LEARNED WARMUP AND TARGET-CONDITIONED HANDOFF FOR CONTEXTUAL BANDITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many contextual bandit deployments are streams of short-lived instances: per title, campaign, participant, or item pool. Each instance starts with few local observations, so standard online algorithms pay the exploration cost repeatedly, even when aggregate traffic is large. We study prior-data fitted networks (PFNs) as a way to amortize this cold-start phase across related bandit tasks, deploying a model through standard forward passes conditioned on an instance's observed interaction prefix.

We instantiate this idea as BANDHA (Bandit Amortization with Neural Decision Handoff and Adaptation), using three mechanisms. First, the sampled PFN environment supplies full-feedback targets for all arms during training, while the model operates only on the censored feedback available online. Second, a permutation-equivariant per-arm state lets the learned update rule apply across different numbers of arms. Third, an outer Thompson-sampling controller selects the *action-generating policy* (AGP) used to produce informative training trajectories. Training uses teacher-augmented synthetic episodes, while deployment updates state only from censored feedback.

On fixed-dimensional contextual bandits, matched full-feedback supervision reduces regret at horizon 200 by 5.09 ± 1.35 points relative to censored supervision across three training runs. The learned update rule also transfers across arm counts and datasets. A single model trained on $K \in \{3, 5, 10\}$, where K is the number of arms, is evaluated at $K \in \{3, 5, 7, 10, 13, 15, 20\}$ using cumulative pseudo-regret. At horizon 200, the model reduces this regret by 4.2% to 33.6% relative to linear upper confidence bound (LinUCB), and improves over linear Thompson sampling (LinTS). At horizon 1000, it reduces regret by 21.6% to 54% relative to LinUCB in paired evaluations on the same instances at $K \in \{3, 5, 10\}$. On 17 real-world classification datasets, the same model remains useful zero-shot after fixed preprocessing. When another model, such as LinUCB, must take over, a warmup prefix optimized for its downstream continuation (target-conditioned handoff) improves on an untargeted prefix, while unconditioned or misaligned prefixes show where amortized exploration can hurt downstream learners.

1 INTRODUCTION

Contextual bandits are often deployed as a stream of related instances: new titles, campaigns, participants, or item pools. Each instance begins with few local observations and spends its early interactions identifying useful arms. Examples include artwork personalization per title Amat et al. (2018), finite advertising campaigns Iacob et al. (2022), mobile-health policies per participant Lei et al. (2017), and news recommenders over article pools that rotate continuously Li et al. (2010); Mary et al. (2014). Aggregate production traffic can be large, so previous instances ought to provide a warm start Tang et al. (2010). Direct reuse works when arm identities and feature definitions remain stable, but item pools rotate and feature pipelines change Li et al. (2010); Amat et al. (2018); Mary et al. (2014); Polyzotis et al. (2017); Agarwal et al. (2016). The bottleneck is not total traffic; it is transferring experience to a new instance whose local evidence is still sparse and whose earlier observations cannot be replayed directly.

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.

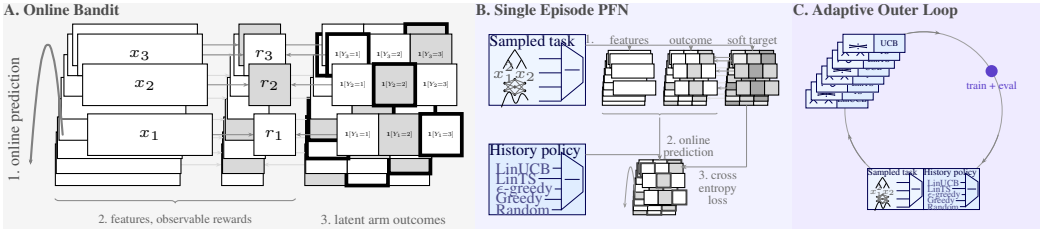


Figure 1: System overview. **A.** Y_t is the sampled winning arm; cell k shows $\mathbf{1}[Y_t = k]$. Deployment reveals only the outlined selected-arm reward $r_t = \mathbf{1}[Y_t = a_t]$. **B.** The sampled task supplies soft full-feedback targets through the fixed construction, while BANDHA still conditions only on censored feedback. **C.** A Thompson-sampling outer loop selects the AGP used to generate each epoch's training trajectories.

