$p(D_n \mid \theta) \propto \exp\left\{\eta(\theta)^\top S_n^{(1)} - \lambda(\theta)^\top S_n^{(2)}\right\}.$$

It then follows from the Fisher–Neyman factorization theorem that $S(D_n) = (S_n^{(1)}, S_n^{(2)})$ is sufficient for θ. In particular, the posterior takes the form

$$p_{\pi_0}(\theta \mid D_n) \propto \pi_0(\theta) \exp\left\{\eta(\theta)^\top S_n^{(1)} - \lambda(\theta)^\top S_n^{(2)}\right\}.$$

Hence $S(\sigma_2(D_n)) = S(D_n)$ implies that D_n and $\sigma_2(D_n)$ induce the same posteriors. For both parts, combining the same context-independent decision kernel $\mathcal{K}(a \mid \theta)$ and the same posteriors, as in Definition 1, leads to the equality of action distributions for every $a \in \mathcal{A}$. □

Proof of Proposition 2. Fix an action $a \in \mathcal{A}$. By assumption, the context continuations and the posteriors in Definition 1 arise from the same joint Bayesian model. Under this model,

$$\pi_{\mathrm{TF}}(a \mid D_t) = \int_\Theta \mathcal{K}(a \mid \theta) p_{\pi_0}(\theta \mid D_t)\, d\theta = \mathbb{E}\left[\mathcal{K}(a \mid \theta) \mid \mathcal{F}_t\right].$$

Since $0 \le \mathcal{K}(a \mid \theta) \le 1$, it is integrable. It then follows from the tower property that

$$\begin{aligned} \mathbb{E}\left[\pi_{\mathrm{TF}}(a \mid D_{n+m}) \mid \mathcal{F}_n\right] &= \mathbb{E}\left[\mathbb{E}\left[\mathcal{K}(a \mid \theta) \mid \mathcal{F}_{n+m}\right] | \mathcal{F}_n\right] && (14)\\ &= \mathbb{E}\left[\mathcal{K}(a \mid \theta) \mid \mathcal{F}_n\right] && (15)\\ &= \pi_{\mathrm{TF}}(a \mid D_n). && (16) \end{aligned}$$

Conditional on D_n, the law for the expectation on the left is precisely $P_{\mathrm{TF}}^{(m)}(\cdot \mid D_n)$, which proves (8). Thus $\{\pi_{\mathrm{TF}}(a \mid D_n)\}_{n \ge 0}$ is a martingale. This argument does not require an exponential-family likelihood or a sufficient statistic. □

B.2 Distinct Context Distributions after sufficient statistics-preserving transformations

Let $D = ((A_1, R_1), \ldots, (A_n, R_n))$ be an ordered context of length $n \geq 2$ from a K-armed bandit. The results below are independent of the exact interaction policy. Conditional on the history and $A_t = a$, the reward law is $\mathcal{N}(\mu_a, \sigma^2)$ for Gaussian bandits, $\mathrm{Bernoulli}(p_a)$ for Bernoulli bandits, and $\mathcal{N}(x_a^\top \theta, \sigma^2)$ for linear bandits with fixed known features x_a. We assume the variance $\sigma^2 > 0$ is known, and $p_a \in (0, 1)$.

Let $I_{\{A_t=a\}}$ be the indicator function of the event: the action taken at time t is a, then let

$$N_a = \sum_{t=1}^{n} I_{\{A_t=a\}} \text{ and } S_a = \sum_{t=1}^{n} I_{\{A_t=a\}} R_t.$$

The sufficient statistics are $\{(N_a, S_a)\}_{a=1}^K$ for Gaussian and Bernoulli bandits, and

$$V = \sum_{a=1}^{K} N_a x_a x_a^\top, \qquad b = \sum_{a=1}^{K} x_a S_a$$

for linear bandits (see Appendix B.3). Thus preserving every arm's count and reward sum suffices is equivalent to preserving the sufficient statistics in all three settings.

Proposition 3 (Context transformations)**.** *The following transformations produce a context D' with the same sufficient statistics as D but a different distribution:*

1. *Sorting (all three bandits): sort the action–reward pairs by ascending reward, retaining their original order within ties.*
2. *Reward replacement (Gaussian and linear bandits only): sample $U_t \sim^{i.i.d.} \mathrm{Uniform}[-c, c]$ independently of D for some fixed $c > 0$; for each observed arm a, set*

$$R'_t = \bar{R}_a + (U_t - \bar{U}_a) I_{\{A_t=a\}}, \quad \textit{where } \bar{R}_a = \frac{S_a}{N_a}, \quad \bar{U}_a = \frac{1}{N_a} \sum_{s=1}^{n} U_s I_{\{A_s=a\}}.$$

Actions remain unchanged. For this claim, suppose there exists an arm that is observed at least twice with positive probability[1], i.e., $\exists a : \mathbb{P}(\max_a N_a \geq 2) > 0$.

Proof. Both transformations preserve the arm counts. Sorting also preserves the arm reward sums, while replacement satisfies, for each observed arm a,

$$\sum_{t=1}^{n} I_{\{A_t=a\}} R'_t = N_a \bar{R}_a + \sum_{t=1}^{n} I_{\{A_t=a\}} (U_t - \bar{U}_a) = S_a.$$

Consequently, both transformations preserve the sufficient statistics.

For sorting, $\mathbb{P}(R'_1 > R'_2) = 0$. In Gaussian and linear bandits setting, conditional on the first observation and the second action, R_2 has a Gaussian distribution on $\mathbb{R}$, so $\mathbb{P}(R_1 > R_2) > 0$. For Bernoulli bandits, $\mathbb{P}(R_1 = 1) > 0$ and the conditional probability of $R_2 = 0$ is positive for every second action, so again $\mathbb{P}(R_1 > R_2) > 0$. This distinguishes the laws of D and D'.

For replacement, whenever $A_s = A_t$,

$$|R'_s - R'_t| = |U_s - U_t| \leq 2c.$$

The repetition assumption implies that some fixed $s < t$ satisfies $\mathbb{P}(A_s = A_t) > 0$. Conditional on D_{t-1} and A_t, the earlier reward R_s is fixed and R_t has a Gaussian distribution with full support. Therefore,

$$\mathbb{P}\big(A_s = A_t,\ |R_s - R_t| > 2c\big) > 0, \qquad \mathbb{P}\big(A_s = A_t,\ |R'_s - R'_t| > 2c\big) = 0.$$

These probabilities distinguish the laws in both Gaussian and linear bandits. □

[1] This can be easily established by choosing $n > K$

B.3 Sufficient Statistics for Bandits

We apply Proposition 1 to the three bandit settings in Section 4. We assume the transformer satisfies the assumptions of the proposition with the corresponding reward likelihood. For a context $D_n = \big((A_1, R_1), \ldots, (A_n, R_n)\big)$ and each arm $a \in \{1, \ldots, K\}$, let $I_{\{A_t=a\}}$ be the indicator variable for the event $\{A_t = a\}$ as in Appendix B.2. Similarly, define the arm count and reward sum:

$$N_a = \sum_{t=1}^{n} I_{\{A_t=a\}}, \qquad S_a = \sum_{t=1}^{n} I_{\{A_t=a\}} R_t.$$

Corollary 1 (Gaussian bandits)**.** *In the Gaussian bandits setting where we have $R_t \mid A_t = a \sim \mathcal{N}(\mu_a, \sigma^2)$ with unknown means $\mu = (\mu_1, \ldots, \mu_K)$ and known variance $\sigma^2 > 0$, the transformer's action distribution depends on D_n only through $\{(N_a, S_a)\}_{a=1}^K$.*

Proof. Up to a constant independent of μ, the context likelihood is

$$p(D_n \mid \mu) \propto \exp\left\{ \frac{1}{\sigma^2} \sum_{a=1}^{K} \mu_a S_a - \frac{1}{2\sigma^2} \sum_{a=1}^{K} \mu_a^2 N_a \right\}.$$

This takes the exponential-family form in Proposition 1, using the standard basis vector in $\mathbb{R}^K$ for x_a, $T(r, x) = rx$, and $\phi(x) = x$. The rest follows directly from the proposition. □

Corollary 2 (Bernoulli bandits)**.** *Here, we have $R_t \mid A_t = a \sim \text{Bernoulli}(p_a)$ with unknown success probabilities $p = (p_1, \ldots, p_K) \in (0, 1)^K$, the transformer's action distribution depends on D_n only through $\{(N_a, S_a)\}_{a=1}^K$.*

Proof. Up to a constant independent of p, the context likelihood is

$$p(D_n \mid p) \propto \prod_{a=1}^{K} p_a^{S_a} (1 - p_a)^{N_a - S_a}.$$

With x_a being the standard basis vector in $\mathbb{R}^K$, $T(r, x) = rx$, and $\phi(x) = x$, Proposition 1 directly implies the result. □

