PFN-TS: THOMPSON SAMPLING FOR CONTEXTUAL BANDITS VIA PRIOR-DATA FITTED NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Thompson sampling is a widely used strategy for contextual bandits: at each round, it samples a reward function from a Bayesian posterior and acts greedily under that sample. Prior-data fitted networks (PFNs), such as TabPFN v2+ and TabICL v2, are attractive candidates for this purpose because they approximate Bayesian posterior predictive distributions in a single forward pass. However, PFNs predict noisy future rewards, while Thompson sampling requires uncertainty over the latent mean reward function. We propose PFN-TS, a Thompson sampling algorithm that converts PFN posterior predictives into mean-reward samples using a subsampled predictive central limit theorem. The method estimates posterior variance from a geometric grid of $O(\log n)$ dataset prefixes rather than the full $O(n)$ predictive sequence used in previous predictive-sequence approaches, and reuses TabICL's cached representations across rounds. We prove consistency of the subsampled variance estimator and, for bounded rewards, bound Bayesian regret by an information-gain term and accumulated sampler error on encountered histories, with an extension to continuous rewards under additional assumptions. Empirically, PFN-TS achieves the best average rank across nonlinear synthetic and OpenML classification-to-bandit benchmarks, remains competitive on linear and BART-generated rewards, and attains the highest estimated policy value in an offline mobile-health evaluation. Code is available at `https://anonymous.4open.science/r/PFN_TS-3C48/`.

1 INTRODUCTION

Contextual bandits formalize sequential decision-making under uncertainty: at each round a learner observes a context, selects an action and receives a stochastic reward, with the goal of minimizing cumulative regret over T rounds. Thompson Sampling (TS) (Thompson, 1933; Russo & Van Roy, 2016) addresses this by maintaining a Bayesian posterior over reward functions and sampling actions from it, but its performance is only as good as the underlying reward model. In tabular applications such as mobile health (Klasnja et al., 2015; Bell et al., 2020) and recommendation systems (Li et al., 2010), linear reward models (Agrawal & Goyal, 2013) are standard yet often badly misspecified: human behavior involves interactions, threshold effects and heterogeneity that linearity cannot capture. Richer alternatives—kernel methods (Chowdhury & Gopalan, 2017), neural networks (Zhang et al., 2021) and tree ensembles (Nilsson et al., 2024; Deng et al., 2026)—each trade one limitation for another: kernel methods scale poorly in high dimensions, neural networks carry heavy computational overhead and tree-based methods impose piecewise-constant structure that fits poorly in smooth or near-linear regimes. Compounding these model-specific weaknesses, all these approaches require hyperparameter choices, such as kernel bandwidth, network architecture or tree depth, that are difficult to tune online.

Recently, tabular foundation models (TFMs) have emerged as state-of-the-art for regression and classification on small-to-medium tabular datasets. Among TFMs, TabPFN (Hollmann et al., 2025) and TabICLv2 (Qu et al., 2026) stand out because they are implemented as amortized Bayesian engines, also known as Prior-Data Fitted Networks (PFNs) (Müller et al., 2022): They are trained to approximate the Bayesian posterior predictive distributions (PPDs) arising from a broad prior over data-generating processes. When deployed on a new dataset, they approximate the PPD via a single forward pass, without any parameter updates, a paradigm known as in-context learning (ICL). This broad prior and ICL approach confer three properties that are unusually well-suited to

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.

bandit deployment. First, *model-fitting is very fast* since it requires no parameter updates. Second, *no hyperparameter tuning* is required and yet these models outperform AutoML baselines on benchmark datasets. Third, the models exhibit *structural adaptivity*: because the prior covers a wide range of DGPs, predictions automatically reflect whichever structure (e.g. sparsity, smoothness, nonlinearity or heteroscedasticity) is consistent with the observed data, avoiding the misspecification that afflicts single-model-class baselines (Zhang et al., 2025b). Unfortunately, while these models output PPDs that quantify uncertainty in a noisy future response, they do not directly output posterior distributions for the latent regression function, which is what is needed by TS.

A potential solution comes from Bayesian predictive inference, which shows how to recover posterior parameter distributions from PPDs (Fong et al., 2023; Fortini & Petrone, 2025). One can either (i) sample from the posterior by recursively simulating future observations from a model's PPDs, or (ii) estimate the posterior variance by tracking how much the PPDs fluctuate as we condition on prefixes of the observed data, thereby constructing a normal approximation. Ng et al. (2026) and Fortini et al. (2026) apply these respective techniques to calculate estimates of regression function posteriors for TabPFN in an offline setting. However, the cost of computing the full sequence of PPDs required to construct either method's posterior estimates means that neither is practical for deployment in an online bandit setting, where a fresh posterior sample is needed for each arm at each round.

In this work, we break this computational bottleneck. Building on Fortini et al. (2026), we construct a normal approximation to the posterior, but using a more efficient consistent variance estimator. More precisely, given an observed dataset of size n, we show that the asymptotic variance of the posterior can be consistently estimated using model snapshots along $O(\log n)$ geometrically spaced prefixes rather than all n prefixes. We embed this posterior approximation for PFNs into a TS framework, which we call *PFN-TS*. PFN-TS further reduces per-round cost by reusing cached representations of the fitted PFN across rounds. We derive a Bayesian regret bound in terms of information acquired along PFN-TS's own trajectory and accumulated posterior-sampling error on the histories it encounters. The bound holds for bounded rewards, with an extension under explicit conditions for continuous rewards. For disjoint sampling, we further decompose the action error into Gaussian snapshot approximation error and PFN predictive-path error, leaving the prior-specific information and approximation rates explicit.

While our methodology and theory apply to any PFN, we instantiate our method with TabICLv2 (Qu et al., 2026) (hereafter TabICL), a PFN for tabular regression and classification that extends TabPFN (Hollmann et al., 2023; 2025) with architectural improvements and, critically, caching of the key-value attention matrices for a fixed training set. In the bandit setting, where the same history is queried once per arm per round, this reduces the per-round cost substantially. We evaluate it against a comprehensive set of baselines on synthetic benchmarks, eight OpenML classification datasets and the Drink Less mobile health trial, following the same experimental protocol as Deng et al. (2026). We find that PFN-TS substantially outperforms all baselines on almost all nonlinear tasks, while matching the performance of LinTS on linear tasks, and achieves the highest estimated policy value on the Drink Less trial.

2 Posterior Uncertainty from Predictive Sequences

2.1 PFNs as amortized Bayesian inference

Prior-Data Fitted Networks (PFNs) (Müller et al., 2022) are transformer models trained to approximate Bayesian PPDs in a single forward pass. Specifically, we define the hierarchical Bayesian model:

$$\mathcal{D} \mid \pi, N \overset{\text{i.i.d.}}{\sim} \pi, \quad y \mid x, \pi \sim \pi(\cdot \mid x), \quad \pi \sim \Pi, \quad N \sim \rho_N, \tag{1}$$

where $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ is a dataset of random size N, π is a joint distribution over the pair (x, y), Π is a prior over such distributions, and ρ_N is a distribution over dataset sizes. During training, synthetic datasets are drawn from this model, and the network minimizes negative log-likelihood given a training set and a query input. In other words, the network learns to approximate the mapping $(x, \mathcal{D}) \mapsto p_\Pi(dy \mid x, \mathcal{D})$, where $p_\Pi(dy \mid x, \mathcal{D}) = \int \pi(dy \mid x)\, \Pi(d\pi \mid \mathcal{D})$ is the PPD.[1]

[1]The observations in $\mathcal{D}$, including their covariates, update $\Pi(d\pi \mid \mathcal{D})$. The query x is treated as an externally supplied prediction point and does not itself update this posterior.

2.2 PREDICTIVE SEQUENCES AS POSTERIOR SAMPLERS

For a query point x and a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, define the sequence of predictive means $m_i(x) = m(x; \mathcal{D}_{1:i})$, $i = 0, \ldots, n$, where $m(x; \mathcal{D})$ is the mean of the PFN's output distribution at query x given context $\mathcal{D}$, and $\mathcal{D}_{1:i}$ is the first i observations in a fixed ordering of $\mathcal{D}$.

Exact PPD predictive sequences. Let $f_0(x) = \int y\, \pi(dy \mid x)$ be the latent mean reward under the realized task $\pi \sim \Pi$. In the idealized setting when the PFN is an exact Bayesian PPD for prior Π, we have

$$m_i(x) = \mathbb{E}_\Pi[f_0(x) \mid \mathcal{F}_i], \qquad \mathcal{F}_i = \sigma(\mathcal{D}_{1:i}),$$

where $f_0(x)$ is assumed integrable under the prior-predictive law. By the tower property, $\mathbb{E}[m_{i+1}(x) \mid \mathcal{F}_i] = m_i(x)$, so the sequence forms a uniformly integrable Doob martingale and Lévy's upward theorem gives $m_i(x) \longrightarrow \mathbb{E}_\Pi[f_0(x) \mid \mathcal{F}_\infty]$ a.s. and in L^1, where $\mathcal{F}_\infty = \sigma(\bigcup_i \mathcal{F}_i)$. To identify this limit, assume that $f_0(x)$ is identifiable from the infinite data sequence, or equivalently is $\mathcal{F}_\infty$-measurable. This condition holds for π_X-almost every x in standard identifiable regression models; a pointwise claim outside the support of π_X requires additional regularity. Under this condition, $m_\infty(x) = f_0(x)$ a.s.

Consequently, the conditional law of the limit is exactly the posterior over $f_0(x)$:

$$\mathcal{L}\big(m_\infty(x) \mid \mathcal{F}_n\big) = \mathcal{L}\big(f_0(x) \mid \mathcal{F}_n\big). \tag{2}$$

The randomness of $m_\infty(x)$ given $\mathcal{F}_n$—equivalently, the randomness in the unobserved future data that reveal the limit—is exactly the Bayesian uncertainty about $f_0(x)$ (Fortini & Petrone, 2025). Thus posterior uncertainty can be represented by the conditional law of the martingale tail $m_\infty(x) - m_n(x)$. Martingality gives the centering and orthogonality structure, but a Gaussian approximation requires second-order regularity of the predictive increments. Under predictive-CLT conditions, this tail satisfies

$$\mathcal{L}\big(\sqrt{n}\,(m_\infty(x) - m_n(x)) \,\big|\, \mathcal{F}_n\big) \xrightarrow{w} \mathcal{N}(0, V(x)) \quad \mathbb{P}\text{-a.s.}, \tag{3}$$

as $n \to \infty$. This yields a tractable Gaussian approximation to the posterior over the latent mean at an identifiable query.

Approximate PPD predictive sequences. Since PFNs only provide an approximation to the exact PPD, the above properties do not hold automatically. Fong et al. (2023) proved that under martingale conditions on the predictive sequence, the limit $m_\infty(x)$ still exists and is identifiable as a random variable, although not necessarily equal to $f_0(x)$. Calling it a martingale posterior, they argue that it still offers a form of uncertainty quantification, which can be computed by simulating future values in the predictive sequence.

Unfortunately, PFNs do not satisfy the exact martingale condition either (Ng et al., 2026). To address this issue, Fortini et al. (2026) introduces more relaxed sufficient conditions on the predictive sequences that still guarantee the existence of a limiting random variable (still called a martingale posterior)and provides empirical evidence that PFN's predictive sequences exhibit such limiting behavior, thereby enabling uncertainty quantification in an offline setting. While they worked with predictive CDFs rather than means, a similar analysis applies to the predictive mean sequence as well. We now state sufficient conditions for the generic predictive-mean sequence $m_i(x)$ produced by an approximate PPD model. Its limit need not equal $f_0(x)$; approximation to the exact posterior is handled separately in Section 4.

Assumption 1 (Asymptotic variance)**.** *There is a strictly positive, $\mathcal{F}_\infty$-measurable limiting variance $V(x)$ such that*

$$i^2 \mathbb{E}\big[(m_i(x) - m_{i-1}(x))^2 \mid \mathcal{F}_{i-1}\big] \longrightarrow V(x) \quad \textit{a.s.}$$

Assumption 2 (Quasi-martingale condition)**.** $\displaystyle\sum_{i \geq 1} \sqrt{i} \left(\mathbb{E}\Big[\big(\mathbb{E}[m_i(x) - m_{i-1}(x) \mid \mathcal{F}_{i-1}]\big)^2\Big]\right)^{1/2} < \infty.$

Assumption 3 (Bounded moments)**.** *The initial value satisfies $m_0(x) \in L^2$, and there exist $C > 0$ and $\epsilon > 0$ such that*

$$\mathbb{E}\big[|m_i(x) - m_{i-1}(x)|^{4+\epsilon}\big] \leq \frac{C}{i^{4+\epsilon}}.$$

Following Fortini et al. (2026, Theorem 4.3) and under these assumptions, one can establish a predictive CLT similar to (3) whose asymptotic variance $V(x)$ can be estimated by $\frac{1}{n}\sum_{i=1}^n i^2\,(m_i(x) -$

$m_{i-1}(x))^2$, but evaluating this requires a forward pass at all n dataset prefixes, which is prohibitive in an online bandit setting where a fresh variance estimate is needed for each arm at each round.

2.3 Subsampled CLT for predictive means

Rather than evaluating all n increments $m_i(x) - m_{i-1}(x)$, we compute block increments at $O(\log n)$ geometrically spaced prefix indices $2 = t_0 < t_1 < \cdots < t_J \le n$, where t_J is the last point in the geometric sequence not exceeding n. For interval $(t_{j-1}, t_j]$, set $D_j = m_{t_j}(x) - m_{t_{j-1}}(x)$ and $w_j = t_j t_{j-1}/(t_j - t_{j-1})$.

To see why this weighting is natural, write each increment as a martingale-difference term plus a predictable drift. When the drift is negligible, martingale orthogonality suggests $\mathbb{E}[D_j^2] \approx V(x) \sum_{i=t_{j-1}+1}^{t_j} i^{-2} \approx V(x)(1/t_{j-1} - 1/t_j)$, so the harmonic weights w_j achieve the same identification target as the i^2 weights in the estimator of Fortini et al. (2026) on a coarser grid.