At round $t \in \{1, \dots, T\}$, a contextual-bandit learner observes context $x_t \in \mathcal{X} \subseteq \mathbb{R}^d$, where d is the context dimension, selects arm $a_t \in [K] := \{1, \dots, K\}$, and receives only that arm's binary reward r_t. We write θ for the latent environment, arm k's conditional mean reward as $p_\theta(k \mid x_t)$, and cumulative pseudo-regret as

$$R_T := \sum_{t=1}^{T} [\max_k p_\theta(k \mid x_t) - p_\theta(a_t \mid x_t)].$$

LinUCB Li et al. (2010), Thompson sampling Agrawal & Goyal (2013), NeuralUCB Zhou et al. (2020), NeuralTS Zhang et al. (2021), and deep Bayesian bandits Riquelme et al. (2018) all enter a fresh instance without local evidence. They must build task-specific statistics, representations, uncertainty estimates, or posterior samples from the same censored stream.

Our goal, therefore, is to learn a transferable rule that maps a short censored interaction prefix to action scores for a new bandit instance. Existing warm-start and safe bandit methods reduce cold-start cost using historical interaction data, validated synthetic priors, or a production baseline policy Oetomo et al. (2023); Cheung & Lyu (2024); Kazerouni et al. (2017). Meta-bandit methods transfer priors across related tasks when those tasks share a usable parameter space Kveton et al. (2021); Cella et al. (2020). We instead amortize this observation-to-action mapping before the new instance has built local evidence. Prior-data fitted networks (PFNs) are designed for this setting: sample many tasks from a prior, train a model to perform posterior inference from in-context observations, then replace per-instance fitting with a forward pass Müller et al. (2022); Hollmann et al. (2023). We adapt this to contextual-bandit warmup, where observations are censored by the learner's own actions and the number of arms may change across instances. A bandit-warmup PFN can act directly or hand off to a downstream bandit. In Appendix D.1, we prove by construction that two downstream learners can require different optimal warmup prefixes.

We introduce BANDHA, Bandit Amortization with Neural Decision Handoff and Adaptation, which uses three mechanisms. First, prior PFN work shows that latent-task supervision from a useful data-generating process (DGP) can support amortized inference. BANDHA adopts this premise, so its synthesized targets are useful when the DGP shares structure with deployment tasks. Online, the learner observes only the selected arm's binary reward, so a PFN trained naively on generated rewards would inherit sparse labels. We instead run each synthetic episode as a bandit while conditioning the model only on the censored interaction prefix available online. After the trajectory is generated, the teacher reveals the sampled environment's soft target probabilities for all arms. This produces soft full-feedback targets for the same censored interaction prefixes available at deployment. We instantiate and evaluate the construction on fixed-dimensional tabular contexts; we describe the corresponding construction for other PFN domains in Appendix J. Second, a *permutation-equivariant* per-arm state table avoids dependence on arm labels and lets one model operate across different K. Third, an outer Thompson-sampling controller chooses the *action-generating policy* (AGP) that produces each epoch's training trajectories, because random, greedy, and confidence-based AGPs expose different evidence.

Contributions

1. A censored-feedback PFN training scaffold in which synthetic teachers provide post-episode full-feedback targets, instantiated here with a soft loss.
2. A per-arm recurrent architecture with shared updates and set-based arm scoring, giving exact label equivariance and K-independent parameter shapes.
3. An outer Thompson-sampling controller for choosing the AGPs used to train the update rule, with ablations showing that strong online bandit policies do not automatically produce useful training trajectories.
4. Direct-policy evaluations against classical and neural bandit baselines across seen, held-out, and extrapolated arm counts; zero-shot real-data transfer after fixed preprocessing; and an analysis showing why optimal warmup prefixes depend on the downstream learner, with target-conditioned handoff experiments.

2 METHOD

Figure 1 summarizes the training pipeline: the sampled DGP, censored online trajectory, full-feedback teacher, and AGP selection loop. Figure 2 shows the per-arm recurrent state update and set-based scorer. We keep the implementation in the finite, discrete-arm tabular setting tested in this paper, while in Appendix J we give the corresponding PFN construction more abstractly.