Corollary 3 (Linear bandits)**.** *Given known arm features $x_1, \ldots, x_K \in \mathbb{R}^d$ and rewards $R_t \mid A_t = a \sim \mathcal{N}(x_a^\top \theta, \sigma^2)$ with unknown parameter $\theta \in \mathbb{R}^d$ and known variance $\sigma^2 > 0$, the transformer's action distribution depends on D_n only through*

$$V = \sum_{t=1}^{n} x_{A_t} x_{A_t}^\top = \sum_{a=1}^{K} N_a x_a x_a^\top, \qquad b = \sum_{t=1}^{n} x_{A_t} R_t = \sum_{a=1}^{K} x_a S_a. \tag{17}$$

Proof. Up to a constant independent of θ, the context likelihood is

$$p(D_n \mid \theta) \propto \exp\left\{ \frac{\theta^\top b}{\sigma^2} - \frac{\theta^\top V \theta}{2\sigma^2} \right\}.$$

Apply Proposition 1 with $T(r, x) = rx$, $\eta(\theta) = \theta/\sigma^2$, $\phi(x) = \text{vec}(xx^\top)$, and $\lambda(\theta) = \text{vec}(\theta\theta^\top)/(2\sigma^2)$. The result follows directly from the proposition. □

C EXPERIMENTAL DETAILS

C.1 PAIRED HOTELLING TEST

Suppose we observe n i.i.d. pairs of random variables (X_i, X_i'), where each vector has $K \geq 2$ nonnegative components summing to 1. Define the paired difference

$$D_i = X_i - X_i', \quad \delta = \mathbb{E}[D_i].$$

The hypotheses are

$$H_0 : \delta = 0 \quad v.s. \quad H_1 : \delta \neq 0.$$

Since $\sum_{j=1}^{K} D_i^j = 0$, the covariance matrix of the full K-dimensional differences is singular. Therefore, we drop any one coordinate and set $p = K - 1$ and $Z_i = (D_i^1, ..., D_i^{K-1})$. Now, $\mathbb{E}[Z_i] = 0$ is equivalent to H_0 from above. Compute

$$\bar{Z} = \frac{1}{n} \sum_{i=1}^{n} Z_i,$$

$$S = \frac{1}{n-1} \sum_{i=1}^{n} (Z_i - \bar{Z})(Z_i - \bar{Z})^\top,$$

$$T^2 = n \bar{Z}^\top S^{-1} \bar{Z}.$$

Since S is consistent, the central limit theorem gives

$$T^2 \xrightarrow{d} \chi_p^2.$$

Thus an asymptotic level-α test rejects when

$$T^2 > \chi_{p,1-\alpha}^2, \quad p = K - 1.$$

In our setting, the observed pair (X, X') is simply $\big(\pi_{\mathrm{TF}}(\cdot \mid D), \pi_{\mathrm{TF}}(\cdot \mid \sigma(D))\big)$.

C.2 Experimental setup and hyperparameters

Table 4 summarizes the configurations for DPT, DPT′, and GDPT. Unless a model-specific setting is indicated, the three models share the listed values. The Gaussian architecture ablations are described below.

Table 4: Environment, model, and test hyperparameters. H denotes the maximum number of context transitions, excluding the query token; TS denotes Thompson sampling. The continuation grids and linear-test support convention are specified in the text.

Parameter	Gaussian	Bernoulli	Linear
Environment and data			
Number of arms K	5	5	10
Feature dimension d	—	—	2
Environment prior	$\mu_a \sim U[0,1]$	$p_a \sim \mathrm{Beta}(1,1)$	$\theta \sim \mathcal{N}(0, I_2/2)$
Reward distribution	$\mathcal{N}(\mu_a, 0.09)$	$\mathrm{Bernoulli}(p_a)$	$\mathcal{N}(x_a^\top \theta, 0.09)$
Context horizon H	500	500	200
Train / validation environments	80,000/20,000	80,000/20,000	800,000/200,000
DPT, DPT′ context policy	Dirichlet-mixture random policy		Linear TS
GDPT context policy	Gaussian TS	Beta–Bernoulli TS	Linear TS
Model and optimization			
Backbone	GPT-2; learned positions; causal attention		
Layers / heads / embedding size	4 / 4 / 32		
Dropout	0 (embedding, attention, and residual)		
Optimizer	AdamW		
Learning rate / weight decay	10^{-4} / 10^{-4}		
Training batch size	64		
Training epochs / checkpoint	400/400	400/400	100/100
Reward head: DPT′, GDPT	Gaussian mean	Bernoulli logit	Gaussian mean
Reward-loss weight	1 for DPT′ and GDPT; no reward loss for DPT		
Evaluation and statistical tests			
Performance environments	200 fresh environments		
Sufficient-statistic test size N	5000 independent environments and contexts		
Sufficient-statistic lengths n	$\{50, 100, 200, 300, 400, 500\}$		$\{25, 50, 100, 150, 200\}$
Context transformations	Sort; Replace	Sort	Sort; Replace
Replacement noise U_t	$U[-1,1]$	—	$U[-1,1]$
Martingale continuations N	5000 per fixed context (DPT′, GDPT)		
Martingale starting lengths n	$\{50, 100, 200, 300, 400\}$		$\{25, 50, 100, 150\}$
Test level / calibration	$\alpha = 0.05$; asymptotic paired Hotelling T^2		
Posterior draws for TV	5000 per nonempty context		
Posterior draws for offline choice	100	5000	5000

Environments and context collection. Arm parameters are drawn independently in the Gaussian and Bernoulli settings. In linear bandits, the ten feature vectors $x_a \sim \mathcal{N}(0, I_2/2)$ are drawn once (NumPy `RandomState(1234)`) and shared by all environments, each of which draws its own θ. The Gaussian reward standard deviation is $\sigma = 0.3$ in both Gaussian and linear bandits. The Dirichlet-mixture policy that collects DPT and DPT′ contexts in Gaussian and Bernoulli bandits draws $q \sim \mathrm{Dirichlet}(\mathbf{1}_K)$, a uniformly random arm j, and $c \sim \mathrm{Uniform}\{0, 0.1, \ldots, 1\}$, and samples every context action from $(1-c)\, q + c\, e_j$; all other contexts are TS histories. Offline evaluation contexts come from the same policy as each model's training contexts. GDPT collects $H+1$ transitions so that a next-pair target exists at every prefix $D_0, \ldots, D_H$.

Training objectives. DPT predicts the optimal action at every nonempty prefix. DPT′ predicts the optimal action together with an independent reward from that arm, and GDPT predicts the next TS action–reward pair, both at prefixes $D_0, \ldots, D_H$. The reward head outputs one parameter per arm, and the target action selects the parameter entering the reward loss: a Gaussian negative log-likelihood with fixed variance 0.09 for Gaussian and linear rewards, and binary cross-entropy for Bernoulli rewards, weighted equally with the action cross-entropy. All models additionally penalize

$\mathrm{KL}(\pi_{\mathrm{TS}}(\cdot \mid D_0) \,\|\, \pi_{\mathrm{TF}}(\cdot \mid D_0))$ with weight 1; the target is uniform for Gaussian and Bernoulli bandits and is induced by the fixed arm geometry under the linear prior. The learning rate is constant. Context shuffling permutes whole transitions for DPT and DPT′ in Gaussian and Bernoulli bandits only; TS-collected histories keep their temporal order. Seeds are listed in Table 4; the Gaussian and Bernoulli performance environments are generated from the data-collection stream after the training and validation sets, and the train–eval base seed also governs replacement noise and martingale context selection and continuations.

Reference policies and decision metrics. Gaussian TS uses independent $\mathcal{N}(0.5, 1/12)$ priors on the arm means, which match the first two moments of the environment prior but not its distribution; we do not assume the transformer learns this prior. Bernoulli TS uses $\mathrm{Beta}(1, 1)$ priors, and linear TS uses the exact Gaussian posterior for θ. Let ν_a be the true expected reward of arm a and $\nu^* = \max_a \nu_a$. Online cumulative regret is $\sum_{s=1}^{t}(\nu^* - \nu_{A_s})$, with actions sampled and rewards drawn from the environment; offline suboptimality is $\nu^* - \nu_{\widehat{a}}$, where $\widehat{a}$ is the most probable action (for TS, the most frequent optimal arm among the posterior draws in Table 4). On offline contexts both policies receive the same history, and we report

$$\mathrm{TV}(D_t) = \frac{1}{2}\sum_{a=1}^{K} |\pi_{\mathrm{TF}}(a \mid D_t) - \pi_{\mathrm{TS}}(a \mid D_t)|\,, \tag{18}$$

with π_{TS} estimated from 5000 posterior draws per nonempty context and computed analytically at D_0. Online TV compares the policies on their own histories, so it also reflects the contexts they collect.