The resulting variance estimator is

$$\hat{V}_{s_n}(x) = \frac{1}{J} \sum_{j=1}^{J} w_j D_j^2, \tag{4}$$

where $s_n = t_J$ is the latest refresh point. The cached sampler uses the snapshot Gaussian approximation $\mathcal{N}(m_{s_n}(x), \hat{V}_{s_n}(x)/s_n)$. With $J = O(\log_b n)$ terms, the default $b = 2$ requires $O(\log_2 n)$ forward passes per arm per round. Pseudocode is in Algorithm 1 (see also Figure 1); the formal guarantee is as follows.

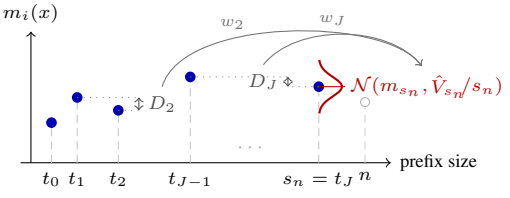

Figure 1: SubCLT evaluates the predictive mean on a geometric prefix grid. The trajectory increments D_j estimate the CLT variance used for the Thompson sample at the latest refresh point s_n.

Theorem 1 (Subsampled predictive CLT)**.** *Under Assumptions 1 to 3, there is an integrable random limit $m_\infty(x)$ such that*

$$\mathcal{L}\big(\sqrt{q}\,(m_\infty(x) - m_q(x)) \mid \mathcal{F}_q\big) \xrightarrow{w} \mathcal{N}(0, V(x)) \quad \mathbb{P}\text{-a.s.}$$

as $q \to \infty$. Therefore the same CLT holds along any diverging sequence of refresh points. Moreover, for a history of size n and the geometric grid obtained by iterating $t_{j+1} = \max(t_j+1, \lfloor b\, t_j \rfloor)$ up to the last point $t_J \le n$, with fixed $b > 1$, the estimator $\hat{V}_{s_n}(x)$ defined in Equation (4) *satisfies $\hat{V}_{s_n}(x) \xrightarrow{p} V(x)$, where $s_n = t_J$.*

For an exact identifiable PPD, the limit is $f_0(x)$; for an approximate PFN, Section 4 separately accounts for the discrepancy from exact posterior sampling.

3 Thompson Sampling with PFNs

3.1 Problem setup

We consider the standard contextual bandit over T rounds. At each round $t \in [T]$, the learner observes a context $X_t \in \mathcal{X} \subseteq [0,1]^p$, selects an action $A_t \in \mathcal{A}$ with $|\mathcal{A}| = K$, and receives a reward $R_t = f_0(X_t, A_t) + \varepsilon_t$, where $f_0 : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is the unknown mean reward function and $\mathbb{E}[\varepsilon_t \mid X_t, A_t, \mathcal{H}_{t-1}] = 0$. The history at round t is $\mathcal{H}_t = \{(X_s, A_s, R_s)\}_{s=1}^t$. The goal is to minimize cumulative regret against the oracle policy $A^*(x) \in \arg\max_{a \in \mathcal{A}} f_0(x, a)$, writing $A_t^* = A^*(X_t)$:

$$\mathrm{Regret}_T = \sum_{t=1}^{T} \big(f_0(X_t, A^*(X_t)) - f_0(X_t, A_t)\big).$$

Thompson Sampling maintains a posterior over reward functions. At each round, after observing X_t, it draws a function $\hat{f}_t$ from this posterior and plays $A_t \in \arg\max_a \hat{f}_t(X_t, a)$. PFN-TS approximates this posterior-sampling step using the predictive mean sequences from Section 2.

3.2 CONTEXT PARAMETERIZATION

A design choice in applying any tabular model to contextual bandits is how to represent arm identity in the input. We consider two strategies.

Disjoint encoding maintains one history per arm, $\mathcal{D}^{(a)} = \{(X_s, R_s) : s < t,\ A_s = a\}$. Each PFN snapshot for arm a is conditioned only on $\mathcal{D}^{(a)}$, so the construction corresponds to independent arm-specific models.

Joint integer encoding maintains one shared history. Let $g(a) \in \{0, \dots, K-1\}$ be the integer arm code and define $\widetilde{x}_a = (x, g(a)) \in \mathbb{R}^{p+1}$. TabICLv2 treats this integer coordinate as categorical. The shared history

$$\mathcal{D} = \{((X_s, g(A_s)), R_s) : s < t\}$$

allows observations from one arm to inform predictions for another without the K additional coordinates of a one-hot representation.

Neither encoding universally dominates. Joint encoding can be preferable when arms share structure, since it pools observations; disjoint encoding can be more effective when arm reward functions are unrelated. A third option, *block encoding*, expands the feature space to $\mathbb{R}^{pK}$ by placing x in the arm's block and zero-padding elsewhere, but the K-fold increase in dimensionality makes it less practical for PFNs.

3.3 THE PFN-TS ALGORITHM

After a randomized round-robin warm-up of τ observations per arm, PFN-TS enters its main Thompson Sampling loop. We describe how it operates under each encoding, then explain the adaptive selection rule. Predictive means are evaluated on the geometric snapshot grid from Section 2.3; each snapshot stores the key–value cache for its history prefix and is reused in subsequent rounds. All reported TabICLv2 variants use one estimator.

Disjoint encoding. For arm a, let $n_a = |\mathcal{D}^{(a)}|$, let $s_{t,a}$ be its latest grid point, and let $\hat{V}_{s_{t,a}}(X_t)$ be the SubCLT estimate. The raw posterior variance estimate is $\hat{V}_{s_{t,a}}(X_t)/s_{t,a}$. To prevent a finite-sample zero estimate from making an arm permanently deterministic, we optionally use the decaying floor

$$\delta_{t,a} = \sigma_0^2 \frac{n_0}{n_a + n_0}, \qquad \hat{\sigma}^2_{t,a} = \max\left\{ \frac{\hat{V}_{s_{t,a}}(X_t)}{s_{t,a}}, \delta_{t,a} \right\}.$$

The floor is enabled for the binary-reward experiments and disabled for the continuous synthetic experiments. PFN-TS independently samples

$$\widetilde{r}_{t,a} \sim \mathcal{N}\Big(m(X_t; \mathcal{D}^{(a)}_{1:s_{t,a}}), \hat{\sigma}^2_{t,a}\Big)$$

and selects the arm with the largest draw, $A_t \in \arg\max_a \widetilde{r}_{t,a}$.

Joint encoding. Under joint encoding, all arms share one model and a grid $t_0 < \cdots < t_J = s_t$ based on total history size. At snapshot j, a single batch of arm queries gives the vector $\boldsymbol{m}_j(X_t)$ with components $m(\widetilde{X}_{t,a}; \mathcal{D}_{1:t_j})$, $a = 1, \dots, K$. Writing $\boldsymbol{d}_j = \boldsymbol{m}_j - \boldsymbol{m}_{j-1}$ and $w_j = t_j t_{j-1}/(t_j - t_{j-1})$, the vector analogue of SubCLT estimates posterior covariance by

$$\widehat{\Sigma}_t = \frac{1}{s_t J} \sum_{j=1}^{J} w_j \boldsymbol{d}_j \boldsymbol{d}_j^\top.$$

This estimate captures dependence between arm values but has rank at most J. We therefore shrink its off-diagonal entries toward zero, with stronger shrinkage when there are few grid increments relative to the number of arms, and raise any marginal variance below the floor used for disjoint encoding. These steps preserve positive semidefiniteness. Let $\widehat{\Sigma}^{\mathrm{reg}}_t$ denote the resulting matrix. PFN-TS draws $\widetilde{\boldsymbol{r}}_t \sim \mathcal{N}(\boldsymbol{m}_J(X_t), \widehat{\Sigma}^{\mathrm{reg}}_t)$ and selects its largest component. The precise regularization rule is given in Algorithm 2. The joint covariance construction is an empirical extension of the scalar guarantee in Theorem 1.

Adaptive encoding selection. Since neither encoding dominates in general and their relative performance is unknown in advance, PFN-TS maintains both in parallel during a *dual-encoding* phase. It selects between them using the Continuous Ranked Probability Score (CRPS) (Gneiting & Raftery, 2007): for a predictive CDF $\widehat{F}$ and observed reward r,

$$\mathrm{CRPS}(\widehat{F}, r) = \int_{-\infty}^{\infty} \big(\widehat{F}(y) - \mathbf{1}\{y \geq r\}\big)^2 \, dy. \tag{5}$$

CRPS is a strictly proper scoring rule: its expected value is minimized by the true predictive distribution. At prespecified switch times, PFN-TS compares the encodings' cumulative scores on observed rewards and uses the lower-scoring encoding for Thompson sampling. After the final switch, the challenger is discarded. Scoring reuses cached snapshots but still requires predictive evaluations at the switch checkpoints. Full pseudocode is in Algorithm 3.

Forward-pass complexity. At round t, write $J_{t,a} + 1$ for the number of cached snapshots of disjoint arm a. Computing all disjoint Thompson draws requires $\sum_a (J_{t,a} + 1) = O(\sum_a \log n_a)$ cached prediction calls. Joint encoding requires $J_t + 1 = O(\log t)$ calls, each with a batch of K arm queries. A new conditioned snapshot is constructed only when a geometric grid point is reached, so the cumulative number of snapshot constructions through horizon T is $O(\sum_a \log n_a(T))$ for disjoint encoding and $O(\log T)$ for joint encoding. During the adaptive dual phase, maintaining challenger snapshots and evaluating CRPS at switch checkpoints add to the active representation's decision cost. After the final switch, only the selected representation is maintained and evaluated.

Related work. Zhang et al. (2025a) propose a closely related framework in which TS uncertainty arises from missing counterfactual outcomes: a generative sequence model, pretrained on bandit data from previous tasks, imputes unobserved arm outcomes, and a policy is then fitted to the completed dataset. Our approach differs in three ways: (i) we use a generic PFN without task-specific training; (ii) we extract latent-mean uncertainty via SubCLT rather than generating a complete set of counterfactual outcomes; and (iii) we operate in a single task without requiring offline data from previous tasks in the same distribution.

4 Bayesian Regret of PFN-TS

The Bayesian regret of TS algorithms is often analyzed via the information ratio argument of Russo & Van Roy (2016). Since PFN-TS is not an exact posterior sampler, we cannot directly apply this argument. However, we can extend it to this setting by considering the per-step overlap between the action probabilities of PFN-TS and exact TS.

More precisely, let $\Theta \sim \Pi$ denote the latent reward environment, including any noise parameters, with mean reward function $f_0(x, a)$. Write Q for the PFN-TS policy and $\mathcal{H}_t^Q$ for its history through round t. At round t, let $p_{t,a} = \Pr_\Pi(A_t^* = a \mid \mathcal{H}_{t-1}^Q, X_t)$ be the exact-TS action probability and $q_{t,a} = Q_t(a \mid \mathcal{H}_{t-1}^Q, X_t)$ the actual action probability. Define the discrepancy on the encountered history and its accumulated expectation by

$$\eta_t = \frac{1}{2} \sum_{a=1}^{K} |p_{t,a} - q_{t,a}|, \qquad \mathcal{E}_T = \sum_{t=1}^{T} \mathbb{E}_Q[\eta_t].$$

Theorem 2 (Bayesian regret with encountered sampler error)**.** *Suppose rewards lie in* $[0, 1]$*. Then, with* $I_T^Q = I_\Pi(\Theta; \mathcal{H}_T^Q)$*,*

$$\mathbb{E}_Q[\mathrm{Regret}_T] \leq \sqrt{\frac{KT}{2} I_T^Q} + \mathcal{E}_T.$$

Remark 1 (Continuous rewards)**.** *If latent means lie in a common interval of width* M *across arms and posterior-supported environments, and centered posterior-predictive rewards are uniformly* s*-sub-Gaussian, the bound becomes* $\mathbb{E}_Q[\mathrm{Regret}_T] \leq \sqrt{2s^2 KT I_T^Q} + M\mathcal{E}_T$*. For conditionally* σ*-sub-Gaussian observation noise and bounded latent means,* $s^2 = \sigma^2 + M^2/4$ *suffices. These are additional assumptions, not consequences of a bound on differences between arms; see Corollary 3 for the proof.*

Remark 2 (Information complexity and comparison with existing methods)**.** *The information term satisfies $I_T^Q \le \gamma_T^{\mathrm{ad}}(\Pi)$, where $\gamma_T^{\mathrm{ad}}(\Pi)$ is the maximum information acquired by an admissible policy over T rounds (Appendix A.3). Thus the bound has the familiar information-complexity form $\sqrt{KT\gamma_T^{\mathrm{ad}}(\Pi)/2}$, plus accumulated sampler error. As reference points, standard linear and squared-exponential kernel models exhibit logarithmic or polylogarithmic information growth under their respective assumptions (Chowdhury & Gopalan, 2017; Vakili et al., 2021). If the PFN prior admits comparable information growth and $\mathcal{E}_T = \widetilde{O}(\sqrt{T})$, our bound yields $\widetilde{O}(\sqrt{T})$ regret for fixed model dimensions and number of arms, subject to the reward assumptions of Theorem 2 and Corollary 3. These comparisons motivate—but do not establish—analogous behavior for PFN priors.*

Remark 3 (The sampler discrepancy)**.** *Proposition 4 bounds η_t at each encountered history by a Gaussian snapshot approximation error and the discrepancy between exact-PPD and PFN predictive-mean paths. Averaging these bounds controls $\mathcal{E}_T$.*

5 Numerical Experiments

Across synthetic and real-data benchmarks, PFN-TS improves over standard baselines when reward functions are nonlinear or heterogeneous, while remaining competitive on problems where simpler inductive biases are well matched to the data. Figure 2 shows selected regret curves; full trajectories are in Appendices D.1 and D.2.