Full-Feedback Supervision with Soft Targets The teacher separates the information available to the model from the information used to train its output: the model receives a censored interaction prefix, while the teacher supplies a target for all K arms. It samples a latent environment θ whose score function comes from a PFN-style prior mixture of formula trees, random multilayer perceptrons (MLPs), and Gaussian processes Müller et al. (2022). A regression task is converted to a K-armed bandit by binning the latent score into a soft target distribution $p_\theta(\cdot \mid x_t) \in \Delta^{K-1}$ via Gaussian CDF thresholding; here Δ^{K-1} is the probability simplex over K arms. The environment then samples $Y_t \sim \mathrm{Categorical}(p_\theta(\cdot \mid x_t))$ (details in Appendix I). Sampling θ lets fixed binning compute $p_\theta(\cdot \mid x_t)$ for all arms before censoring; this conditional mean reward vector is unavailable at deployment. In these experiments, the learner observes only the binary reward $r_t = \mathbf{1}[a_t = Y_t]$. Training is teacher-forced on AGP-generated censored trajectories rather than autoregressive self-rollouts. Panels A and B of Figure 1 make the information boundary explicit. The model-facing path is identical to deployment: only the context, chosen arm, and chosen-arm reward enter the recurrent state. The teacher-facing path in Panel B retains the synthetic full-feedback target vector $p_\theta(k \mid x_t)$ after the trajectory has been generated. This supplies counterfactual-arm targets without changing deployment information.

Let h_t be the observed prefix before round t, ϕ the learned parameters, and $q_{t,k} = q_\phi(k \mid h_t, x_t)$ the predicted probability of arm k. We sharpen the teacher target with γ=2 (Appendix B):

$$\widetilde{p}_{t,k} = \frac{p_\theta(k \mid x_t)^\gamma}{\sum_{j=1}^K p_\theta(j \mid x_t)^\gamma}. \tag{1}$$

At a sampled position, the soft cross-entropy and differentiable soft-target regret surrogate are

$$\ell_{\mathrm{CE},t} = -\sum_{k=1}^K \widetilde{p}_{t,k} \log q_{t,k}, \quad \ell_{\mathrm{reg},t} = \max_k \widetilde{p}_{t,k} - \sum_{k=1}^K q_{t,k}\widetilde{p}_{t,k}, \qquad \mathcal{L} = \mathbb{E}_{(\theta,h_t,x_t)\sim d_{\mathcal{P}}^{\beta}}[\ell_{\mathrm{CE},t} + \lambda \ell_{\mathrm{reg},t}], \tag{2}$$

where $\mathcal{L}$ is the training objective, λ=0.5 is the surrogate weight, and $d_{\mathcal{P}}^{\beta}$ is the distribution of retained positions in tasks from prior $\mathcal{P}$ and trajectories generated by AGP β. The conditional population minimizer of standard cross-entropy is the posterior mean reward vector. Sharpening targets the posterior mean of $\widetilde{p}$, but can change the greedy action after posterior averaging (Section 3). At deployment, the model receives only contexts, selected arms, and rewards.

Permutation-Equivariant Per-Arm State The architecture (Figure 2) maintains one state $S_{t,k} \in \mathbb{R}^D$ per arm, initialized identically at episode start, where D is the state dimension. A shared gated recurrent unit (GRU) applies event encoder ϕ_{event} to the previous context and reward; $\mathbf{1}$ marks the pulled arm:

$$S_{t,a_{t-1}} = \mathrm{GRU}\big(S_{t-1,a_{t-1}},\ \phi_{\mathrm{event}}(x_{t-1}, r_{t-1}, \mathbf{1})\big), \qquad S_{t,k} = S_{t-1,k} \text{ for } k \neq a_{t-1}. \tag{3}$$