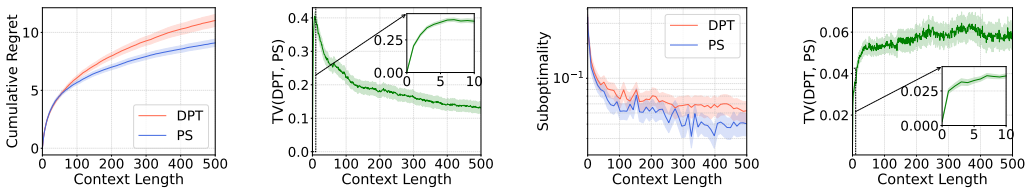
Statistical tests. Sufficient-statistic tests pair the original and transformed action-probability vectors over $N = 5000$ independent environment–context samples per deployment mode. Sorting is stable and ascending; reward replacement is the transformation of Proposition 3 with $U_t \sim U[-1, 1]$. Martingale tests use continuation lengths $m \in \{100, 200, 300, 400, 450\}$ at $n = 50$ and positive multiples of 100 up to $500 - n$ at $n \in \{100, 200, 300, 400\}$ for Gaussian and Bernoulli bandits, and $m \in \{50, 100, 150, 175\}$ at $n = 25$ and positive multiples of 50 up to $200 - n$ at $n \in \{50, 100, 150\}$ for linear bandits, with 5000 continuations sampled from the model's action and reward heads for each fixed D_n. Gaussian and Bernoulli starting contexts are nested prefixes of one offline and one model-collected online history. Linear starting contexts are selected per deployment mode and n from at most 200 candidate environments, using 500 (DPT′) or 1000 (GDPT) pilot continuations that are discarded afterwards, so that the test covariance is full rank at every reported m; linear increment plots show the first 200 continuations. Each reported p-value is a single joint Hotelling test at level 0.05 without multiplicity correction. Gaussian and Bernoulli tests drop one coordinate and use χ^2_{K-1}. Linear tests restrict to the convex-hull vertices $\mathcal{V}$ of the arm features, the only arms that can be uniquely optimal, drop one of them, and use $\chi^2_{|\mathcal{V}|-1}$, after checking that no interior arm is empirically optimal and that interior-arm differences have negligible variance (standard deviation at most 10^{-6} times the smallest vertex-arm standard deviation).

Architecture ablations. The Gaussian DPT and DPT′ ablations keep the data and optimization settings above, remove positional embeddings, replace causal with bidirectional attention, and at inference canonically order the context tokens and mean-pool their hidden states (excluding the query token), which makes predictions permutation invariant; each prefix is evaluated separately. They train for 400 epochs and evaluate the epoch-150 checkpoint. Exact equality under sorting is reported as "Equal" without a p-value. GDPT is not ablated because its next-pair objective requires causal attention.

D ADDITIONAL TABLES AND PLOTS

D.1 DPT FIGURES

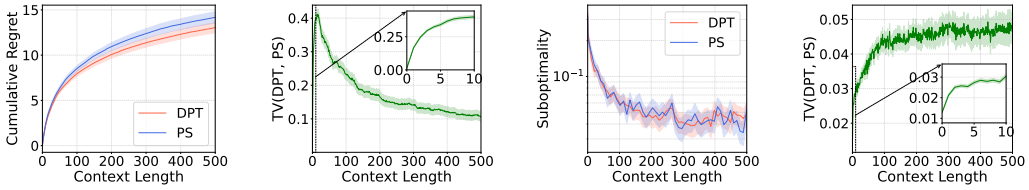
D.1.1 GAUSSIAN EVALUATION FIGURES



(a) Cumulative Regret (b) TV Distance (c) Suboptimality (d) TV Distance

Figure 2: DPT: Gaussian bandit

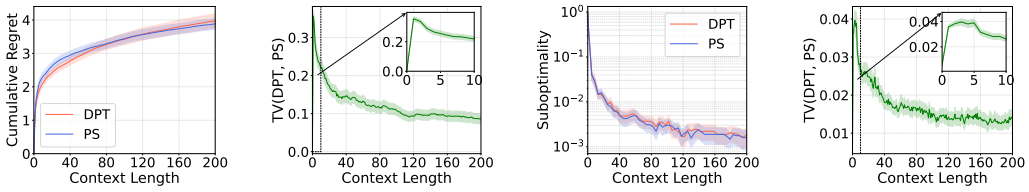
D.1.2 BERNOULLI EVALUATION FIGURES



(a) Cumulative Regret (b) TV Distance (c) Suboptimality (d) TV Distance

Figure 3: DPT: Bernoulli bandit

D.1.3 LINEAR EVALUATION FIGURES



(a) Cumulative Regret (b) TV Distance (c) Suboptimality (d) TV Distance

Figure 4: DPT: linear bandit

D.2 DPT′ SUFFICIENT-STATISTICS TABLE AND EVALUATION FIGURES

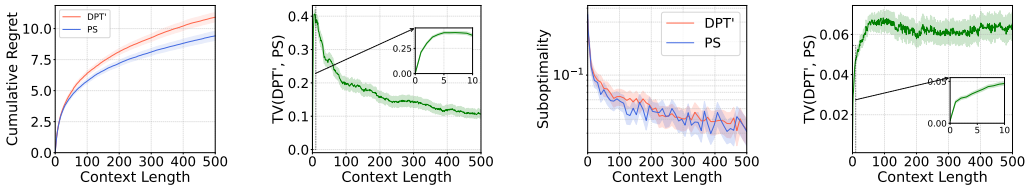
D.2.1 SUFFICIENT-STATISTICS TABLE

Table 5: DPT′: p-values for sufficient-statistic invariance tests.

	Gaussian				Bernoulli	
	Offline		Online		Offline	Online
n	Sort	Replace	Sort	Replace	Sort	Sort
50	2.31×10^{-105}	9.28×10^{-19}	$< 10^{-300}$	6.36×10^{-205}	1.67×10^{-196}	2.17×10^{-215}
100	3.09×10^{-194}	6.26×10^{-60}	$< 10^{-300}$	$< 10^{-300}$	5.66×10^{-160}	4.75×10^{-237}
200	7.78×10^{-157}	1.04×10^{-126}	3.59×10^{-265}	1.35×10^{-285}	1.08×10^{-125}	8.26×10^{-250}
300	6.67×10^{-137}	1.42×10^{-156}	2.58×10^{-226}	3.05×10^{-239}	2.00×10^{-120}	5.46×10^{-229}
400	8.63×10^{-133}	1.37×10^{-188}	1.88×10^{-205}	1.02×10^{-193}	3.39×10^{-112}	2.36×10^{-226}
500	7.34×10^{-139}	9.23×10^{-216}	2.56×10^{-222}	3.69×10^{-189}	2.08×10^{-111}	3.16×10^{-216}

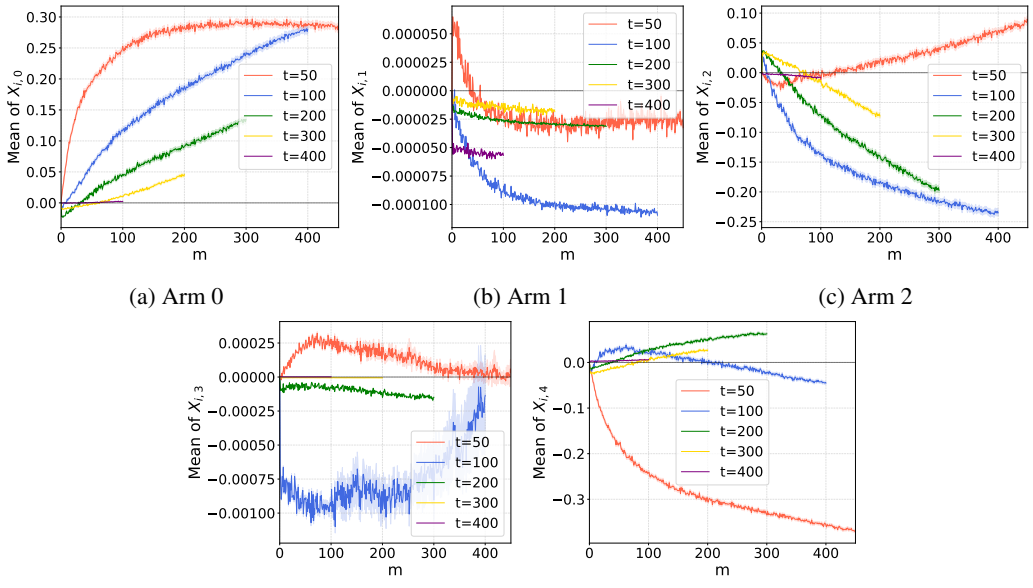
	Linear			
	Offline		Online	
n	Sort	Replace	Sort	Replace
25	1.96×10^{-102}	5.71×10^{-28}	1.15×10^{-86}	4.19×10^{-29}
50	6.30×10^{-96}	2.33×10^{-27}	7.27×10^{-93}	2.22×10^{-28}
100	9.44×10^{-108}	9.99×10^{-18}	3.07×10^{-99}	1.10×10^{-21}
150	1.78×10^{-93}	3.22×10^{-16}	8.76×10^{-84}	3.62×10^{-14}
200	2.68×10^{-83}	1.17×10^{-11}	3.13×10^{-79}	2.73×10^{-11}