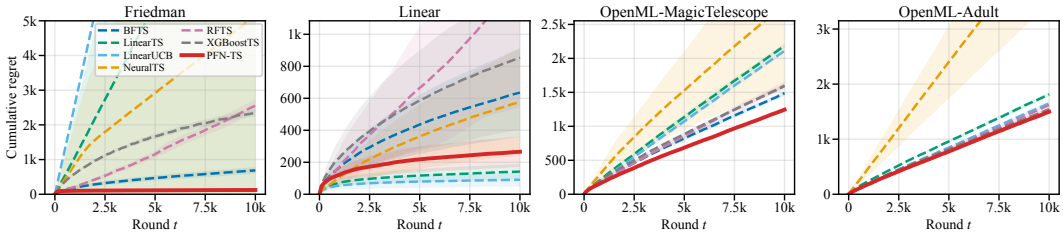

Figure 2: Cumulative regret trajectories on selected synthetic and OpenML datasets (mean ± population SD). Full results are in Appendices D.1 and D.2.

Experimental setup. We compare PFN-TS with **LinTS** (linear Thompson Sampling), **LinUCB** (Li et al., 2010), **NeuralTS** (Zhang et al., 2021), **BFTS** (Deng et al., 2026), **RFTS**, and **XGBoostTS** (Nilsson et al., 2024). We use a horizon of $T = 10{,}000$ or the dataset size, whichever is smaller, and repeat each experiment $R = 12$ times. Implementation details are in Appendix C.2.

5.1 Synthetic Benchmarks

We evaluate on three families of data-generating processes (DGPs): (i) **Linear** rewards with Gaussian noise, (ii) **Friedman** nonlinear rewards with varying sparsity, correlation structure, and heteroscedastic noise, and (iii) **SynBART**, where each arm's reward function is sampled from a BART prior, providing a correctly-specified benchmark for BFTS. Full DGP definitions are in Appendix C.3.

Table 1 reports final cumulative regret across all scenarios. PFN-TS performs consistently across all variants of Friedman datasets. This suggests that TabICL adapts to these structures without scenario-specific tuning. Its weaker performance on the Linear benchmark is expected and reflects the advantage of a correctly specified linear model. Conversely, PFN-TS remains competitive on SynBART despite that DGP being generated from the prior used by BFTS.

SubCLT uncertainty diagnostics. We assess uncertainty quality using offline regression diagnostics. Figure 3 shows that SubCLT (with TabICL) slightly undercovers, but coverage improves with sample size, as expected from a large-sample approximation. Relative to FullCLT, SubCLT gives a reasonable approximation while being substantially faster. Its credible sets also tend to be narrower than BART. See Appendix E for implementation details.

Table 1: Final cumulative regret on synthetic benchmarks (mean ± sample SE). Average rank is computed across all eight scenarios. **Bold**: lowest mean per scenario.

Method	Friedman	Friedman-Hetero.	Friedman-Sparse	Friedman-Sparse-Dis.	Friedman2	Friedman3	Linear	SynBART	Rank
PFN-TS	**122.9 ± 17.1**	**132.5 ± 18.7**	**200.4 ± 15.6**	**587.0 ± 19.4**	**65.8 ± 17.1**	**108.1 ± 10.2**	265.8 ± 28.5	**43.3 ± 6.4**	1.250
BFTS	687.8 ± 28.7	715.5 ± 40.9	814.2 ± 26.1	660.1 ± 15.6	377.4 ± 25.3	592.3 ± 19.6	635.7 ± 83.5	44.5 ± 6.1	2.500
LinTS	10951.5 ± 3907.0	21888.2 ± 2951.6	6345.8 ± 3305.8	6560.4 ± 556.7	557.2 ± 53.4	17463.0 ± 3723.2	141.7 ± 14.1	277.7 ± 46.1	5.000
LinUCB	26305.1 ± 60.6	26265.6 ± 41.3	21133.0 ± 2278.3	10219.1 ± 1647.1	777.7 ± 48.8	24048.4 ± 2186.5	**90.6 ± 12.6**	268.2 ± 63.2	5.875
NeuralTS	5149.9 ± 2911.9	14328.1 ± 3659.0	8708.3 ± 3039.7	8785.6 ± 2278.9	452.6 ± 295.5	219.4 ± 53.6	577.5 ± 100.9	158.3 ± 47.4	4.500
RFTS	2558.2 ± 51.1	2677.6 ± 66.1	2561.7 ± 54.5	7683.3 ± 59.3	1071.6 ± 24.9	947.4 ± 62.5	1301.7 ± 227.6	92.7 ± 14.5	4.750
XGBoostTS	2352.2 ± 30.9	2422.2 ± 25.0	2622.3 ± 19.9	5082.6 ± 24.7	799.1 ± 13.0	1736.4 ± 15.2	853.5 ± 136.1	58.2 ± 8.0	4.125

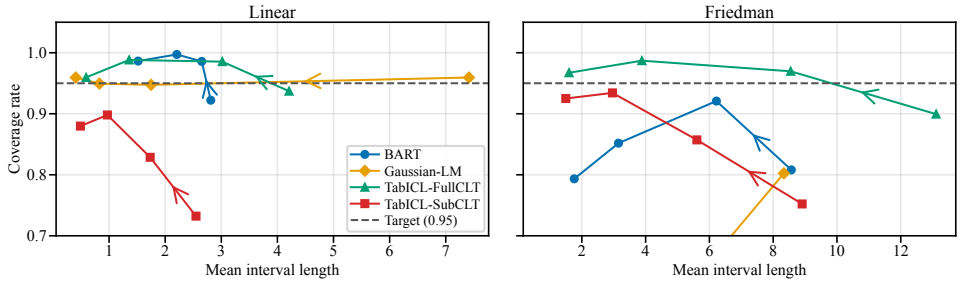

Figure 3: Coverage–length diagnostics for nominal 95% latent-mean intervals. Each marker is one sample size, and arrows indicate increasing sample size ($n = 16, 64, 256, 1024$). The Friedman Gaussian-LM curve extends below the plot. Lower interval length at fixed coverage is better.

5.2 OpenML Benchmarks

We transform eight OpenML classification tasks into contextual bandit problems following standard protocol (Riquelme et al., 2018; Zhang et al., 2021; Nilsson et al., 2024; Deng et al., 2026): each class label becomes an arm, and the reward is 1 for choosing the correct class and 0 otherwise.

Table 2 reports final cumulative regret and average rank across all eight datasets. PFN-TS's gains are largest where the linear baselines perform poorly. The exceptions are PageBlocks and Shuttle, where tree-based methods are strongest. Shuttle combines seven arms with severe class imbalance, limiting feedback for rare actions and making uncertainty estimation and encoding selection more difficult. This likely contributes to PFN-TS's instability on that dataset.

Runtime. Figures 14 and 15 report cumulative runtime for each method on the synthetic and OpenML benchmarks respectively. PFN-TS is slower than most baselines but faster than its closest competitor BFTS on all 16 scenarios at $T = 1{,}000$, on 8 of 16 at $T = 5{,}000$, and on 1 of 14 at $T = 10{,}000$.

5.3 Ablation Studies

Encoding strategy. We compare adaptive encoding with fixed *Disjoint* and *Joint* encodings. This tests whether sharing observations across arms helps and whether the adaptive rule can choose an appropriate representation. Joint encoding can be less effective when arm histories should remain separate, while disjoint encoding loses the benefit of shared structure. The adaptive rule is competitive with the better fixed encoding; full trajectories are in Figure 9.

Subsampling grid. We vary the geometric base $b \in \{1.5, 2, 3\}$. Smaller bases provide more prefix increments for variance estimation but require more forward passes. The default $b = 2$ balances regret and computational cost; Figure 10 shows the comparison across synthetic scenarios.

Decision rule. We compare PFN-TS with *PFN-PS*, which samples directly from the posterior predictive distribution, and *PFN-Greedy*, which selects the highest predictive mean. PFN-PS performs substantially worse: its samples include observation noise rather than isolating uncertainty about latent mean rewards. Greedy selection can underexplore. These comparisons isolate the role of SubCLT in Thompson sampling; full results are in Figure 13.

Covariance regularization. We also vary the degree of covariance regularization (off-diagonal shrinkage and variance floor). The effect is most prominent in Shuttle, which suffers from vanishing marginal variances; see Figures 11 and 12 for full results.

Table 2: Final cumulative regret on OpenML benchmarks (mean ± sample SE). The rightmost column reports average rank across all eight datasets. Lower is better.

Method	Adult	Covertype	EEGEyeState	GasDrift	MagicTelescope	Mushroom	PageBlocks	Shuttle	Rank
PFN-TS	**1498.4 ± 11.0**	**3098.8 ± 17.2**	**698.3 ± 13.7**	**421.1 ± 23.4**	**1245.4 ± 7.5**	**44.2 ± 2.8**	305.9 ± 6.9	191.9 ± 12.1	1.625
BFTS	1535.5 ± 8.5	3734.3 ± 48.6	2273.5 ± 12.0	818.1 ± 22.8	1488.2 ± 11.1	53.9 ± 2.3	**266.3 ± 4.9**	**106.9 ± 2.5**	2.250
LinTS	1813.4 ± 11.2	4838.6 ± 16.8	3746.9 ± 8.4	2862.0 ± 13.9	2186.8 ± 11.1	551.7 ± 3.7	539.1 ± 6.0	1371.9 ± 6.6	6.250
LinUCB	1651.7 ± 10.5	3709.9 ± 14.1	3615.3 ± 9.4	922.6 ± 6.3	2111.8 ± 9.2	416.5 ± 14.7	372.8 ± 4.7	766.9 ± 10.7	4.750
NeuralTS	4779.1 ± 394.4	4683.8 ± 203.5	5094.1 ± 127.2	7504.2 ± 69.6	2798.6 ± 369.6	102.8 ± 7.4	918.6 ± 382.1	2049.7 ± 824.5	6.625
RFTS	1619.6 ± 13.1	3568.9 ± 19.8	2603.1 ± 11.8	2074.1 ± 129.2	1603.1 ± 17.3	67.9 ± 3.4	298.1 ± 5.6	199.6 ± 21.0	3.625
XGBoostTS	1543.3 ± 11.0	3529.1 ± 17.6	2116.8 ± 9.7	1221.2 ± 11.1	1594.0 ± 9.8	75.9 ± 2.0	305.3 ± 3.9	180.1 ± 24.0	2.875

5.4 Off-policy Evaluation: Drink Less mHealth Trial

To assess performance on real-world logged data, we apply PFN-TS to the Drink Less micro-randomized trial (Bell et al., 2020), a 30-day study in which $n = 349$ participants at risk of hazardous drinking were randomized daily among three push-notification actions (no message, standard message, tailored message) with static propensities $(0.4, 0.3, 0.3)$. The reward is a binary proximal engagement indicator (app opening within the following hour). Following Deng et al. (2026), we unfold the panel into sequential decision points, preserve within-participant temporal order, and simulate each algorithm's policy in offline replay; the complete replication protocol is given in Appendix C.5. Policy value is estimated using self-normalized importance sampling (SNIPS; Swaminathan & Joachims (2015)), with a doubly-robust (DR) estimator as a robustness check (Dudík et al., 2011). Figure 4 shows that PFN-TS has the highest estimated SNIPS value by the final horizon, but performs worse than BFTS on shorter horizons. OPE diagnostics (importance weight distributions and DR estimates) are in Appendix D.3. All ablations are reported in Appendix F.

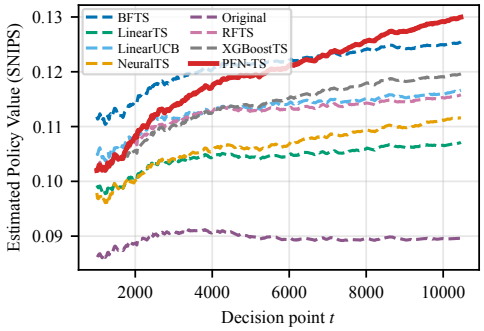

Figure 4: Estimated policy value (SNIPS) on the Drink Less trial. Higher is better.

6 Discussion

PFN-TS shows that tabular in-context learning can serve as an effective reward model for sequential decision-making once efficient uncertainty extraction and adaptive encoding are in place.

Computation. Runtime remains a limitation: SubCLT reduces the number of prefixes used for uncertainty estimation, but each decision still requires repeated PFN evaluations. One-estimator inference, snapshot caching, and batched joint queries make the method practical at moderate horizons; cumulative runtime grows substantially by $T = 10{,}000$ (Appendix G). More efficient inference and history management remain useful extensions.

Finite-sample calibration. SubCLT is asymptotic and can be poorly calibrated when arm histories are short or weakly informative, as in the small-sample coverage diagnostics, early Drink Less horizons, and imbalanced classification tasks such as Shuttle. Covariance shrinkage and a decaying variance floor stabilize the current implementation, but do not replace finite-sample calibration guarantees. Prior-informed pseudo-observations and other regularizers are possible extensions.

Extensions. Posterior contraction and information-gain bounds for PFN priors would sharpen the regret analysis, which leaves information complexity and accumulated encountered sampler error explicit. Although the regret inequality permits any encoding, controlling the posterior-sampling error of joint and adaptive encoding requires further analysis. Continuous rewards also require the stated predictive-tail and latent-mean conditions. Large or infinite action spaces would require candidate generation or action search. SubCLT-style uncertainty estimates may also be useful in active learning, adaptive experimentation, and online model selection.

Deployment. PFN-TS requires application-specific validation before use in high-stakes adaptive decision systems. In settings such as mobile health, evaluation should include subgroup performance, safety constraints, and feedback loops induced by adaptive data collection.

AI USE STATEMENT

Generative AI tools were used to assist with drafting sections of the paper and language editing, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, and write code to implement the methods and experiments. The authors independently checked the mathematical arguments, experimental claims and final text, and take responsibility for the complete contents of the submission.

REPRODUCIBILITY STATEMENT

The appendix gives complete proofs, algorithmic details, data-processing procedures and experimental specifications. Anonymous code accompanying the submission contains the implementations and configurations used to produce the reported results.

REFERENCES

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pp. 127–135, 2013.

Lauren Bell, Claire Garnett, Tianchen Qian, Olga Perski, Henry WW Potts, and Elizabeth Williamson. Notifications to improve engagement with an alcohol reduction app: protocol for a micro-randomized trial. *JMIR research protocols*, 9(8):e18690, 2020.