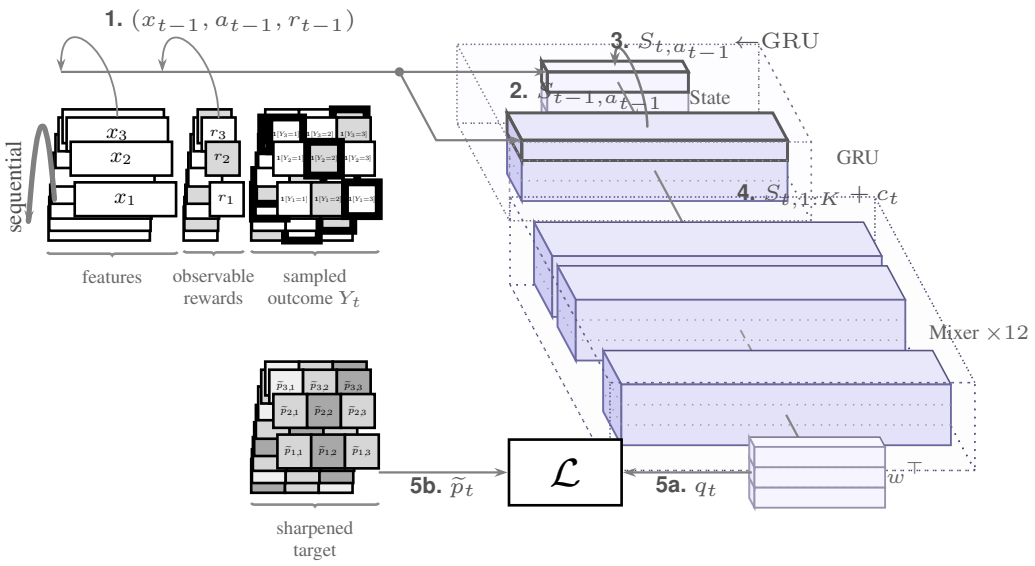


Figure 2: Per-arm state table architecture. **Left:** a bandit trajectory. Features x_t, selected arms a_t (outlined cells), and rewards r_t are observable online. Arm cell k shows $\mathbf{1}[Y_t = k]$; this full outcome vector and the sharpened soft target $\widetilde{p}_t$ are training-only. **Right:** at episode start, the state table is initialized with K identical rows, one per arm, where K is determined by the episode and requires no architectural change. (**1**) The previous observation $(x_{t-1}, a_{t-1}, r_{t-1})$ selects row a_{t-1}; the GRU receives event embedding $\phi_{\text{event}}(x_{t-1}, r_{t-1}, 1)$, where 1 marks the selected row. (**2**) The selected arm's state $S_{t-1,a_{t-1}}$ is read. (**3**) Only the pulled arm's state is updated via $S_{t,a_{t-1}} \leftarrow \text{GRU}(S_{t-1,a_{t-1}}, \phi_{\text{event}}(x_{t-1}, r_{t-1}, 1))$; unpulled arms retain their states. (**4**) All arm states plus the context embedding are read by the 12-layer Transformer arm mixer, and a shared linear head $w^\top$ produces per-arm logits. (**5a**) After the full episode, predicted probabilities q_t and (**5b**) training-only soft targets $\widetilde{p}_t$ enter the loss $\mathcal{L}$. Dotted partition lines show the K-arm broadcast structure: every component uses shared weights applied identically to each arm, making the architecture exactly permutation-equivariant over arm labels (Appendix E).

All arms are scored jointly by a 12-layer, 256-dimensional Transformer encoder without positional embeddings; ϕ_x is the context embedding, w the shared head vector, and $\ell_{t,k}$ the arm-k logit:

$$\ell_{t,k} = w^\top \text{ArmMixer}(S_{t,1}+c_t, \ldots, S_{t,K}+c_t)_k, \qquad c_t = \phi_x(x_t). \tag{4}$$

A chunked variant scores state snapshots in batches, reducing K=10 epoch time from 320s to 19s (16.8×). Permutation equivariance is exact by construction (Appendix E). This is how the model handles changing action sets: relabeling arms relabels outputs, and adding arms adds state rows rather than learned parameters. The same checkpoint can predict beyond its training horizon because prior observations are summarized in recurrent per-arm states rather than appended to the Transformer sequence.

Mixed-K Training To test generalization across arm counts, a single model is trained on a mixture of arm counts via round-robin alternation over $K \in \{3, 5, 10\}$. Because the shared components in Figure 2 do not depend on K, the same weights apply across arm counts without surgery. We evaluate the resulting model on held-out K=7 (interpolation) and K=13, 15, 20 (extrapolation).