D.2.2 GAUSSIAN EVALUATION FIGURES AND MARTINGALE PLOTS



(a) Cumulative Regret (b) TV Distance (c) Suboptimality (d) TV Distance

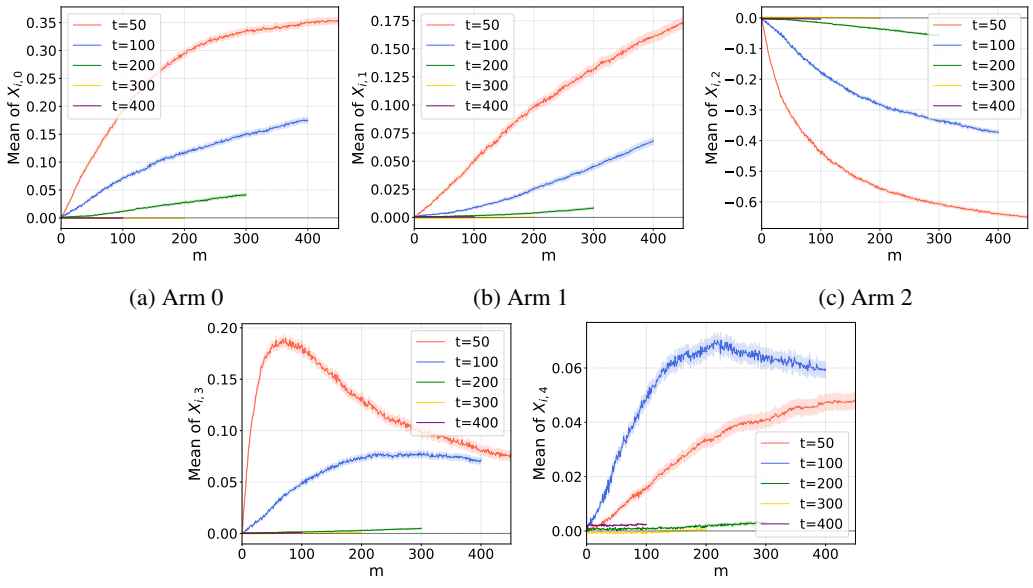
Figure 5: DPT′: Gaussian bandit



(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

Figure 6: DPT′: Gaussian bandit martingale increments, offline context.

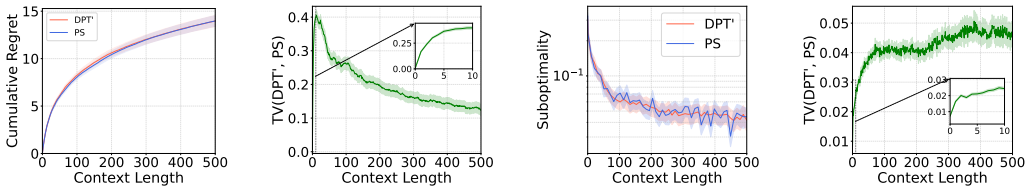


(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

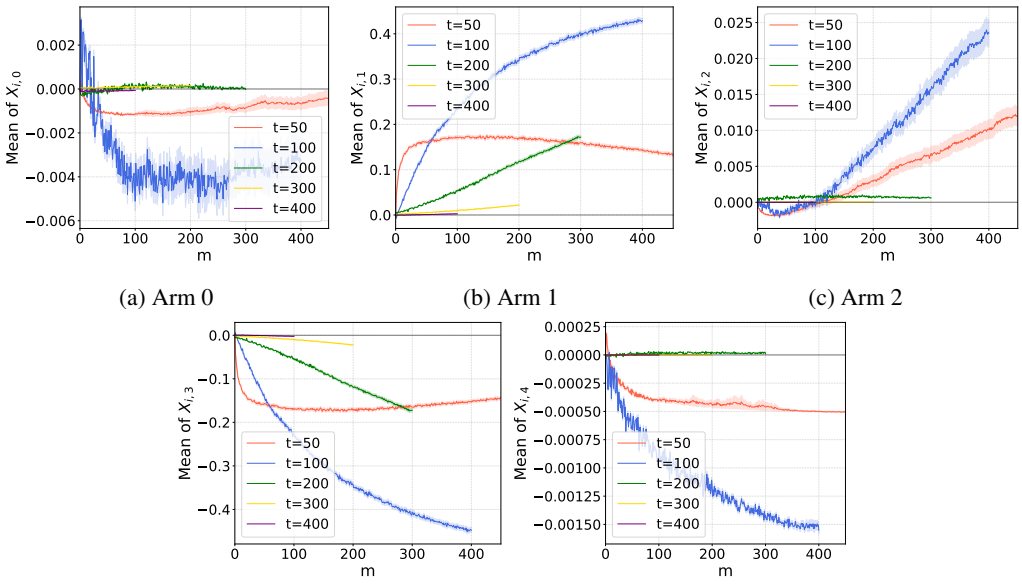
Figure 7: DPT′: Gaussian bandit martingale increments, online context.

D.2.3 BERNOULLI EVALUATION FIGURES AND MARTINGALE PLOTS



(a) Cumulative Regret (b) TV Distance (c) Suboptimality (d) TV Distance

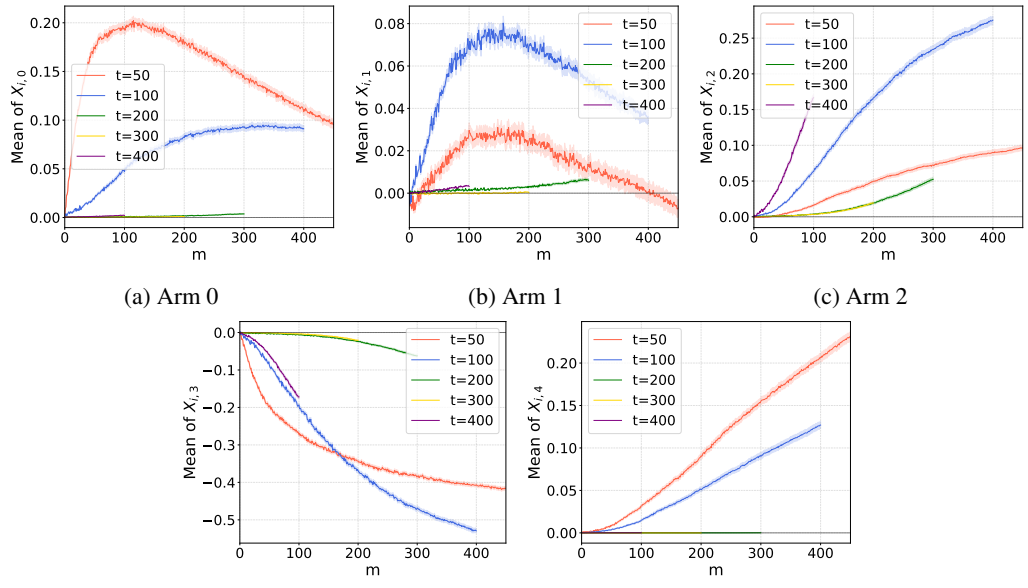
Figure 8: DPT′: Bernoulli bandit



(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

Figure 9: DPT′: Bernoulli bandit martingale increments, offline context.

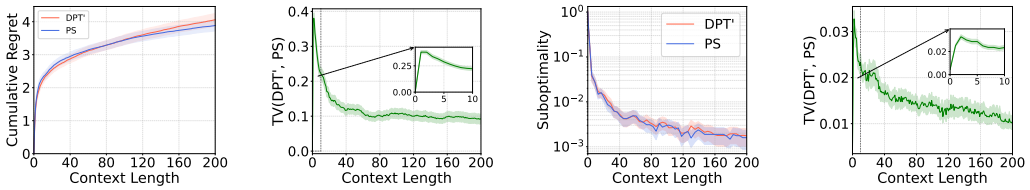


(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

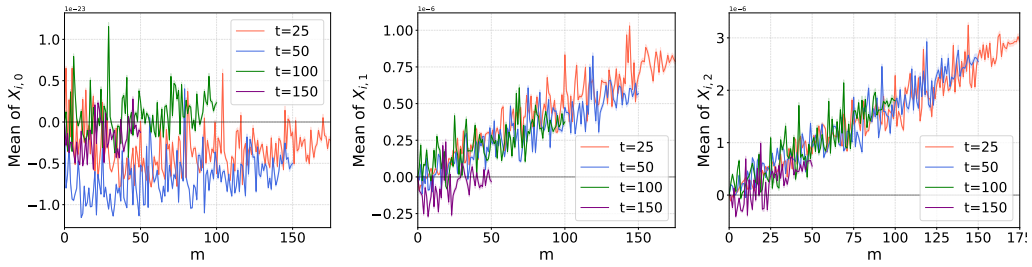
Figure 10: DPT′: Bernoulli bandit martingale increments, online context.