Patrizia Berti, Irene Crimaldi, Luca Pratelli, and Pietro Rigo. A central limit theorem and its applications to multicolor randomly reinforced urns. *Journal of Applied Probability*, 48(2):527 – 546, 2011.

Hugh A Chipman, Edward I George, and Robert E McCulloch. BART: Bayesian additive regression trees. *Annals of Applied Statistics*, 4(1):266–298, 2010.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pp. 844–853, 2017.

Ruizhe Deng, Bibhas Chakraborty, Ran Chen, and Yan Shuo Tan. BFTS: Thompson sampling with Bayesian additive regression trees. In *Forty-third International Conference on Machine Learning*, 2026.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 1097–1104, 2011.

Edwin Fong, Chris Holmes, and Stephen G Walker. Martingale posterior distributions. *Journal of the Royal Statistical Society: Series B*, 85(5):1357–1391, 2023.

Sandra Fortini and Sonia Petrone. Exchangeability, Prediction and Predictive Modeling in Bayesian Statistics. *Statistical Science*, 40(1):40 – 67, 2025.

Sandra Fortini, Kenyon Ng, Sonia Petrone, Judith Rousseau, and Susan Wei. A principled framework for uncertainty decomposition in TabPFN. *arXiv preprint arXiv:2602.04596v1*, 2026.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Predrag Klasnja, Eric B Hekler, Saul Shiffman, Audrey Boruvka, Daniel Almirall, Ambuj Tewari, and Susan A Murphy. Microrandomized trials: An experimental design for developing just-in-time adaptive interventions. *Health psychology*, 34(S):1220, 2015.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pp. 661–670, 2010.

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 297–306, 2011.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022.

Kenyon Ng, Edwin Fong, David Tyler Frazier, Jeremias Knoblauch, and Susan Wei. TabMGP: Martingale posterior with TabPFN. In *Forty-third International Conference on Machine Learning*, 2026.

Hannes Nilsson, Rikard Johansson, Niklas Åkerblom, and Morteza Haghir Chehreghani. Tree ensembles for contextual bandits. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. In *Forty-third International Conference on Machine Learning*, 2026.

Carlos Riquelme, George Tucker, and Jasper Snoek. Deep Bayesian bandits showdown: An empirical comparison of Bayesian deep networks for Thompson sampling. In *International Conference on Learning Representations*, 2018.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of Thompson sampling. *Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. *Advances in Neural Information Processing Systems*, 28, 2015.

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.

Sattar Vakili, Kia Khezeli, and Victor Picheny. On information gain and regret bounds in Gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 82–90, 2021.

Joaquin Vanschoren, Jan N. Van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, June 2014. ISSN 1931-0145, 1931-0153.

Kelly W. Zhang, Tiffany Cai, Hongseok Namkoong, and Daniel Russo. Contextual Thompson sampling via generation of missing data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Qiong Zhang, Yan Shuo Tan, Qinglong Tian, and Pengfei Li. TabPFN: One model to rule them all? *arXiv preprint arXiv:2505.20003*, 2025b.

Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural Thompson sampling. In *International Conference on Learning Representations*, 2021.

A PROOFS

A.1 PROOF OF THEOREM 1

Proof. Write $\delta_i = m_i(x) - m_{i-1}(x)$ and decompose each increment as

$$\delta_i = \zeta_i + \xi_i,$$

where $\zeta_i = \delta_i - \mathbb{E}[\delta_i \mid \mathcal{F}_{i-1}]$ is the martingale-difference part and $\xi_i = \mathbb{E}[\delta_i \mid \mathcal{F}_{i-1}]$ is the conditional-mean (bias) part.

Part 1: CLT for $\sqrt{n}(m_\infty(x) - m_n(x))$.

We apply the quasi-martingale CLT of Berti et al. (2011) (Proposition 1, restated as Theorem G.4 in Fortini et al. (2026)) to the scalar sequence $(m_n(x))$. We verify its four requirements.

(UI) Uniform integrability of $(m_n(x))$. Put

$$A_i = i^2 \mathbb{E}[\delta_i^2 \mid \mathcal{F}_{i-1}].$$

Conditional Jensen and Assumption 3 give

$$\mathbb{E}\Big[A_i^{2+\epsilon/2}\Big] \le i^{4+\epsilon}\mathbb{E}[|\delta_i|^{4+\epsilon}] \le C.$$

In particular $\sup_i \mathbb{E}[A_i] < \infty$, and hence $\mathbb{E}[\delta_i^2] = \mathbb{E}[A_i]/i^2 = O(i^{-2})$. The martingale $M_n = \sum_{i=1}^n \zeta_i$ is therefore bounded in L^2. The predictable part $B_n = \sum_{i=1}^n \xi_i$ converges absolutely in L^2 because

$$\sum_i \|\xi_i\|_2 \le \sum_i \sqrt{i}\,\|\xi_i\|_2 < \infty$$

by Assumption 2. Together with $m_0(x) \in L^2$, this shows that $(m_n(x))$ is L^2-bounded and therefore uniformly integrable. It also gives the a.s. existence of $m_\infty(x) = m_0(x) + \sum_{i=1}^\infty \delta_i$.

(i) Quasi-martingale condition with $\sqrt{i}$ weights. Theorem G.4 requires $\sum_{i\ge1} \sqrt{i}\,\mathbb{E}[|\xi_i|] < \infty$. Since $\mathbb{E}[|\xi_i|] \le \|\xi_i\|_2$ by Jensen's inequality, this follows immediately from Assumption 2.

(ii) Supremum bound. Since $\left(\sup_{i\ge1}\sqrt{i}\,|\delta_i|\right)^4 \le \sum_{i=1}^\infty i^2\delta_i^4$, Lyapunov's inequality and Assumption 3 imply

$$\mathbb{E}\left[\left(\sup_{i\ge1}\sqrt{i}\,|\delta_i|\right)^4\right] \le \sum_{i=1}^\infty i^2\,\mathbb{E}[\delta_i^4] \le C'\sum_{i=1}^\infty i^{-2} < \infty,$$

so $\sup_i \sqrt{i}\,|\delta_i| \in L^4 \subset L^1$, verifying condition (ii).

(iii) A.s. variance convergence. Apply Fortini et al. (2026), Lemma G.5 with $W_i = i^2\delta_i^2$. The two lemma conditions are: (a) $\sum_{i\ge1} i^{-2}\mathbb{E}[W_i^2] = \sum_i i^2\,\mathbb{E}[\delta_i^4] \le C'\sum_i i^{-2} < \infty$ by Assumption 3; (b) $\mathbb{E}[W_{i+1} \mid \mathcal{F}_i] = (i+1)^2\,\mathbb{E}[\delta_{i+1}^2 \mid \mathcal{F}_i] \to V(x)$ a.s. by Assumption 1. The lemma then gives $n\sum_{i\ge n} W_i/i^2 = n\sum_{i\ge n}\delta_i^2 \to V(x)$ a.s., which is condition (iii).

All four requirements of Theorem G.4 are satisfied, giving $\mathcal{L}(\sqrt{n}(m_\infty(x) - m_n(x)) \mid \mathcal{F}_n) \xrightarrow{w} \mathcal{N}(0, V(x))$, $\mathbb{P}$-a.s.

Part 2: Consistency of $\hat{V}_{s_n}(x)$.

Write $D_j = M_j + R_j$ where $M_j = \sum_{i=t_{j-1}+1}^{t_j}\zeta_i$ and $R_j = \sum_{i=t_{j-1}+1}^{t_j}\xi_i$.

Step 2a: Predictable block variance and drift. Let $I_j = \{t_{j-1}+1, \dots, t_j\}$. On the geometric grid, for all large j, $t_j/t_{j-1} \le b$ and $t_j - t_{j-1} \ge c_b t_{j-1}$ for a constant $c_b > 0$. Hence $w_j = O(t_{j-1})$ and

$$\sum_{i\in I_j} i^{-2} = w_j^{-1}(1 + o(1)).$$

Recall $A_i = i^2\mathbb{E}[\delta_i^2 \mid \mathcal{F}_{i-1}]$. The $L^{2+\epsilon/2}$ bound established in Part 1 makes $\{A_i^2\}$ uniformly integrable. Since $A_i \to V(x)$ a.s., Vitali's theorem gives $A_i \to V(x)$ in L^2; it also follows that

$V(x) \in L^2$. A triangular-array form of Toeplitz' lemma, together with the display above, now gives

$$B_j := w_j \sum_{i \in I_j} \frac{A_i}{i^2} \longrightarrow V(x) \quad \text{in } L^2.$$

We next remove the predictable drift. Let $T_j = \sum_{i \geq t_{j-1}} \sqrt{i}\, \|\xi_i\|_2$; by Assumption 2, $T_j \to 0$. Then

$$\|R_j\|_2 \leq \sum_{i \in I_j} \|\xi_i\|_2 \leq \frac{T_j}{\sqrt{t_{j-1}}}, \qquad w_j \sum_{i \in I_j} \mathbb{E}[\xi_i^2] \leq \frac{w_j}{t_{j-1}} T_j^2 = o(1).$$

Since $\mathbb{E}[\zeta_i^2 \mid \mathcal{F}_{i-1}] = \mathbb{E}[\delta_i^2 \mid \mathcal{F}_{i-1}] - \xi_i^2$, the random predictable block variance

$$Z_j := w_j \sum_{i \in I_j} \mathbb{E}[\zeta_i^2 \mid \mathcal{F}_{i-1}] = B_j - w_j \sum_{i \in I_j} \xi_i^2$$

converges to $V(x)$ in L^1. In particular, $\mathbb{E}[Z_j] = O(1)$, so $\|M_j\|_2 = O(w_j^{-1/2})$. Expanding $D_j^2 = (M_j + R_j)^2$ and using the preceding bounds gives

$$2w_j \mathbb{E}|M_j R_j| \leq 2w_j \|M_j\|_2 \|R_j\|_2 = O(T_j) \to 0,$$

and $w_j \mathbb{E}[R_j^2] \leq (w_j / t_{j-1}) T_j^2 = o(1)$.

Step 2b: Second-moment bound for a single block. By the martingale Burkholder–Rosenthal inequality,

$$\mathbb{E}[M_j^4] \leq C_R \left(\mathbb{E}\left[\left(\sum_{i=t_{j-1}+1}^{t_j} \mathbb{E}[\zeta_i^2 \mid \mathcal{F}_{i-1}] \right)^2 \right] + \sum_{i=t_{j-1}+1}^{t_j} \mathbb{E}[\zeta_i^4] \right)$$

The predictable-variation term equals $w_j^{-2} \mathbb{E}[Z_j^2]$. Because $0 \leq Z_j \leq B_j$ and (B_j) is bounded in L^2, this is $O(w_j^{-2}) = O(t_{j-1}^{-2})$. Moreover, $\mathbb{E}[\zeta_i^4] \leq C' \mathbb{E}[\delta_i^4] = O(i^{-4})$, so the fourth-moment term is $O(t_{j-1}^{-3})$. Therefore $\mathbb{E}[M_j^4] = O(t_{j-1}^{-2})$. Since $w_j = O(t_{j-1})$, this gives $\mathbb{E}[(w_j M_j^2)^2] = O(1)$. Individual blocks do not concentrate to $V(x)$; averaging is essential.

Step 2c: Averaging over J blocks. We first remove the drift contribution in $D_j^2 = (M_j + R_j)^2$. Step 2a gives $w_j \mathbb{E}|M_j R_j| = O(T_j)$ and $w_j \mathbb{E}[(R_j)^2] = O(T_j^2)$, so Cesàro's lemma gives

$$\frac{1}{J} \sum_{j=1}^{J} w_j \big(D_j^2 - M_j^2 \big) \xrightarrow{L^1} 0.$$

It remains to average $w_j M_j^2$. Let

$$Y_j = \mathbb{E}[w_j M_j^2 \mid \mathcal{F}_{t_{j-1}}], \qquad U_j = w_j M_j^2 - Y_j.$$

Then $\{U_j\}$ is a martingale difference sequence with respect to the block filtration $\{\mathcal{F}_{t_j}\}$. By Step 2b,

$$\mathbb{E}\left[\left(\frac{1}{J} \sum_{j=1}^{J} U_j \right)^2 \right] = \frac{1}{J^2} \sum_{j=1}^{J} \mathbb{E}[U_j^2] = O(1/J) \to 0.$$

Finally, we control the predictable part. Since martingale cross terms vanish conditionally,

$$Y_j = \mathbb{E}\left[w_j \sum_{i \in I_j} \mathbb{E}[\zeta_i^2 \mid \mathcal{F}_{i-1}] \,\middle|\, \mathcal{F}_{t_{j-1}} \right].$$

Thus $Y_j = \mathbb{E}[Z_j \mid \mathcal{F}_{t_{j-1}}]$. Although $V(x)$ is random, it is $\mathcal{F}_\infty$-measurable because it is the a.s. limit of (A_i). Therefore

$$\begin{aligned} \|Y_j - V(x)\|_1 \leq{}& \big\| \mathbb{E}[Z_j - V(x) \mid \mathcal{F}_{t_{j-1}}] \big\|_1 \\ &+ \big\| \mathbb{E}[V(x) \mid \mathcal{F}_{t_{j-1}}] - V(x) \big\|_1 \longrightarrow 0. \end{aligned}$$

The first term vanishes because $Z_j \to V(x)$ in L^1; the second vanishes by the martingale convergence theorem. Hence, by Cesàro's lemma,

$$\mathbb{E}\left[\left|\frac{1}{J}\sum_{j=1}^{J}(Y_j - V(x))\right|\right] \to 0.$$

Combining the drift removal, martingale-difference average, and predictable average gives $\hat{V}_{s_n}(x) \xrightarrow{p} V(x)$. □

Remark 4 (On the moment condition)**.** *Assumption 3 is a convenient sufficient condition rather than a tight requirement. The CLT itself can be proved under weaker Lindeberg-type or maximal-increment conditions, and the variance-estimator argument mainly needs enough uniform integrability to control the weighted block squares $w_j D_j^2$. We state the stronger $4+\epsilon$ moment condition because it gives a short proof of both requirements while keeping the theorem aligned with the predictive-CLT assumptions used above.*