Outer-Loop AGP Selection We implement AGP selection as a five-armed Thompson-sampling approximation to the stateful training problem. At each epoch, the controller chooses random, LinUCB, LinTS, ϵ-greedy, or a mixture. Random trajectories cover arms evenly; LinUCB and LinTS concentrate on arms that already look good under their own estimates; ϵ-greedy sits between those regimes.

After the epoch, the controller scores that choice by validation-regret improvement and updates its posterior over AGPs. Thus it is rewarded for producing useful training trajectories, not for low regret

by the AGP itself. The ablations in Appendix H match this distinction: in the same-initialization run, the adaptive controller outperforms the fixed random AGP, while pure LinTS and LinUCB produce less useful trajectories despite being strong online policies.

Target-Conditioned Handoff When downstream learner X takes over after H rounds, let s_t be its state at round t (distinct from BANDHA's per-arm states $S_{t,k}$), $R_J(X \mid s)$ its expected regret over a J-round tail from state s, and $U_X(s, x, a, r)$ its state after observing (x, a, r). Low prefix regret need not produce a useful interaction prefix, so the teacher values an observation by its reduction in X's expected tail regret:

$$\Psi_X(a \mid s, x) = R_J(X \mid s) - \mathbb{E}_{r|s,x,a}[R_J(X \mid U_X(s, x, a, r))]. \tag{5}$$

Here $\widehat{\Delta}_t(a)$ is estimated immediate pseudo-regret and $\alpha \geq 0$ its dimensionless tradeoff weight against downstream value. The handoff teacher scores each action by $-\widehat{\Delta}_t(a) + \alpha\widehat{\Psi}_X(a \mid s_t, x_t)$, with $\widehat{\Psi}_X$ estimated by tail rollouts that update X. Untargeted prefixes omit this term; targeted and cross-target prefixes use the eventual or another successor, respectively (Appendix D.1).

3 ANALYSIS

Let $\mathcal{S}$ denote a supervision scheme. For each interaction prefix and current context (h, x), let $m^{\mathcal{S}}(h, x)$ be the conditional population target of its cross-entropy term. Let $m(h, x) = \mathbb{E}[p_\theta(\cdot \mid x) \mid h, x]$ be the aligned posterior mean. For a probability vector p, define

$$[s_\gamma(p)]_k = \frac{p_k^\gamma}{\sum_j p_j^\gamma}.$$

The reported scheme uses γ=2, so $m^{\mathcal{S}}(h, x) = \mathbb{E}[s_2(p_\theta(\cdot \mid x)) \mid h, x]$. Write $m_t = m(h_t, x_t)$ and $m_t^{\mathcal{S}} = m^{\mathcal{S}}(h_t, x_t)$, and let $\hat{q}_t = q_\phi(\cdot \mid h_t, x_t)$ be the learned predictor. The deployed policy $\hat{\pi}$ selects $\arg\max_k \hat{q}_t(k)$, while π^{B} selects $\arg\max_k m_t(k)$ on the same interaction prefixes. Write d_t^β and $d_t^{\hat{\pi}}$ for the round-t distributions of (h_t, x_t) under the AGP and deployed policy. Let $R_H(\hat{\pi})$ be the expected cumulative pseudo-regret of the deployed policy over the first H rounds, and let $R_H(\pi^{\mathrm{B}})$ be the counterfactual regret from taking the posterior-mean action on those same prefixes. Define pointwise target error $e_{\mathcal{S}}(h, x) = \|q_\phi(\cdot \mid h, x) - m^{\mathcal{S}}(h, x)\|_\infty$ and

$$\bar{b}_{\mathcal{S}} = \frac{1}{H}\sum_{t=1}^{H} \mathbb{E}_{d_t^{\hat{\pi}}}[\|m_t^{\mathcal{S}} - m_t\|_\infty], \tag{6}$$

$$\bar{s}_{\mathcal{S}} = \frac{1}{H}\sum_{t=1}^{H} \left[\mathbb{E}_{d_t^{\hat{\pi}}} e_{\mathcal{S}} - \mathbb{E}_{d_t^{\beta}} e_{\mathcal{S}}\right]_+. \tag{7}$$