D.2.4 LINEAR EVALUATION FIGURES AND MARTINGALE PLOTS

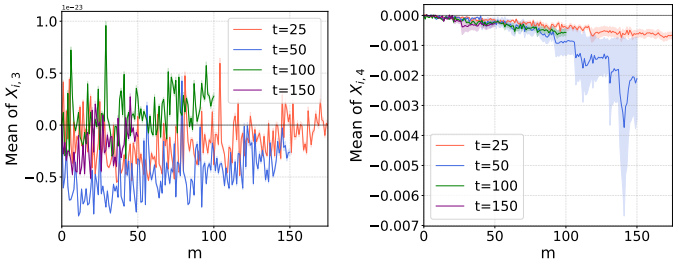


(a) Cumulative Regret (b) TV Distance (c) Suboptimality (d) TV Distance

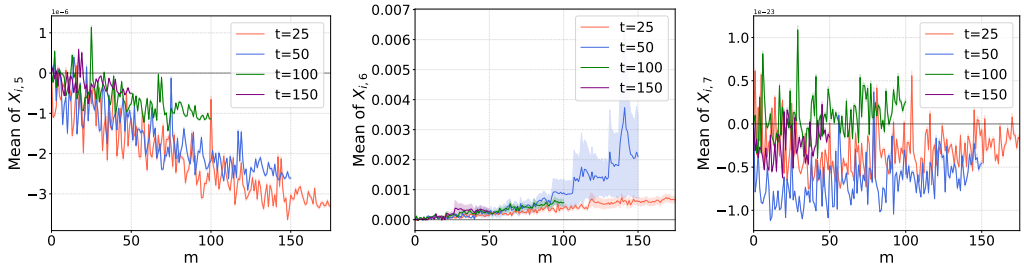
Figure 11: DPT′: linear bandit



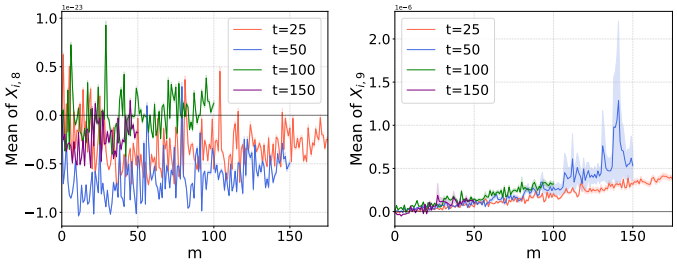
(a) Arm 0 (b) Arm 1 (c) Arm 2



(d) Arm 3 (e) Arm 4

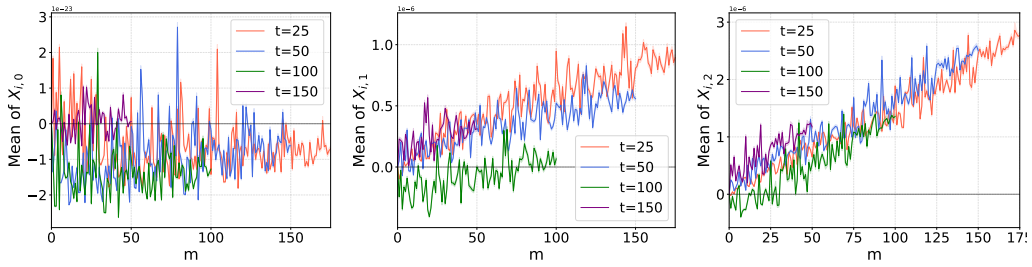


(f) Arm 5 (g) Arm 6 (h) Arm 7

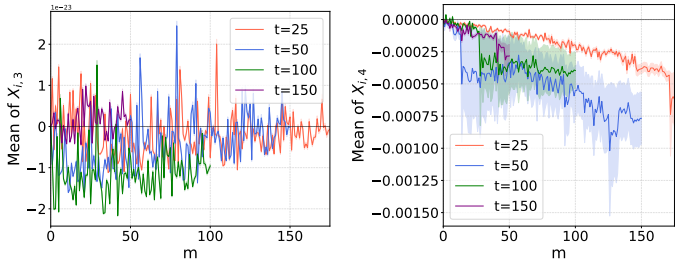


(i) Arm 8 (j) Arm 9

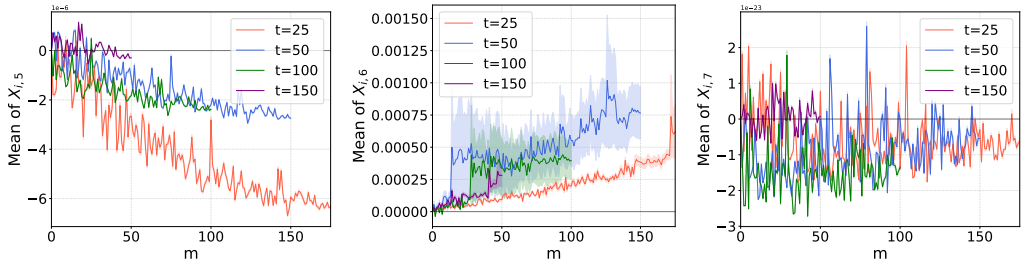
Figure 12: DPT′: Linear bandit martingale increments, offline context.



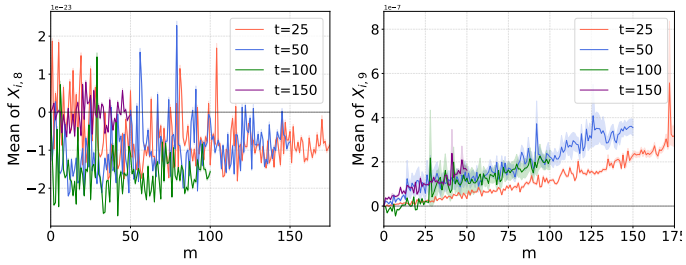
(a) Arm 0 (b) Arm 1 (c) Arm 2



(d) Arm 3 (e) Arm 4



(f) Arm 5 (g) Arm 6 (h) Arm 7



(i) Arm 8 (j) Arm 9

Figure 13: DPT′: Linear bandit martingale increments, online context.

D.3 GDPT SUFFICIENT-STATISTICS TABLE AND EVALUATION FIGURES

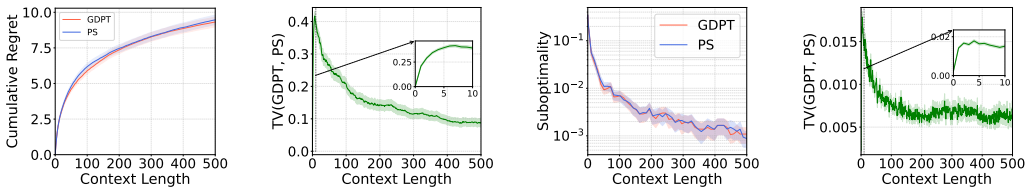
D.3.1 SUFFICIENT-STATISTICS TABLE

Table 6: GDPT: p-values for sufficient-statistic invariance tests at each context length n ($N = 5000$, $\alpha = 0.05$). Gaussian and Bernoulli bandits use $K = 5$; linear bandits use $K = 10$. Sort and Replace denote the sorting and reward-replacement transformations described in the text. Bernoulli tests use sorting.

	Gaussian				Bernoulli	
	Offline		Online		Offline	Online
n	Sort	Replace	Sort	Replace	Sort	Sort
50	$< 10^{-300}$	1.53×10^{-106}	$< 10^{-300}$	2.50×10^{-80}	1.36×10^{-127}	4.33×10^{-125}
100	3.40×10^{-233}	6.19×10^{-73}	7.93×10^{-208}	7.01×10^{-60}	3.05×10^{-81}	1.94×10^{-81}
200	1.55×10^{-134}	1.79×10^{-59}	4.62×10^{-135}	7.43×10^{-43}	2.07×10^{-26}	1.06×10^{-20}
300	6.21×10^{-128}	8.78×10^{-102}	4.39×10^{-132}	1.79×10^{-106}	1.78×10^{-25}	1.50×10^{-18}
400	8.54×10^{-114}	7.85×10^{-109}	1.72×10^{-116}	2.18×10^{-113}	1.26×10^{-34}	1.53×10^{-31}
500	3.38×10^{-92}	5.25×10^{-128}	1.45×10^{-102}	3.10×10^{-140}	5.24×10^{-50}	2.27×10^{-47}