A.2 Proof of Theorem 2

Proof. Condition on $\mathcal{G}_t = \sigma(\mathcal{H}_{t-1}^Q, X_t)$, writing $\mathbb{E}_t$ and I_t for expectation and information under this conditional law. Let $Y_{t,a}$ have the reward law if action a were selected, with $\mathbb{E}[Y_{t,a} \mid \Theta, \mathcal{G}_t] = f_0(X_t, a)$. No independence between potential rewards for different actions is needed. Define

$$\mu_{t,a} = \mathbb{E}_t[Y_{t,a}], \quad j_{t,a} = I_t(A_t^*; Y_{t,a}), \quad J_t(q_t) = \sum_a q_{t,a} j_{t,a}.$$

Fix a history and suppress t. For $p_a > 0$, put

$$\mu_a^* = \mathbb{E}[Y_a \mid A^* = a], \quad d_a = \mu_a^* - \mu_a, \quad D_a = \mathrm{KL}(\mathcal{L}(Y_a \mid A^* = a) \,\|\, \mathcal{L}(Y_a)).$$

Terms with $p_a = 0$ are omitted. Bounded rewards and Pinsker's inequality give $|d_a| \le \sqrt{D_a/2}$, while the information decomposition gives

$$j_a = \sum_b p_b \,\mathrm{KL}(\mathcal{L}(Y_a \mid A^* = b) \,\|\, \mathcal{L}(Y_a)) \ge p_a D_a.$$

Let $r_a = \min(p_a, q_a)$, $u_a = p_a - r_a$, and $v_a = q_a - r_a$. Then $\sum_a u_a = \sum_a v_a = \eta$ and $r_a^2 \le p_a q_a$. The policy draw is conditionally independent of Θ, so conditional regret is

$$g(q) = \sum_a p_a \mu_a^* - \sum_a q_a \mu_a = \sum_a r_a d_a + \sum_a u_a \mu_a^* - \sum_a v_a \mu_a.$$

Since $\mu_a^*, \mu_a \in [0,1]$, the last two sums together are at most η. Cauchy–Schwarz yields

$$\begin{aligned} g(q) &\le \sum_a r_a \sqrt{D_a/2} + \eta \le \sqrt{\frac{K}{2}\sum_a r_a^2 D_a} + \eta \\ &\le \sqrt{\frac{K}{2}\sum_a q_a p_a D_a} + \eta \le \sqrt{\frac{K}{2} J(q)} + \eta. \end{aligned}$$

This allows $q_a = 0$: its omitted posterior mass is charged to η. Because A_t^* is a function of (Θ, X_t), data processing gives $J_t(q_t) \le \sum_a q_{t,a} I_t(\Theta; Y_{t,a})$. Hence

$$\begin{aligned} \sum_t \mathbb{E}_Q[J_t(q_t)] &\le \sum_t I_\Pi(\Theta; R_t \mid \mathcal{H}_{t-1}^Q, X_t, A_t) \\ &\le I_\Pi(\Theta; \mathcal{H}_T^Q) = I_T^Q. \end{aligned}$$

The action conveys no additional information given the past and current context. Context information adds nonnegative terms to the chain rule; the second inequality is equality when contexts are conditionally non-informative. If they are informative, the posterior defining p_t must include the current context. Cauchy–Schwarz across rounds and histories now gives

$$\mathbb{E}_Q[\mathrm{Regret}_T] \le \sum_t \mathbb{E}_Q[\sqrt{K J_t(q_t)/2}] + \mathcal{E}_T \le \sqrt{\frac{KT}{2} I_T^Q} + \mathcal{E}_T.$$

□

Corollary 3 (Continuous rewards)**.** *At every encountered history, suppose all latent means $f_0(X_t, a)$ lie in a common interval of width M, uniformly over arms and posterior-supported environments. Suppose, for a fixed $s < \infty$,*

$$\mathbb{E}_t[e^{\lambda(Y_{t,a}-\mu_{t,a})}] \le e^{\lambda^2 s^2/2} \quad \textit{for every } \lambda \in \mathbb{R},\ a, t.$$

Then $\mathbb{E}_Q[\mathrm{Regret}_T] \le \sqrt{2s^2 K T I_T^Q} + M\mathcal{E}_T$.

Proof. The KL variational inequality gives $|d_a| \le \sqrt{2s^2 D_a}$ (Russo & Van Roy, 2016, Appendix A.1). Both μ_a^* and μ_a lie in the same interval of width M, so the unmatched terms above are at most $M\eta$. Thus $g_t(q_t) \le \sqrt{2s^2 K J_t(q_t)} + M\eta_t$, and the same information accounting proves the result. If observation noise is conditionally σ-sub-Gaussian given the latent environment and history, Hoeffding's lemma gives

$$\mathbb{E}_t[e^{\lambda(Y_{t,a}-\mu_{t,a})}] \le e^{\lambda^2\sigma^2/2}\mathbb{E}_t[e^{\lambda(f_0(X_t,a)-\mu_{t,a})}] \le e^{\lambda^2(\sigma^2+M^2/4)/2}.$$

□

The common interval bounds variation across latent environments as well as arms. A bound only on $|f_0(x, a) - f_0(x, b)|$ within each environment does not suffice. In particular, an unrestricted Gaussian-leaf BART prior need not satisfy it; we do not assert that this corollary covers every continuous synthetic prior in the experiments. For rewards in an interval of width L, Theorem 2 instead scales directly to $L\sqrt{KTI_T^Q/2} + L\mathcal{E}_T$.

A.3 Information complexity and a weaker-assumption alternative

The information term concerns the actual policy, not a separate exact-TS run. A policy-independent upper bound is the adaptive information complexity

$$\gamma_T^{\mathrm{ad}}(\Pi) = \sup_{\mu \in \mathfrak{M}_T} I_\Pi(\Theta; \mathcal{H}_T^\mu),$$

where $\mathfrak{M}_T$ consists of nonanticipating policies for the given environment and context mechanism. It can replace I_T^Q in Theorem 2 and Corollary 3. We do not derive its rate for the broad PFN prior.

For disjoint conditional reward models with independent latent arm parameters $\Theta_a \sim \Pi_a$ and conditionally non-informative contexts, the full-history likelihood factors by arm. Policy and context probabilities have no additional parameter dependence given the observed past and cancel from Bayes' formula. Writing $\Pi_{a,T}$ for the resulting arm factor of the full-history posterior,

$$I_\Pi(\Theta; \mathcal{H}_T^Q) = \sum_a \mathbb{E}_Q[\mathrm{KL}(\Pi_{a,T} \,\|\, \Pi_a)].$$

This is a decomposition, not an information-gain bound. The general regret inequality does not require this factorization.

A same-history alternative needs only a within-environment mean-reward span bound L, but leaves an information ratio explicit. Using the notation above, let $\Gamma_t^* = g_t(p_t)^2/J_t(p_t)$ and $\bar{\Gamma}_T^Q = T^{-1}\sum_t \mathbb{E}_Q[\Gamma_t^*]$, with $0/0 = 0$ and a positive numerator over zero interpreted as infinity. Since $0 \le j_{t,a} \le H_t(A_t^*) \le \log K$,

$$J_t(p_t) \le J_t(q_t) + \eta_t \log K, \qquad g_t(q_t) \le g_t(p_t) + L\eta_t.$$

Whenever these quantities are finite, Cauchy–Schwarz and the information budget give

$$\mathbb{E}_Q[\mathrm{Regret}_T] \le \sqrt{T\bar{\Gamma}_T^Q\big(I_T^Q + (\log K)\mathcal{E}_T\big)} + L\mathcal{E}_T.$$

Here Γ_t^* is evaluated on Q's histories and uses information about the current optimal action. Controlling it is an additional requirement. Under bounded rewards, Theorem 2 avoids placing sampler error inside the square root.

Proposition 4 (Sampler error from PPD approximation)**.** *Fix an encountered history $(\mathcal{H}_{t-1}^{Q}, X_t)$ after warm-up and write $x = X_t$. Consider the disjoint conditional-reward comparator with product arm priors and conditionally non-informative contexts described above. All quantities below are evaluated at this history and may be random across runs. For each arm a, let $\mathcal{D}^{(a)}$ denote the arm-specific history used by disjoint encoding, let s_a be its latest refresh size, and write $\mathcal{D}_s^{(a)} = \mathcal{D}_{1:s_a}^{(a)}$. Let $m^{\Pi}(x; \mathcal{D}_s^{(a)})$ be the exact posterior predictive mean under Π, and let $m(x; \mathcal{D}_s^{(a)})$ be the predictive mean used by PFN-TS. Let $\hat{V}^{\Pi}(x; \mathcal{D}_s^{(a)})$ be the SubCLT variance estimate obtained by applying SubCLT to the exact Π-predictive mean sequence, and let $\hat{V}(x; \mathcal{D}_s^{(a)})$ be the SubCLT variance estimate from the TabICL predictive sequence. For a nonnegative posterior variance floor $\underline{v}_{t,a}$ that is fixed given the current history and applied to both paths, define*

$$v_{t,a}^{\Pi} = \max\left\{ \frac{\hat{V}^{\Pi}(x; \mathcal{D}_s^{(a)})}{s_a}, \underline{v}_{t,a} \right\}, \qquad v_{t,a} = \max\left\{ \frac{\hat{V}(x; \mathcal{D}_s^{(a)})}{s_a}, \underline{v}_{t,a} \right\}.$$

Assume $v_{t,a}^{\Pi}, v_{t,a} > 0$. For $\mathcal{D}^{(1:K)} = (\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(K)})$, define

$$G_t^{\Pi}(x, \mathcal{D}^{(1:K)}) = \bigotimes_{a=1}^{K} \mathcal{N}\Big(m^{\Pi}(x; \mathcal{D}_s^{(a)}), v_{t,a}^{\Pi} \Big),$$

$$G_t(x, \mathcal{D}^{(1:K)}) = \bigotimes_{a=1}^{K} \mathcal{N}\Big(m(x; \mathcal{D}_s^{(a)}), v_{t,a} \Big).$$

Let $S_t^{\Pi}(\cdot \mid x, \mathcal{D}^{(1:K)})$ be the exact posterior sampling distribution over action-value vectors under Π. Suppose PFN-TS samples an action-value vector from $G_t(x, \mathcal{D}^{(1:K)})$ at round t, and let

$$\rho_t = \mathrm{TV}\Big(S_t^{\Pi}(\cdot \mid x, \mathcal{D}^{(1:K)}), G_t^{\Pi}(x, \mathcal{D}^{(1:K)}) \Big)$$

denote the encountered Gaussian approximation error for exact posterior sampling under Π, including the error from comparing exact full-history posterior sampling with the snapshot posterior at refresh sizes $(s_1, \dots, s_K)$. Define

$$\Delta_{t,a}(x, \mathcal{D}^{(a)}) = m(x; \mathcal{D}_s^{(a)}) - m^{\Pi}(x; \mathcal{D}_s^{(a)}), \qquad r_{t,a}(x, \mathcal{D}^{(a)}) = \frac{v_{t,a}}{v_{t,a}^{\Pi}}.$$

Then

$$\eta_t \le \rho_t + \frac{1}{2} \left(\sum_{a=1}^{K} \left[r_{t,a}(x, \mathcal{D}^{(a)}) + \frac{\Delta_{t,a}^2(x, \mathcal{D}^{(a)})}{v_{t,a}^{\Pi}} - 1 - \log r_{t,a}(x, \mathcal{D}^{(a)}) \right] \right)^{1/2}. \tag{6}$$

Proof of Proposition 4. The action is a measurable function of the sampled action-value vector, so data processing gives

$$\eta_t \le \mathrm{TV}\Big(G_t(x, \mathcal{D}^{(1:K)}), S_t^{\Pi}(\cdot \mid x, \mathcal{D}^{(1:K)}) \Big).$$

By the triangle inequality for total variation,

$$\mathrm{TV}(G_t, S_t^{\Pi}) \le \mathrm{TV}(G_t, G_t^{\Pi}) + \mathrm{TV}(G_t^{\Pi}, S_t^{\Pi}),$$

where all terms are evaluated at a fixed $(x, \mathcal{D}^{(1:K)})$. The last term is bounded by ρ_t. It remains to bound the TV distance between the two diagonal Gaussians. By Pinsker's inequality and the KL divergence between diagonal Gaussians,

$$\begin{aligned} \mathrm{TV}(G_t, G_t^{\Pi}) &\le \sqrt{\frac{1}{2} \mathrm{KL}(G_t \,\|\, G_t^{\Pi})} \\ &= \frac{1}{2} \left(\sum_{a=1}^{K} \left[r_{t,a}(x, \mathcal{D}^{(a)}) + \frac{\Delta_{t,a}^2(x, \mathcal{D}^{(a)})}{v_{t,a}^{\Pi}} - 1 - \log r_{t,a}(x, \mathcal{D}^{(a)}) \right] \right)^{1/2}. \end{aligned}$$

This proves the pointwise bound in Equation (6). Since $\eta_t \le 1$, its right-hand side can be capped at one before taking expectations under Q. Summing these expectations, and charging at most one for each warm-up round, controls $\mathcal{E}_T$. □

Remark 5 (Interpreting the residual term)**.** *The residual ρ_t is the Gaussian approximation error for exact posterior sampling under Π. It includes the normal-approximation error, the finite-grid variance error, and the error from using the latest snapshot rather than the full history. The predictive CLT in Theorem 1 is pointwise and weak; weak convergence alone does not imply convergence in total variation. Controlling $\mathcal{E}_T$ requires bounds on normal approximation, variance estimation, and snapshot staleness on histories encountered under Q. A fixed geometric grid need not make the snapshot-to-full-history discrepancy vanish, even when refresh sizes diverge. We do not establish these rates here.*

Remark 6 (Variance error from predictive-mean paths)**.** *The variance term in Proposition 4 can also be controlled by approximation error of the predictive means along the SubCLT grid. Let $t_0 < \cdots < t_J$ be the grid for an arm history $\mathcal{D}^{(a)}$, and write*

$$D_j = m(x; \mathcal{D}^{(a)}_{1:t_j}) - m(x; \mathcal{D}^{(a)}_{1:t_{j-1}}), \qquad D_j^{\Pi} = m^{\Pi}(x; \mathcal{D}^{(a)}_{1:t_j}) - m^{\Pi}(x; \mathcal{D}^{(a)}_{1:t_{j-1}}).$$

If $\hat{V} = J^{-1}\sum_{j=1}^{J} w_j D_j^2$ and $\hat{V}^{\Pi} = J^{-1}\sum_{j=1}^{J} w_j (D_j^{\Pi})^2$, then

$$|\hat{V} - \hat{V}^{\Pi}| \le \frac{1}{J}\sum_{j=1}^{J} w_j |D_j - D_j^{\Pi}| \left(|D_j| + |D_j^{\Pi}|\right).$$

Moreover, with $e_i(x; \mathcal{D}^{(a)}) = |m(x; \mathcal{D}^{(a)}_{1:i}) - m^{\Pi}(x; \mathcal{D}^{(a)}_{1:i})|$,

$$|D_j - D_j^{\Pi}| \le e_{t_j}(x; \mathcal{D}^{(a)}) + e_{t_{j-1}}(x; \mathcal{D}^{(a)}).$$

Thus the variance component is also controlled by TabICL's approximation to the exact PPD predictive means, but along the full SubCLT prefix path rather than only at the final history. Applying the same deterministic floor to both posterior variances does not amplify this discrepancy because the map $u \mapsto \max\{u, \underline{v}_{t,a}\}$ is one-Lipschitz.