Here $[u]_+ = \max(u, 0)$. If the predictor has average Kullback-Leibler (KL) fit level $\epsilon_{\mathcal{S}}$, meaning $H^{-1}\sum_{t=1}^{H} \mathbb{E}_{d_t^\beta}\mathrm{KL}(m_t^{\mathcal{S}} \| \hat{q}_t) \leq \epsilon_{\mathcal{S}}^2$ on AGP-generated trajectories, then the following bound holds (Appendix B, Theorem 1):

$$R_H(\hat{\pi}) \leq R_H(\pi^{\mathrm{B}}) + 2\sqrt{2}\, H\epsilon_{\mathcal{S}} + 2H\bar{b}_{\mathcal{S}} + 2H\bar{s}_{\mathcal{S}}. \tag{8}$$

The last three terms separate KL fit, target bias, and pointwise-target-error shift. The first term is the regret of the posterior-mean action on prefixes induced by $\hat{\pi}$; it can remain even when the other three terms vanish. Supervision changes target bias and KL fit, shared estimation acts on KL fit across arm counts, and horizon behavior depends on whether the resulting decision errors persist.

Synthesized Full-Feedback Supervision. As stated in Section 2, during a training episode the selected AGP chooses a_t at step t, and the online interaction prefix contains only the bit $r_t = \mathbf{1}[a_t = Y_t]$. The teacher still has the sampled outcome Y_t, so it can write down the reward vector for every arm. BANDHA does not feed that vector back into the recurrent state; it uses it only as the supervised target. The state seen by the model is still the censored prefix h_t.

For K>2, full feedback resolves ambiguity left by a failed pull: the pull says the chosen arm was wrong but not which other arm was right. In Appendix A.2, we show the corresponding local Fisher

loss under a uniform AGP while predictions are diffuse. The positive-only censored baseline is an even weaker training signal: under a uniform AGP, a $1 - 1/K$ fraction of positions have zero gradient, and with a nonuniform AGP the target is tilted by the AGP propensity (Appendix A.1).

The actual teacher supplies the conditional mean of such full-feedback outcomes. It retains $p_\theta(\cdot \mid x)$ from Eq. (1). Fix (θ, x) and draw n independent outcomes

$$Y^{(j)} \overset{\text{iid}}{\sim} \text{Categorical}(p_\theta(\cdot \mid x)). \tag{9}$$

With e_y the one-hot encoding of outcome y, their average satisfies

$$\frac{1}{n}\sum_{j=1}^{n} e_{Y^{(j)}} \longrightarrow \mathbb{E}[e_Y \mid \theta, x] = p_\theta(\cdot \mid x). \tag{10}$$

Because cross-entropy is linear in the label, the soft update is the limit of averaging sampled full-feedback one-hot updates. It is also the Rao-Blackwellized update: it keeps the expected gradient and removes the conditional covariance $\text{diag}(p_\theta) - p_\theta p_\theta^\top$. The trace $1 - \|p_\theta\|_2^2$ is 0.314 for formula tasks, 0.210 for GPs, and 0.205 for MLPs on the held-out $K{=}5$ suite (Table 11). After averaging over the posterior on θ, the same construction gives m_t, whose greedy action is the one-step posterior-regret minimizer.

Equation (8) separates these mechanisms. Minimizing the expected naive censored loss gives

$$m_k^{\mathcal{S}}(h, x) = \frac{\beta(k \mid h, x) m_k(h, x)}{\sum_j \beta(j \mid h, x) m_j(h, x)}.$$

This produces target bias for a nonuniform AGP; for the uniform AGP used in the matched comparison this bias is zero. Soft and sampled one-hot full feedback both have population minimizer m, so on common AGP trajectories their difference can appear only through $\epsilon_{\mathcal{S}}$, the KL-fit term. The sharpened loss instead changes both $\bar{b}_{\mathcal{S}}$ and potentially $\epsilon_{\mathcal{S}}$. The trained system also uses a regret surrogate and masks early positions, so its $\epsilon_{\mathcal{S}}$ must be measured rather than inferred from the population results (Appendix B). Across three training runs, matched full feedback reduces $T{=}200$ regret by 5.09 ± 1.35 relative to censored feedback (Appendix H.1).