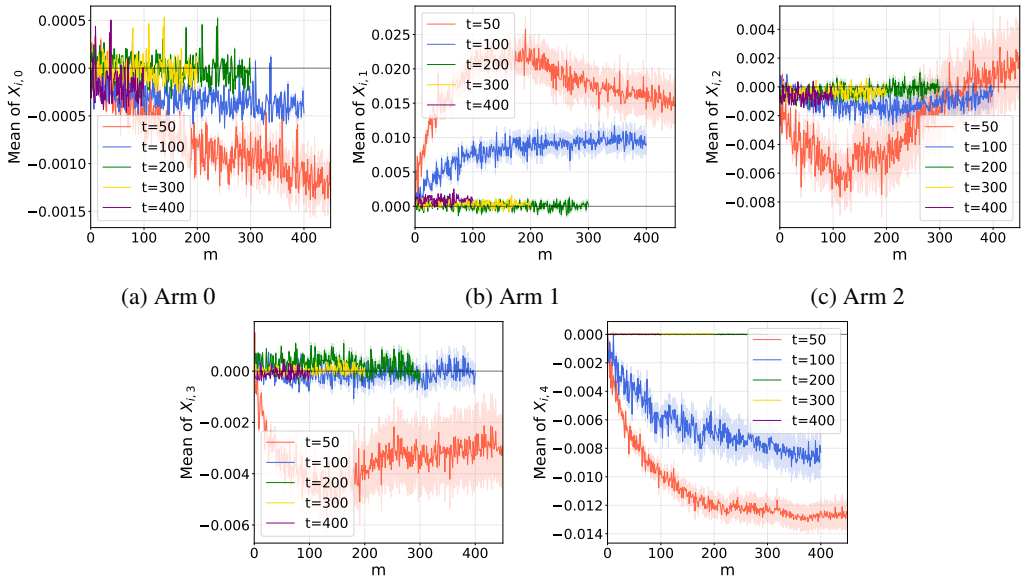
	Linear			
	Offline		Online	
n	Sort	Replace	Sort	Replace
25	1.37×10^{-107}	3.64×10^{-36}	6.02×10^{-91}	3.81×10^{-36}
50	5.32×10^{-91}	1.83×10^{-41}	1.22×10^{-81}	1.34×10^{-32}
100	1.99×10^{-49}	2.37×10^{-51}	2.73×10^{-42}	1.91×10^{-39}
150	2.60×10^{-31}	2.88×10^{-49}	3.15×10^{-29}	9.43×10^{-45}
200	2.73×10^{-30}	8.05×10^{-46}	2.60×10^{-32}	4.09×10^{-47}

D.3.2 GAUSSIAN EVALUATION FIGURES AND MARTINGALE PLOTS



(a) Cumulative Regret (b) TV Distance (c) Suboptimality (d) TV Distance

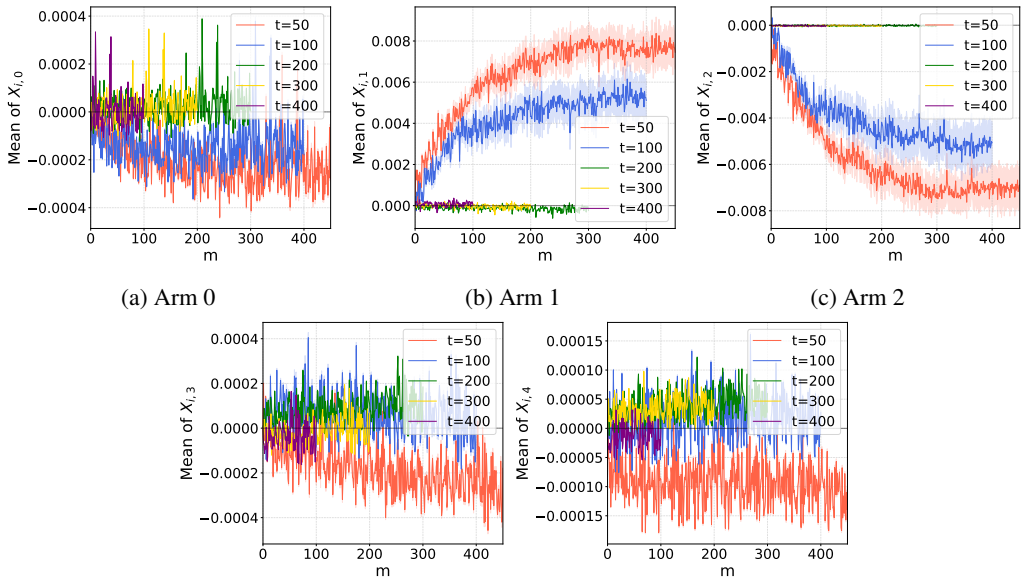
Figure 14: GDPT: Gaussian bandit



(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

Figure 15: GDPT: Gaussian bandit martingale increments, offline context.

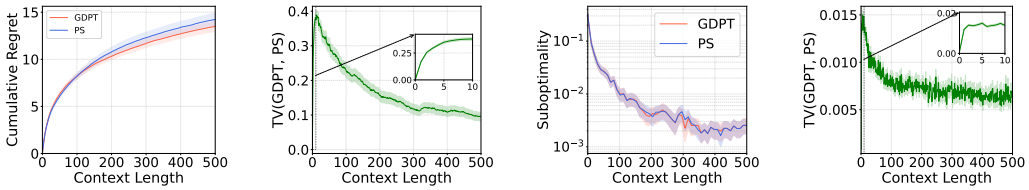


(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

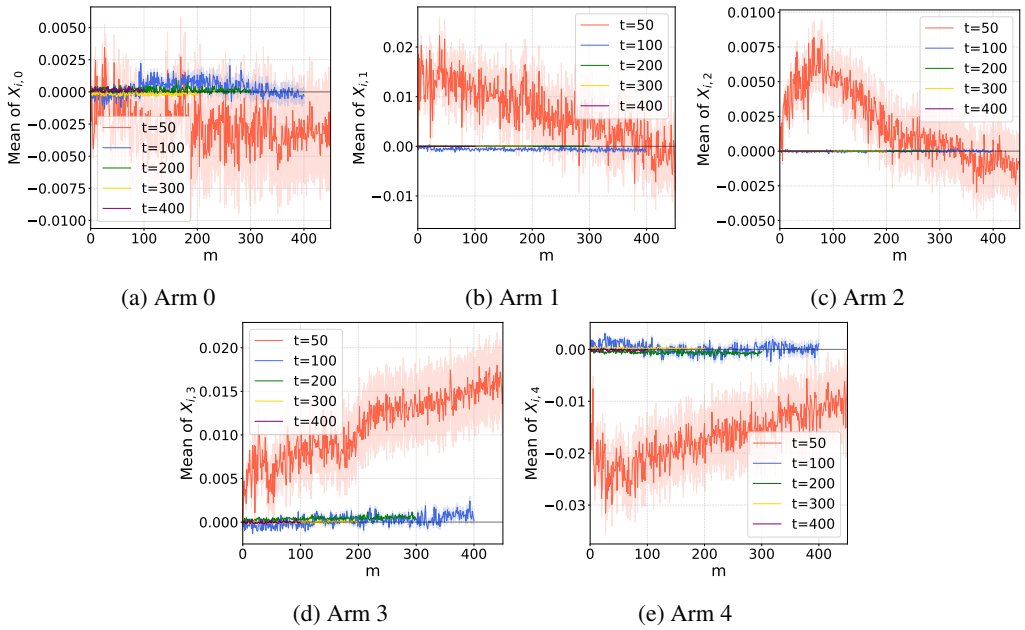
Figure 16: GDPT: Gaussian bandit martingale increments, online context.

D.3.3 BERNOULLI EVALUATION FIGURES AND MARTINGALE PLOTS



(a) Cumulative Regret (b) TV Distance (c) Suboptimality (d) TV Distance

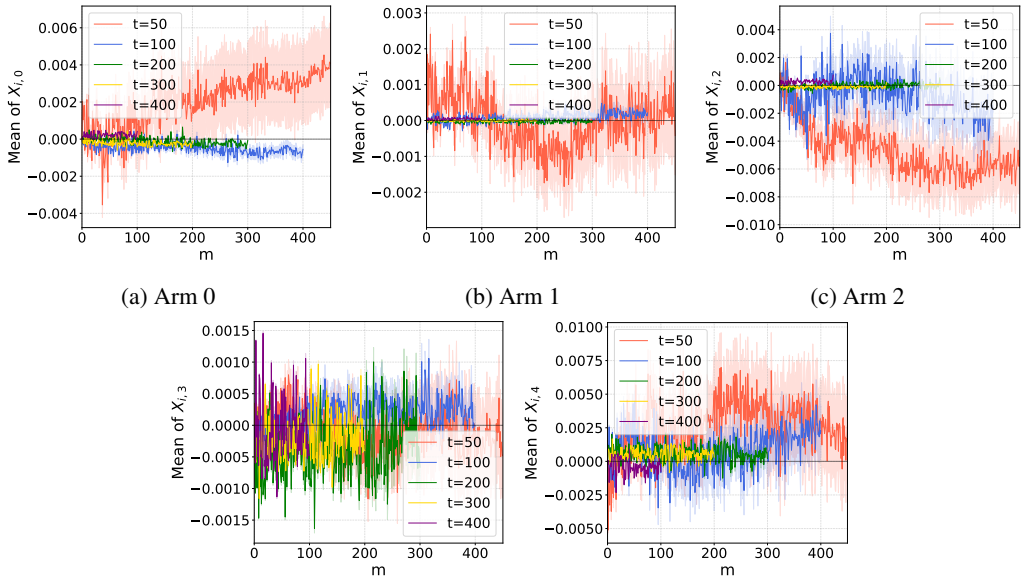
Figure 17: GDPT: Bernoulli bandit



(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

Figure 18: GDPT: Bernoulli bandit martingale increments, offline context.