Remark 7 (What remains unquantified)**.** *Theorem 2 leaves information complexity and accumulated encountered sampler error explicit. Bounds for a particular linear, kernel, or tree prior do not automatically transfer to the broad prior represented by TabICL. Model-specific information bounds and control of $\sum_t \mathbb{E}_Q[\eta_t]$ are needed for a PFN-TS regret rate. For continuous rewards, the additional assumptions of Corollary 3 must also hold.*

B Algorithm Details

B.1 Algorithm Specifications

The implementation is presented in three layers. Algorithm 1 computes a SubCLT posterior variance estimate for a scalar query or a posterior covariance estimate for a vector-valued query. Algorithm 2 uses this estimate to draw rewards and select an arm under either encoding. Algorithm 3 gives the complete adaptive procedure, including warm-up, geometric snapshot caching, and CRPS-based encoding selection.

For a history of size n, let $G_b(n) = \{t_0, \ldots, t_J\}$ denote the geometric snapshot grid, where $t_0 = 2$, $t_{j+1} = \max\{t_j + 1, \lfloor bt_j \rfloor\}$, and $t_J \le n < t_{J+1}$. We write $J_b(n) = |G_b(n)| - 1$ for its number of increments.

For a one-row query, $\widehat{\Sigma}$ is the scalar SubCLT posterior variance estimate $\hat{V}_{s_n}/s_n$ established by Theorem 1. For a multirow query, the same construction is the empirical vector-valued posterior covariance extension. The theorem does not assert a joint CLT.

Algorithm 1 Subsampled CLT (SUBCLT)

Require: History $\mathcal{D} = \{(x_i, r_i)\}_{i=1}^{n}$, query matrix X^{query}, and geometric snapshot grid $G_b(n)$
Hyperparameters: PFN model

$G_b(n) = \{t_0, \ldots, t_J\}$; $s_n \leftarrow t_J$
for $j = 0, \ldots, J$ **do**
  Retrieve or construct the cached snapshot conditioned on $\mathcal{D}_{1:t_j}$
  $\boldsymbol{m}_j \leftarrow \mathrm{PFN}(X^{\text{query}}; \mathcal{D}_{1:t_j})$
end for
for $j = 1, \ldots, J$ **do**
  $\boldsymbol{D}_j \leftarrow \boldsymbol{m}_j - \boldsymbol{m}_{j-1}$
  $w_j \leftarrow t_j t_{j-1}/(t_j - t_{j-1})$
end for
$\widehat{\Sigma} \leftarrow (s_n J)^{-1} \sum_{j=1}^{J} w_j \boldsymbol{D}_j \boldsymbol{D}_j^{\top}$
return $\boldsymbol{m}_J, \widehat{\Sigma}$

Algorithm 2 PFN-TS arm-selection draw

Require: Encoding $e \in \{\text{disjoint}, \text{joint}\}$, histories, query x, and counts $n_{1:K}$
Hyperparameters: Number of arms K, geometric base b, covariance shrinkage strength ρ, optional floor initial standard deviation σ_0, and floor prior count n_0

if $e = \text{disjoint}$ **then**
  for $a = 1, \ldots, K$ **do**
    Retrieve or extend $G^{(a)} \leftarrow G_b(n_a)$
    $(m_a, \hat{\sigma}_a^2) \leftarrow \text{SUBCLT}(\mathcal{D}^{(a)}, [x], G^{(a)})$
  end for
  $\boldsymbol{m} \leftarrow (m_1, \ldots, m_K)^{\top}$; $\widehat{\Sigma} \leftarrow \mathrm{diag}(\hat{\sigma}_1^2, \ldots, \hat{\sigma}_K^2)$
else
  Retrieve or extend $G \leftarrow G_b(|\mathcal{D}|)$; $J \leftarrow J_b(|\mathcal{D}|)$
  $X^{\text{query}} \leftarrow [(x, 1), \ldots, (x, K)]$
  $(\boldsymbol{m}, \widehat{\Sigma}) \leftarrow \text{SUBCLT}(\mathcal{D}, X^{\text{query}}, G)$
  $\lambda_J \leftarrow \rho K/(J + \rho K)$ ▷ use endpoint conventions for $\rho \in \{0, \infty\}$
  $\widehat{\Sigma} \leftarrow (1 - \lambda_J)\widehat{\Sigma} + \lambda_J \,\mathrm{diag}(\widehat{\Sigma})$
end if
if the posterior variance floor is enabled **then**
  for $a = 1, \ldots, K$ **do**
    $q_a \leftarrow \max\{0, \sigma_0^2 n_0/(n_a + n_0) - \widehat{\Sigma}_{aa}\}$
  end for
  $\widehat{\Sigma} \leftarrow \widehat{\Sigma} + \mathrm{diag}(q_1, \ldots, q_K)$
end if
Draw $\widetilde{\boldsymbol{r}} \sim \mathcal{N}(\boldsymbol{m}, \widehat{\Sigma})$
return $\arg\max_a \widetilde{r}_a$

Algorithm 3 PFN-TS with adaptive encoding

Require: Bandit interaction providing context X_t and reward R_t after action A_t
Hyperparameters: Horizon T, arms K, warm-up cycles τ, switch times $S = \{s_1 < \cdots < s_L\}$, joint threshold K_{thr}, and PFN-TS draw procedure

$e^* \leftarrow$ disjoint if $K < K_{\text{thr}}$, otherwise joint; $e^\dagger \leftarrow$ the other encoding
Initialize histories, snapshot caches, and cumulative CRPS for e^* and $e^\dagger$
for $t = 1, \ldots, T$ **do**
  Observe X_t
  if $t \le \tau K$ **then**
    $A_t \leftarrow$ next arm in randomized round-robin order
  else
    $A_t \leftarrow$ PFNTSDRAW$(e^*, X_t, \ldots)$
  end if
  Observe R_t and update the history for e^*
  if dual caching is active **then**
    Update the history for $e^\dagger$
  end if
  Extend each maintained cache if its history reaches a geometric grid point
  if $t \in S$ **then**
    Score e^* and $e^\dagger$ on the new interval using their cached snapshots
    Add the interval scores to the cumulative CRPS totals
    Swap e^* and $e^\dagger$ if $e^\dagger$ has lower cumulative CRPS
    if $t = s_L$ **then**
      Discard the history and cache for $e^\dagger$; stop dual caching
    end if
  end if
end for

B.2 Caching and computational accounting

A snapshot is conditioned exactly once, when its geometric grid point is first reached, and is then retained. At a decision time, however, SubCLT must query every retained snapshot to recover the predictive-mean path at the current context. Thus the cached conditioning count and the cached prediction count are distinct. With n_a observations on arm a, disjoint encoding stores $O(\log_b n_a)$ snapshots and makes $O(\sum_a \log_b n_a)$ scalar-query calls per decision. Joint encoding stores $O(\log_b t)$ shared snapshots and makes $O(\log_b t)$ prediction calls per decision, each on one K-row query batch. CRPS checks reuse these conditioned snapshots, although they still require cached CDF evaluations for the observations being scored.

C Experimental Setup and Data

C.1 Common Experimental Settings

All experiments use global random seed 42, with per-replication seeds derived from it.

Compute environment. PFN-TS and NeuralTS used one NVIDIA L40S GPU, whereas LinTS, LinUCB, BFTS, RFTS, and XGBoostTS used 4× Intel(R) Xeon(R) Gold 6448H processors. Wall-clock results therefore reflect each implementation's intended execution mode rather than a hardware-matched comparison. Experiments used Python 3.11, PyTorch 2.11.0+cu128, and TabICL pinned to commit `46b91961db4f`.

C.2 Method-Specific Hyperparameters and Tuning

PFN-TS. Unless stated otherwise in the ablation studies, PFN-TS uses the following defaults across all experiments:

- **Subsampling base**: $b = 2$ (geometric grid).

- **Context window**: no truncation; the full observed history for the relevant arm/encoding is retained.
- **Warm-up**: $\tau = 5$ round-robin pulls per arm before Thompson sampling.
- **Backbone model**: one TabICL (Qu et al., 2026) estimator with KV-cache enabled for the fixed training set per arm.
- **Encoding**: adaptive CRPS-based selection between disjoint and joint representations at switch checkpoints (Section 3.3).
- **Covariance shrinkage**: $\rho = 1$ under joint encoding.
- **Variance floor**: $\sigma_0 = 0.1$ and $n_0 = 10$ for binary rewards; disabled for continuous rewards.

Baselines. Baseline implementations and fixed settings follow Deng et al. (2026) unless noted below.

- **LinTS**: exploration parameter ν tuned from $\{1.0, 0.1, 0.01\}$ on a pilot run; $\lambda = 1$ (fixed).
- **LinUCB**: exploration parameter $\alpha = 1$ (fixed).
- **NeuralTS**: λ tuned from $\{1, 0.1, 0.01, 0.001\}$ and ν from $\{0.1, 0.01, 0.001, 0.0001\}$ on a pilot run.
- **BFTS**: default settings of (Deng et al., 2026) with $m = 100$ trees per arm, depth-exponential prior with $\alpha = 0.45$, Dirichlet-sparse split probabilities, $n_{\text{post}} = 500$, $n_{\text{burn}} = 500$, 4 chains, logarithmic refresh schedule, and the $\tau = 5$ warm-up convention.
- **RFTS, XGBoostTS**: default settings of Nilsson et al. (2024) with $\nu = 1$ (fixed).

C.3 Synthetic DGP Specifications

The encoding, subsampling, and decision-rule ablations use $T = 2000$ and $R = 5$ on the same eight synthetic scenarios. The interaction proceeds at each round $t = 1, \dots, T$ as follows: a context $X_t \in \mathbb{R}^P$ arrives, the agent selects arm $a_t \in \{1, \dots, K\}$, and observes reward $Y_t = \mu_{a_t}(X_t) + \varepsilon_t$ with independent noise. Unless stated otherwise, $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$.

Linear ($P = 10,\ K = 3$). Contexts are i.i.d. $X_t \sim \text{Unif}([0,1]^P)$. For each arm a, coefficients $\beta_a \in \mathbb{R}^P$ are drawn with $\beta_{a,j} \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$. The mean reward is $\mu_a(x) = \beta_a^\top x$.

Friedman variants ($P \in \{5, 20\},\ K = 2$). Contexts are i.i.d. $X_t \sim \text{Unif}([0,1]^P)$. The base Friedman1 function is

$$f(x_1, \dots, x_5) = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5.$$

For FRIEDMAN2 and FRIEDMAN3, the first four coordinates are rescaled as $x_1' = 100x_1$, $x_2' = 40\pi + 520\pi x_2$, $x_3' = x_3$, $x_4' = 1 + 10x_4$, with

$$f_2(x) = \tfrac{1}{125}\sqrt{(x_1')^2 + \left(x_2'x_3' - (x_2'x_4')^{-1}\right)^2}, \qquad f_3(x) = \tfrac{1}{0.1}\arctan\!\left(\tfrac{x_2'x_3' - (x_2'x_4')^{-1}}{x_1'}\right).$$

Arm 1 uses the scenario-specific function (f, f_2, or f_3). Arm 2 uses a *shared* variant $\mu_2(x) = f(x_1, \dots, x_5) + 5\sin(\pi x_1 x_2)$, or a *disjoint* variant $\mu_2(x) = f(x_P, x_{P-1}, \dots, x_1)$ (reversed feature order). Sparse variants increase context dimensionality to $P = 20$ while keeping the reward functions low-dimensional.

Friedman-Heteroscedastic. Same mean functions as the shared Friedman variant, with arm-specific noise: for each arm a, $\sigma_a^2 = 10^{U_a}$ where $U_a \sim \text{Unif}(-1, 1)$ is drawn once per replication.

SynBART ($P = 4,\ K = 3$). Contexts are i.i.d. $X_t \sim \text{Unif}([0,1]^P)$. Each arm's mean reward function $\mu_a(\cdot)$ is sampled from a BART prior (Chipman et al., 2010) with $m = 100$ trees, depth-geometric splitting prior with $\alpha = 0.45$, Gaussian leaf prior with $\kappa = 2$, and Dirichlet-sparse split probabilities with $(\zeta, \xi) = (1, 1)$. Noise variance is fixed to $\sigma^2 = 0.01$. This DGP is correctly specified for BFTS and represents a challenging nonlinear benchmark for methods with a different prior.

C.4 OpenML Dataset Details

We evaluate on eight OpenML (Vanschoren et al., 2014) classification datasets: Adult, Covertype, EEGEyeState, GasDrift, MagicTelescope, Mushroom, PageBlocks, and Shuttle. Each dataset defines a contextual bandit: the feature vector is the context, each class label is an arm, and the reward is 1 for selecting the true class and 0 otherwise. We follow the preprocessing of Zhang et al. (2021); Deng et al. (2026): features are standardized, categorical variables are one-hot encoded, and the interaction horizon is capped at $T = 10000$ (or the dataset size if smaller). Mushroom therefore has $T = 8124$, PageBlocks has $T = 5473$, and the other six datasets have $T = 10000$. LinTS and NeuralTS hyperparameters are selected on a separate low-budget pilot run (one replication with a different seed) to avoid post-hoc tuning.

Remark 8. *Pilot tuning gives LinTS and NeuralTS a limited offline tuning budget relative to a strictly online setting. PFN-TS uses the same default settings across datasets.*

C.5 Drink Less Data and OPE Protocol

The Drink Less data come from a micro-randomized trial (MRT) (Bell et al., 2020) involving $n = 349$ participants over 30 days. At 8:00 p.m. daily (local time), participants were randomized among three push-notification actions with static propensities $(0.4, 0.3, 0.3)$: (i) no message, (ii) a standard message, and (iii) a message drawn from a bank of tailored variants. The binary reward $R_{i,d} \in \{0, 1\}$ indicates proximal engagement: whether participant i opened the app within one hour of the 8 p.m. decision point on day d.

The panel $\{(X_{i,d}, A_{i,d}, R_{i,d})\}$ is unfolded into a sequential stream of $T = 349 \times 30 = 10470$ decision points, ordered by day d within each participant i and then interleaved across participants. This ordering preserves within-participant temporal structure while allowing sequential simulation of adaptive algorithms. The context X_t includes static participant covariates (`AUDIT_score`, `age`, `gender`, `employment_type`), the time-varying covariate `days_since_download`, and a one-hot user identifier to capture individual fixed effects. Features are standardized prior to bandit simulation.

OPE methodology. Since the algorithms cannot be deployed prospectively, we use offline replay-style sequential simulation (Li et al., 2011): at each step t, the algorithm proposes an action and is updated only if its proposal matches the logged action A_t, effectively using the logged data as an unbiased estimator of the interactive bandit environment under the logging propensities. Policy value is estimated via self-normalized importance sampling (SNIPS; Swaminathan & Joachims (2015)):

$$\widehat{V}_{\mathrm{SNIPS}}(\pi) = \frac{\sum_t w_t(\pi)\, R_t}{\sum_t w_t(\pi)}, \qquad w_t(\pi) = \frac{\pi(A_t \mid X_t)}{\pi_0(A_t)},$$

where $\pi_0(a) \in \{0.4, 0.3, 0.3\}$ are the known logging propensities. Uncertainty is quantified using 50 completed user-level bootstrap/policy-seed replications, with every stochastic agent freshly initialized in each replication A doubly-robust (DR) estimator (Dudík et al., 2011) is provided as a robustness check (see Appendix D.3).

Since the logging propensities are bounded away from zero, importance weights are bounded by $1/\min_a \pi_0(a) = 10/3$, ruling out severe positivity violations. Weight diagnostics are reported in Appendix D.3.

D Additional Benchmark Results

D.1 Full Synthetic Regret Curves

Figure 5 provides cumulative regret trajectories across all synthetic scenarios described in Appendix C.3.

D.2 Full OpenML Results

Figure 6 provides cumulative regret trajectories across all eight OpenML datasets.

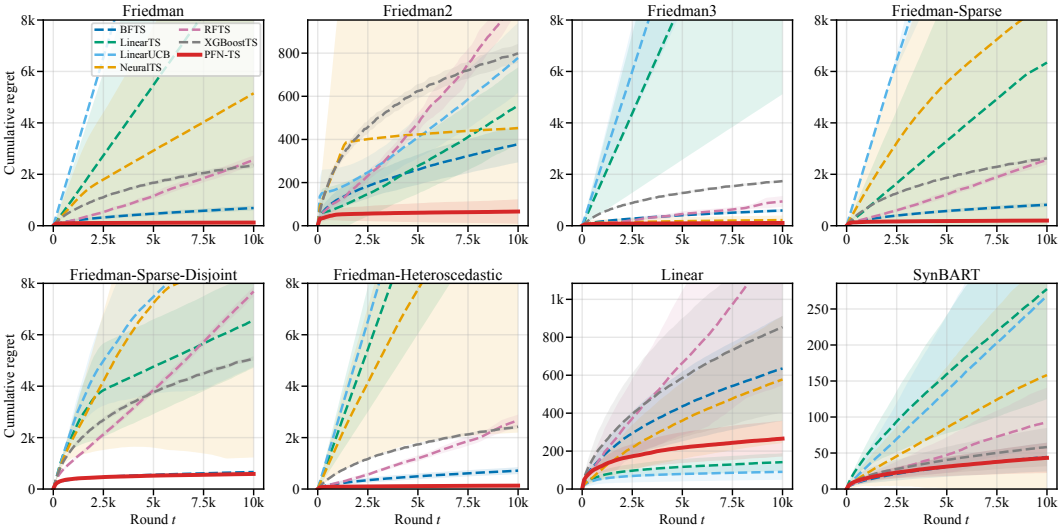

Figure 5: Cumulative regret trajectories across all eight synthetic scenarios (mean ± population SD).

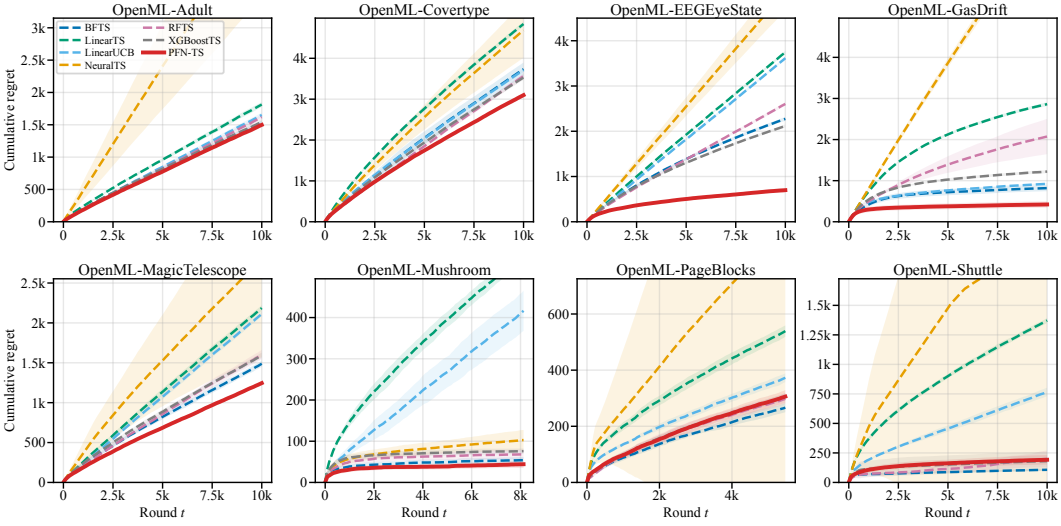

Figure 6: Cumulative regret trajectories on all eight OpenML datasets (mean ± population SD).

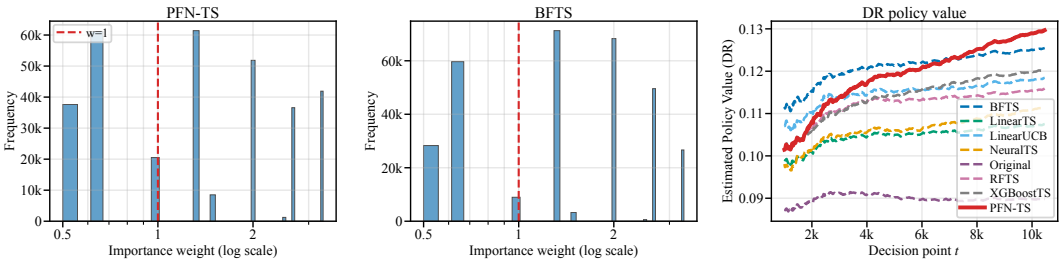

Figure 7: Drink Less OPE diagnostics. Left and centre: empirical importance weight distributions for PFN-TS and BFTS on a log scale. Right: policy value estimated with the doubly-robust estimator.

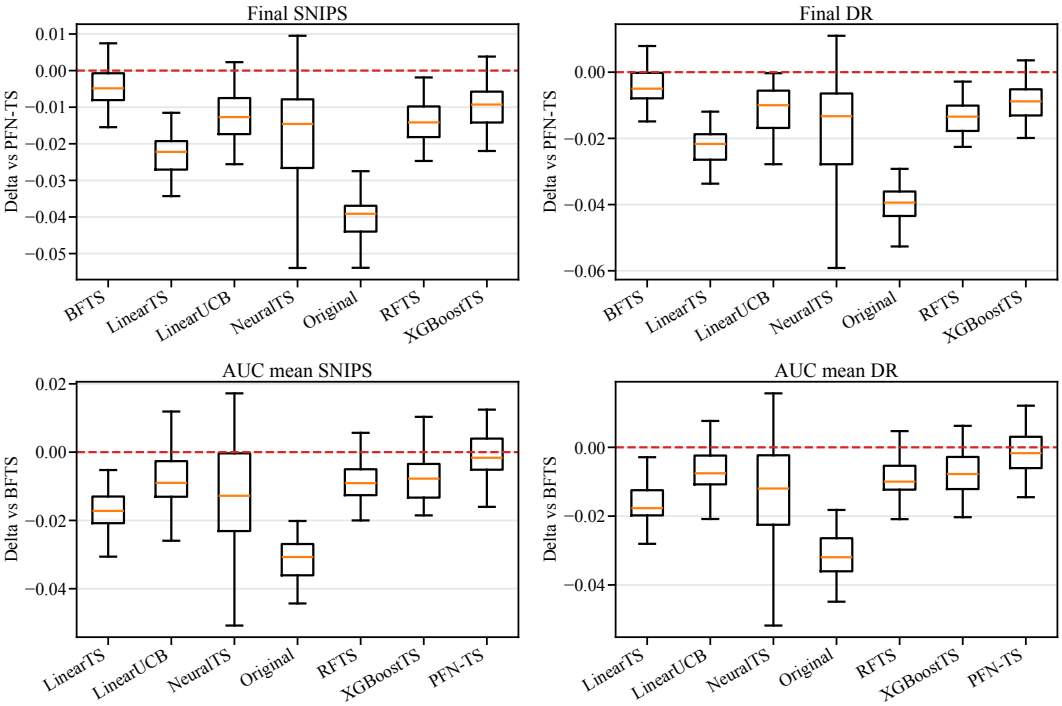

Figure 8: Bootstrap distributions of the policy-value gap between each method and the best-performing baseline on the Drink Less trial. Top: gap at the final horizon. Bottom: gap integrated over the full horizon (AUC mean). Positive values indicate that the method outperforms the best baseline.

D.3 OPE Diagnostics

Importance weight diagnostics. Figure 7 (left and centre) shows the empirical importance weight distributions for PFN-TS and BFTS.

Doubly-robust estimates. For each bootstrap replicate, a per-arm ridge outcome model $\hat{q}(x, a)$ is fit via 2-fold cross-fitting and used to form the DR estimate

$$\widehat{V}_{\mathrm{DR}}(\pi) = \frac{1}{n}\sum_{i=1}^{n}\Big[\sum_{a}\pi(a \mid x_i)\,\hat{q}(x_i, a) + w_i\big(r_i - \hat{q}(x_i, a_i)\big)\Big].$$

Figure 7 (right) shows the DR policy value curve over the horizon; the pattern closely tracks the SNIPS estimates, confirming robustness to the choice of estimator.

E SubCLT Coverage Diagnostics

To isolate the quality of the SubCLT approximation from adaptive data collection, we run offline regression diagnostics on the Linear and Friedman DGPs. The DGPs match those in Appendix C.3, but the bandit structure is removed: we treat each arm in isolation and evaluate coverage on arm 0 only (Linear: $P = 10$, $\sigma^2 = 1$; Friedman: $P = 5$, Friedman1 variant with shared correlation, $\sigma^2 = 1$). For each DGP and training size $n \in \{16, 64, 256, 1024\}$, we independently resample the DGP parameters and training set for every dataset. Coverage is evaluated for the latent mean reward $f_0(x)$, not for a noisy reward draw.

We compare four nominal 95% latent-mean intervals. TabICL-SubCLT uses the predictive-sequence approximation $\mathcal{N}(m_{s_n}(x), \hat{V}_{s_n}(x)/s_n)$ on the geometric prefix grid with base $b = 2$. TabICL-FullCLT instead evaluates every nonempty prefix and uses center $m_n(x)$ and variance

$$\widehat{\sigma}^2_{\mathrm{FullCLT}}(x) = \frac{1}{n(n-1)} \sum_{k=2}^{n} k^2 \{m_k(x) - m_{k-1}(x)\}^2, \tag{7}$$

with the $k = 1$ increment omitted because TabICL does not expose the zero-context prediction $m_0(x)$. Both TabICL intervals are the corresponding mean plus or minus $z_{0.975}$ times the estimated standard deviation. Gaussian-LM fits a linear model with an intercept. Writing $\tilde{x} = (1, x^\top)^\top$, $d = P + 1$ $\widehat{\sigma}^2 = \mathrm{RSS}/(n-d)$ and $\widehat{\beta}$ the least squares estimator, its mean-response interval is

$$\tilde{x}^\top \widehat{\beta} \pm t_{0.975, n-d} \sqrt{\widehat{\sigma}^2 \tilde{x}^\top (\widetilde{X}^\top \widetilde{X})^{-1} \tilde{x}}. \tag{8}$$

BART uses the 2.5th and 97.5th percentiles of 500 posterior draws of the latent mean ($m = 100$ trees, $n_{\mathrm{post}} = 500$, $n_{\mathrm{burn}} = 500$, $\alpha = 0.95$, $\beta = 2.0$). We use the default BART prior of Chipman et al. with their recommended settings, which differs from the BFTS choices in Appendix C.2.