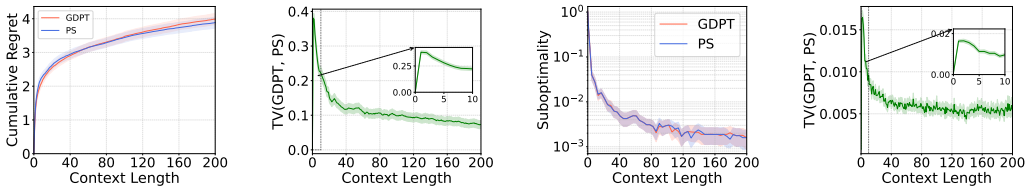


(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

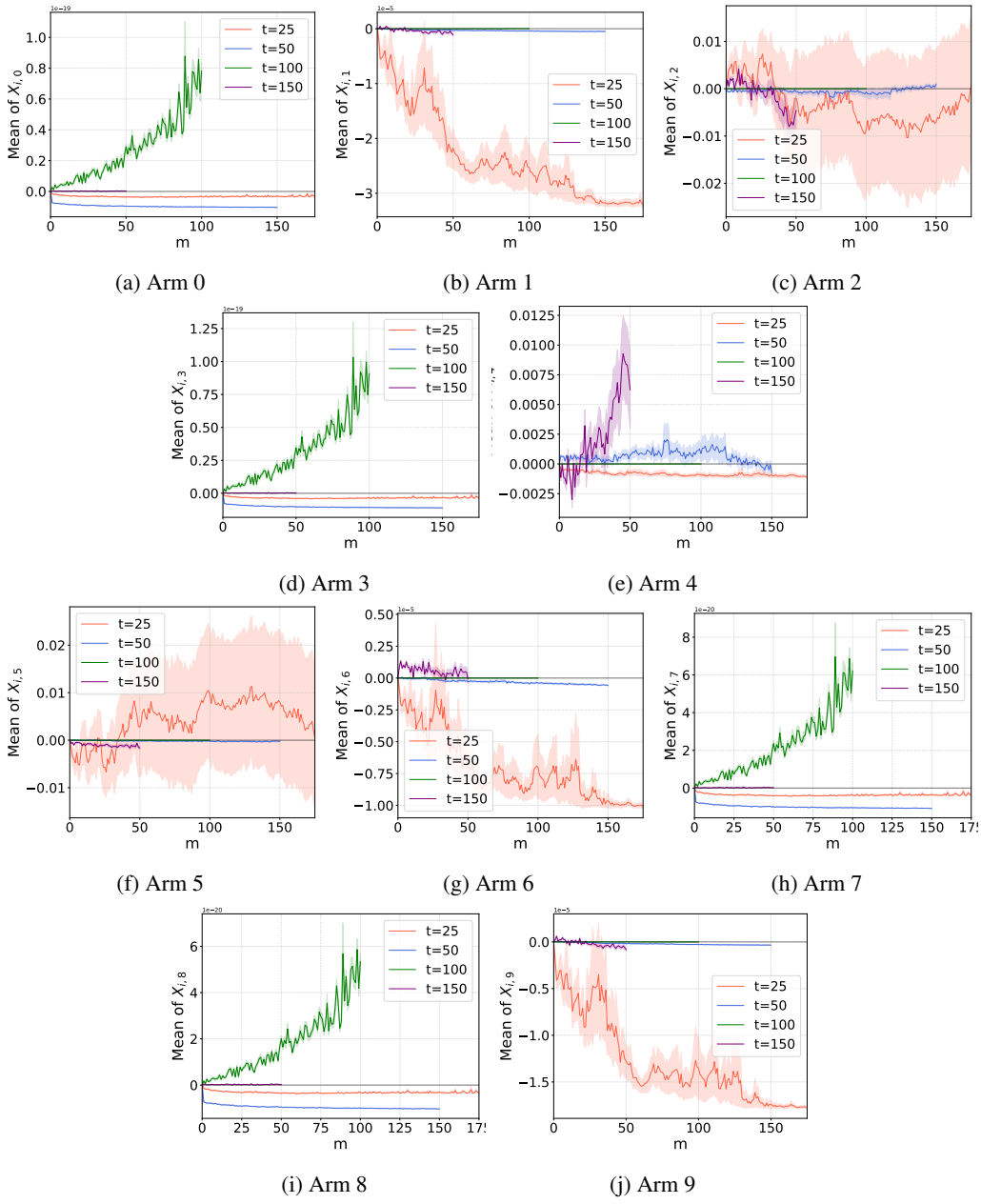
Figure 19: GDPT: Bernoulli bandit martingale increments, online context.

D.3.4 LINEAR EVALUATION FIGURES AND MARTINGALE PLOTS



(a) Cumulative Regret (b) TV Distance (c) Suboptimality (d) TV Distance

Figure 20: GDPT: linear bandit



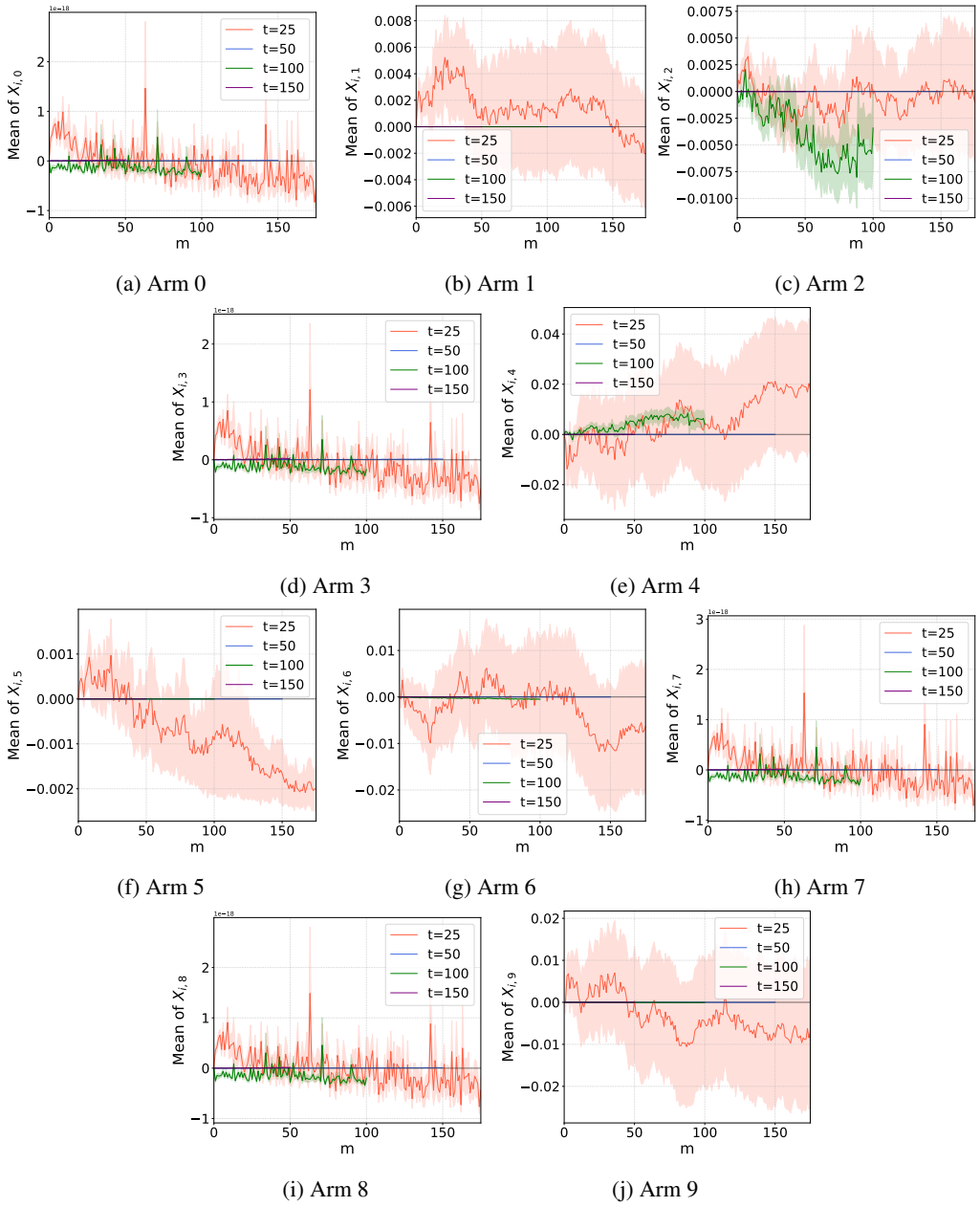
(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

(f) Arm 5 (g) Arm 6 (h) Arm 7

(i) Arm 8 (j) Arm 9

Figure 21: GDPT: Linear bandit martingale increments, offline context.