For each DGP and training size, let $[L_{rq}, U_{rq}]$ denote the interval for query $q \in \{1, \ldots, 50\}$ in independently generated dataset $r \in \{1, \ldots, 100\}$ and let f_{rq} be its true latent mean. We first compute the dataset-level coverage rate and mean interval length,

$$C_r = \frac{1}{50} \sum_{q=1}^{50} \mathbf{1}\{L_{rq} \le f_{rq} \le U_{rq}\}, \qquad A_r = \frac{1}{50} \sum_{q=1}^{50} (U_{rq} - L_{rq}). \tag{9}$$

The plotted coverage and mean interval length are $\widehat{C} = 100^{-1} \sum_{r=1}^{100} C_r$ and $\widehat{A} = 100^{-1} \sum_{r=1}^{100} A_r$, respectively. Thus each plotted point summarizes 100 independently generated datasets (seeds 1000–1099) and 5,000 query intervals. When reported, standard errors are the sample standard deviations of $\{C_r\}_{r=1}^{100}$ and $\{A_r\}_{r=1}^{100}$ divided by $\sqrt{100}$.

F Ablation Studies

F.1 Encoding Strategy

Figure 9 compares PFN-TS (adaptive), PFN-TS-Disjoint (fixed separate model), and PFN-TS-Joint (arm index concatenated to context) across the eight synthetic scenarios.

F.2 Subsampling Grid

Figure 10 compares PFN-TS with geometric base $b \in \{1.5, 2, 3\}$ across the synthetic scenarios.

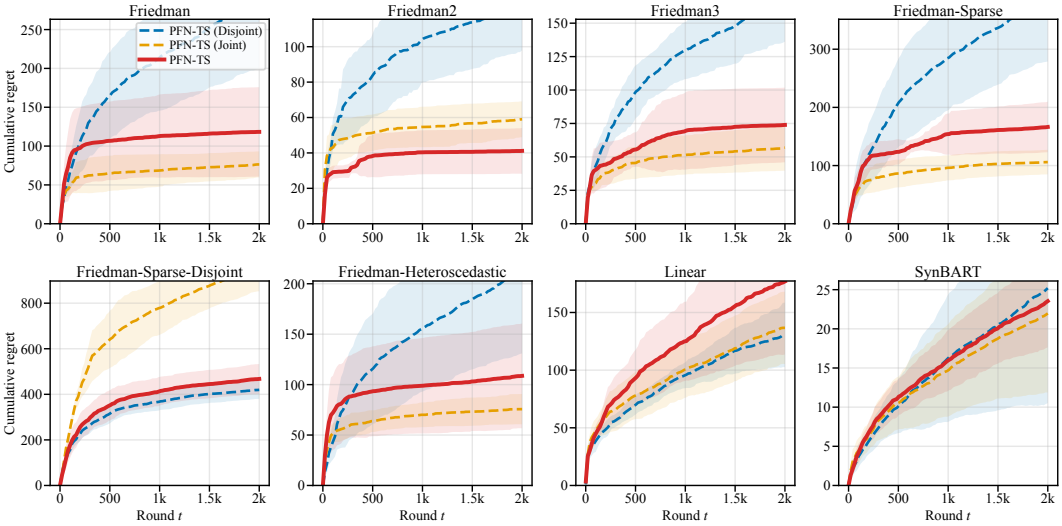

Figure 9: Encoding ablation: cumulative regret trajectories comparing the adaptive encoding rule against *Disjoint* (separate context per arm) and *Joint* (arm index appended to context) across all synthetic scenarios (mean $\pm$ population SD).

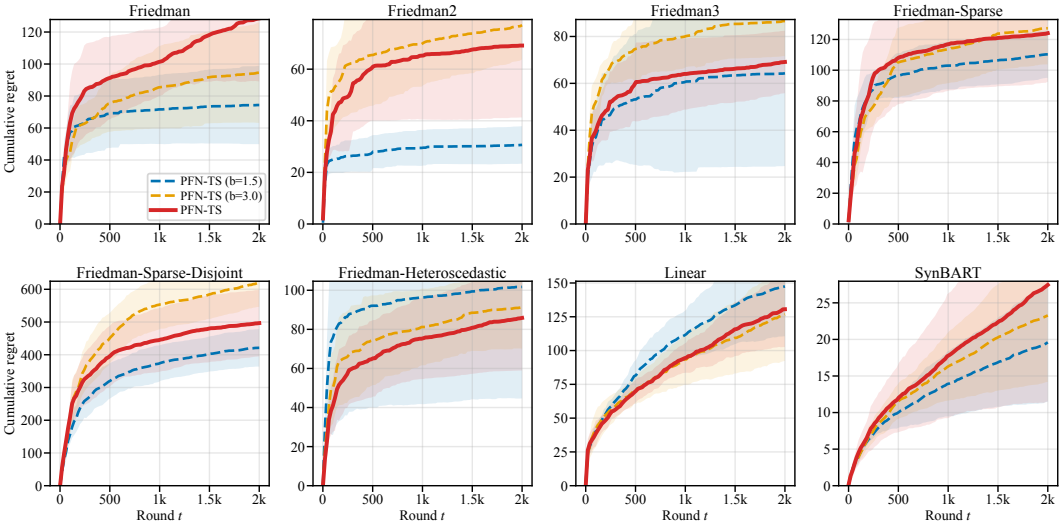

Figure 10: Subsampling grid ablation: cumulative regret trajectories for geometric base $b \in \{1.5, 2, 3\}$ across all synthetic scenarios (mean $\pm$ population SD).

F.3 Covariance Regularization

Figures 11 and 12 report the covariance shrinkage and variance floor ablations with fixed joint encoding on OpenML Adult, MagicTelescope, Mushroom and Shuttle ($T = 2000$, $R = 5$).

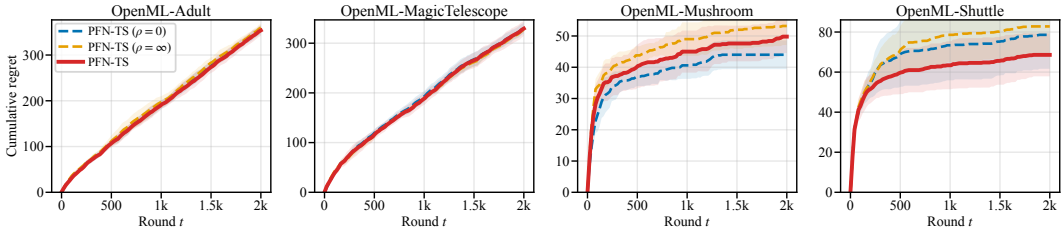

Figure 11: Covariance-shrinkage ablation: cumulative regret trajectories for PFN-TS with fixed joint encoding and $\rho \in \{0, 1, \infty\}$ across four OpenML tasks (mean $\pm$ population SD). The variance floor is enabled throughout.

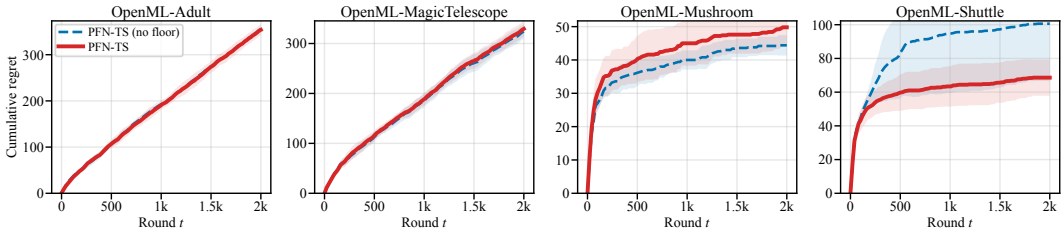

Figure 12: Variance-floor ablation: cumulative regret trajectories for PFN-TS with fixed joint encoding, comparing the default decaying floor with no floor across four OpenML tasks (mean $\pm$ population SD). Shrinkage is fixed at $\rho = 1$.

F.4 Decision Rule

Figure 13 compares Thompson sampling (PFN-TS), posterior predictive sampling (PFN-PS), and a greedy baseline (highest predictive mean only).

G Wall-clock time

Figures 14 and 15 report cumulative wall-clock time on the synthetic and OpenML benchmarks, respectively. PFN-TS has reasonable runtime at smaller horizons but becomes slower at $T = 10000$.

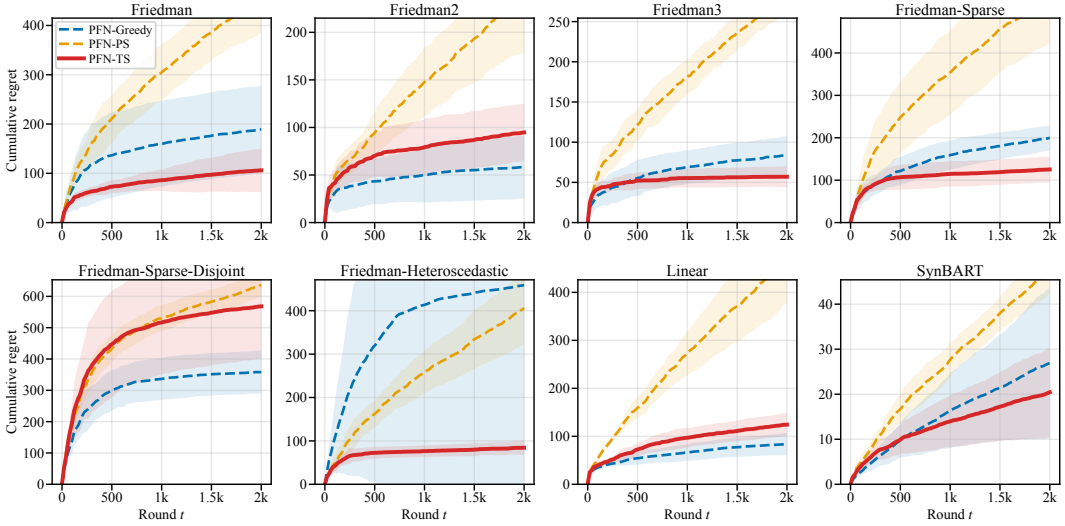

Figure 13: Decision-rule ablation: cumulative regret trajectories comparing Thompson sampling (PFN-TS), PFN-PS, and greedy across synthetic scenarios (mean ± population SD).

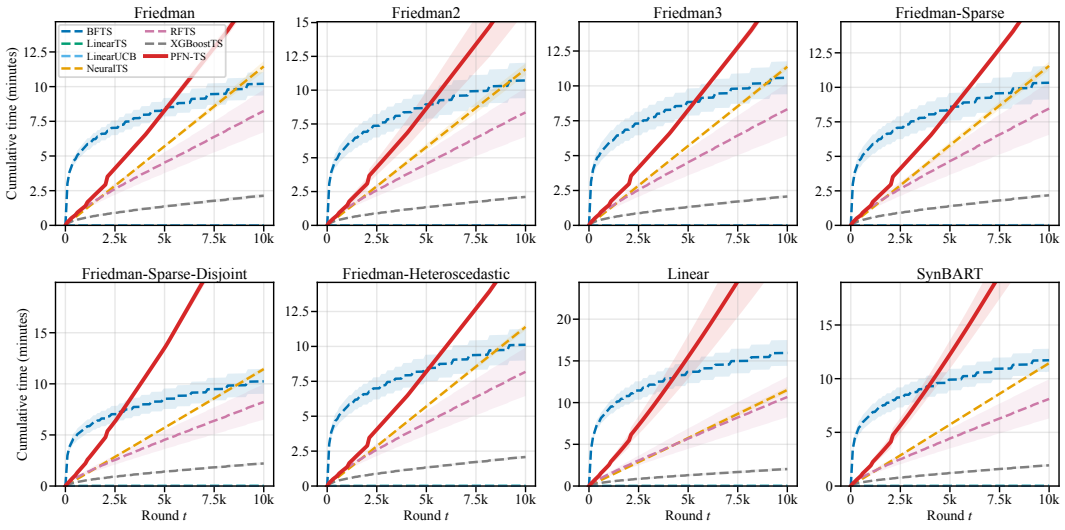

Figure 14: Cumulative wall-clock runtime on the eight synthetic scenarios by method (mean ± population SD).

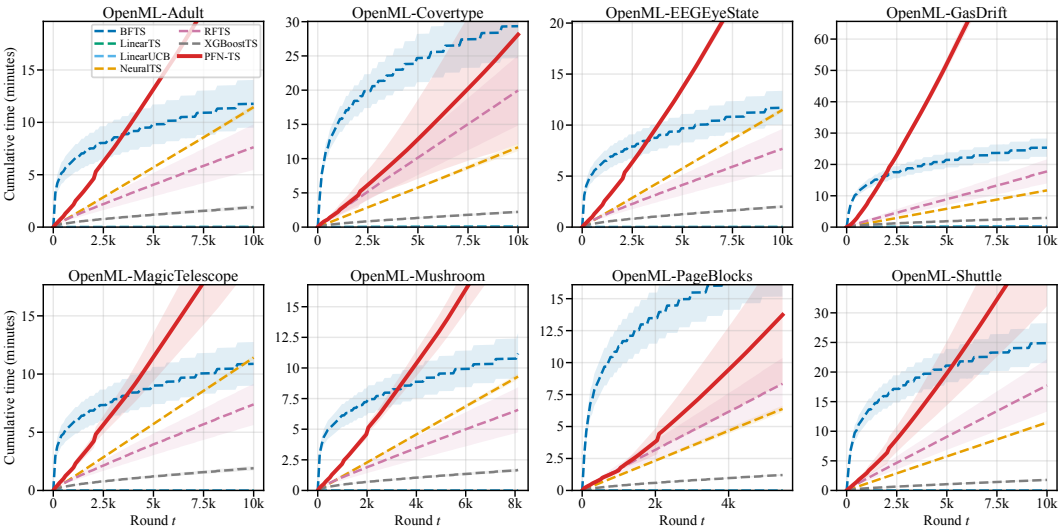

Figure 15: Cumulative wall-clock runtime on the eight OpenML datasets by method (mean ± population SD).