(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

(f) Arm 5 (g) Arm 6 (h) Arm 7

(i) Arm 8 (j) Arm 9

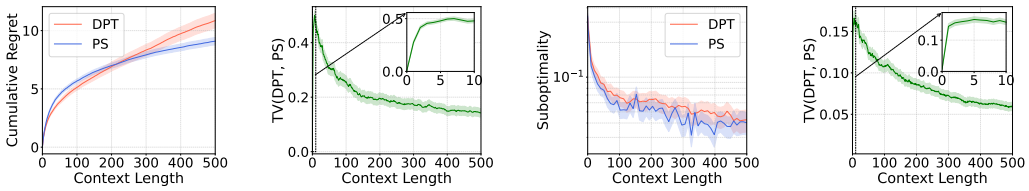
Figure 22: GDPT: Linear bandit martingale increments, online context.

D.4 Ablation studies

D.4.1 DPT sufficient-statistics table and evaluation figures

Table 7: DPT Gaussian architecture ablation: p-values for sufficient-statistic invariance tests ($K = 5$, $N = 5000$, $\alpha = 0.05$). "Equal" denotes exact equality, with no p-value reported.

	Gaussian			
	Offline		Online	
n	Sort	Replace	Sort	Replace
50	Equal	6.65×10^{-36}	Equal	6.60×10^{-90}
100	Equal	2.79×10^{-94}	Equal	2.54×10^{-62}
200	Equal	5.38×10^{-217}	Equal	1.46×10^{-46}
300	Equal	8.01×10^{-305}	Equal	2.11×10^{-44}
400	Equal	$< 10^{-300}$	Equal	6.13×10^{-56}
500	Equal	$< 10^{-300}$	Equal	2.55×10^{-67}



(a) Cumulative Regret (b) TV Distance (c) Suboptimality (d) TV Distance

Figure 23: DPT ablation: Gaussian bandit

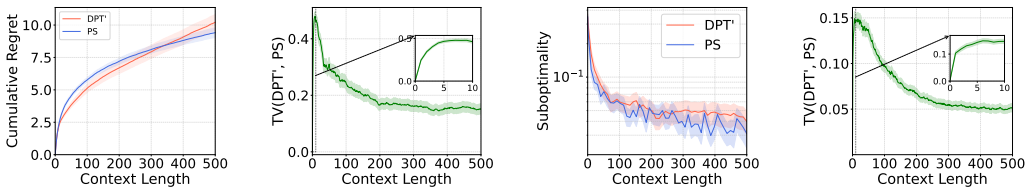
D.4.2 DPT′ sufficient-statistics and martingale tables and evaluation figures

Table 8: DPT′ Gaussian architecture ablation: p-values for sufficient-statistic invariance tests ($K = 5$, $N = 5000$, $\alpha = 0.05$). "Equal" denotes exact equality, with no p-value reported.

	Gaussian			
	Offline		Online	
n	Sort	Replace	Sort	Replace
50	Equal	2.46×10^{-233}	Equal	$< 10^{-300}$
100	Equal	1.09×10^{-293}	Equal	$< 10^{-300}$
200	Equal	$< 10^{-300}$	Equal	$< 10^{-300}$
300	Equal	$< 10^{-300}$	Equal	$< 10^{-300}$
400	Equal	$< 10^{-300}$	Equal	$< 10^{-300}$
500	Equal	$< 10^{-300}$	Equal	$< 10^{-300}$

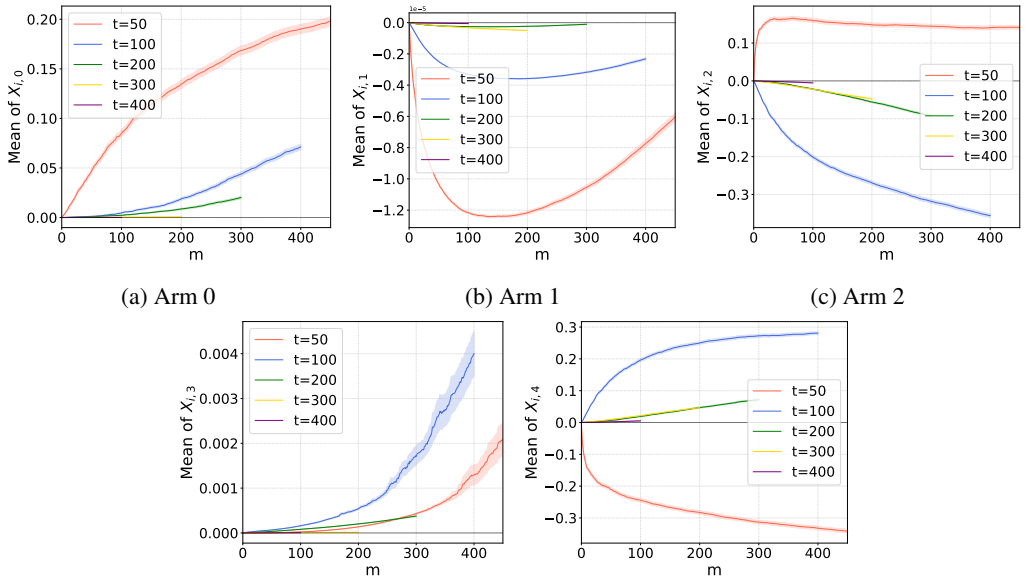
Table 9: DPT′ Gaussian architecture ablation: p-values for martingale tests at starting context length n and continuation length m ($K = 5$, $N = 5000$, $\alpha = 0.05$). Each test uses one fixed D_n and 5000 independent model-generated continuations.

		Gaussian	
n	m	Offline	Online
50	100	$< 10^{-300}$	$< 10^{-300}$
	200	$< 10^{-300}$	$< 10^{-300}$
	300	$< 10^{-300}$	$< 10^{-300}$
	400	$< 10^{-300}$	$< 10^{-300}$
	450	$< 10^{-300}$	$< 10^{-300}$
100	100	$< 10^{-300}$	1.22×10^{-37}
	200	$< 10^{-300}$	1.17×10^{-159}
	300	$< 10^{-300}$	$< 10^{-300}$
	400	$< 10^{-300}$	$< 10^{-300}$
200	100	$< 10^{-300}$	1.16×10^{-74}
	200	$< 10^{-300}$	1.53×10^{-256}
	300	$< 10^{-300}$	$< 10^{-300}$
300	100	$< 10^{-300}$	$< 10^{-300}$
	200	$< 10^{-300}$	1.92×10^{-253}
400	100	$< 10^{-300}$	1.41×10^{-236}



(a) Cumulative Regret (b) TV Distance (c) Suboptimality (d) TV Distance

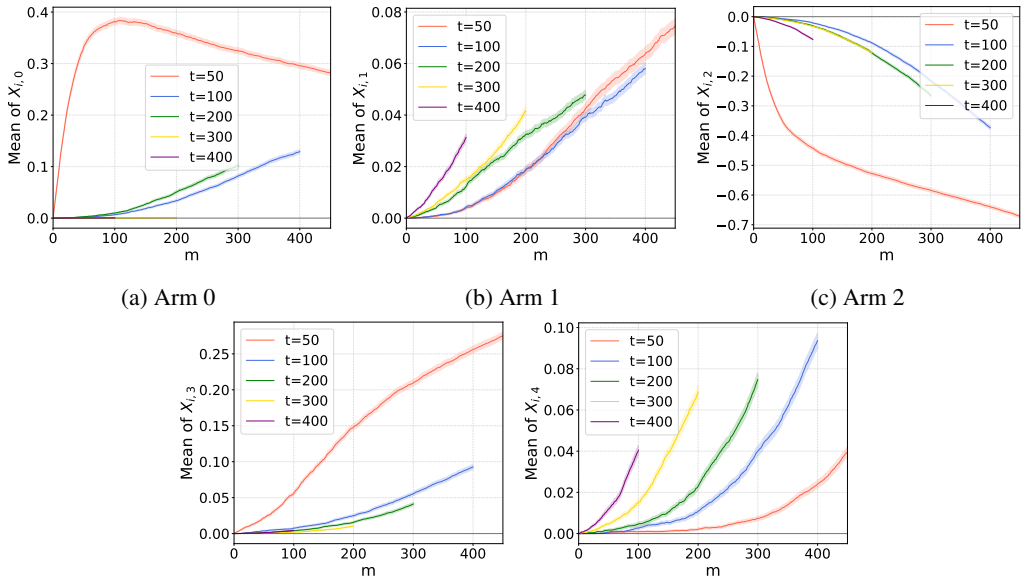
Figure 24: DPT′ ablation: Gaussian bandit



(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

Figure 25: DPT′ ablation: Gaussian bandit martingale increments, offline context.



(a) Arm 0 (b) Arm 1 (c) Arm 2

(d) Arm 3 (e) Arm 4

Figure 26: DPT′ ablation: Gaussian bandit martingale increments, online context.