Arm Counts. The supervision scheme fixes the target and training signal; the architecture fixes the class in which that target is approximated. Both affect the KL-fit level $\epsilon_{\mathcal{S}}$ in Eq. (8). For a permutation σ of the K arms, let σh_t denote the interaction prefix obtained by relabeling every action and arm state, and for $v \in \mathbb{R}^K$ define $[\sigma v]_k = v_{\sigma^{-1}(k)}$.

Shared estimation. The synthetic prior is exchangeable in the arm labels, and the architecture uses the same update rule for every arm. Together these give

$$m(\sigma h_t, x_t) = \sigma m(h_t, x_t), \qquad q_\phi(\cdot \mid \sigma h_t, x_t) = \sigma q_\phi(\cdot \mid h_t, x_t). \tag{11}$$

Thus the target and model obey the same relabeling symmetry. For the reported $D{=}256, d{=}3$ model, the shared state-update block has $P_{\text{upd}} = 7D^2 + Dd + 13D = 462{,}848$ parameters. Making one copy per arm would add $(K-1)P_{\text{upd}}$, or about 0.93M, 1.85M, 4.17M, and 8.79M parameters at $K{=}3, 5, 10, 20$. It would also split the data. Under uniform sampling, a slot-specific updater sees only T/K transitions per episode; the shared updater sees all T. At the trained arm counts, these symmetry and sharing effects appear in the KL-fit term $\epsilon_{\mathcal{S}}$ in Eq. (8).

Changing the set size. Figure 2 makes the cross-K factorization explicit: steps 1 to 3 encode each arm with the same K-independent recurrent update, while step 4 compares the resulting arm states with a normalized set mixer,

$$q_\phi(\cdot \mid h_t, x_t) = \text{Mix}_K\big(\text{Enc}(h_t^{(1)}), \ldots, \text{Enc}(h_t^{(K)}); x_t\big), \tag{12}$$

where $h_t^{(k)}$ is arm k's selected-feedback subhistory and Enc is the shared recurrent encoder that maps this subhistory to an arm state. Thus only the comparison stage changes with K. For one attention head, let v_j be arm j's value vector, z_{kj} its score from arm k, $\omega_{kj}^{(K)} = \text{softmax}_j(z_{k1:K})$, and $o_k^{(K)} = \sum_j \omega_{kj}^{(K)} v_j$. If $\|v_j\| \le B$ and $|z_{kj}| \le \Lambda$, adding one arm gives

$$o_k^{(K+1)} - o_k^{(K)} = \omega_{k,K+1}^{(K+1)}(v_{K+1} - o_k^{(K)}), \tag{13}$$

$$\|o_k^{(K+1)} - o_k^{(K)}\| \le \frac{2Be^{2\Lambda}}{e^{2\Lambda} + K} \tag{14}$$

Table 1: Cumulative regret at T=200 for one 12-layer model with 256-dimensional hidden states trained on $K \in \{3, 5, 10\}$. † marks unseen arm counts. Learned-policy entries are means ± standard errors over 100 environments and 10 rollout seeds (n=1000). The uniform mean is exact per environment; its standard error is across the 100 environments. The last column gives percentage regret reduction and paired win rate against LinUCB.

K	BANDHA	LinUCB	LinTS	Uniform	Reduction / win vs. LinUCB (%)
3	**37.7±0.8**	56.8±0.7	65.8±0.7	102.6±1.5	33.6 / 92
5	**68.9±1.0**	90.8±0.6	105.4±0.5	124.2±1.3	24.1 / 87
7†	**92.3±0.9**	111.4±0.6	124.5±0.5	134.2±1.3	17.1 / 81
10	**115.7±0.8**	129.3±0.5	139.1±0.4	143.3±1.2	10.5 / 79
13†	**124.3±0.7**	137.0±0.5	142.2±0.4	144.4±1.3	9.3 / 88
15†	**130.9±0.6**	139.9±0.5	144.8±0.4	146.4±1.2	6.4 / 84
20†	**140.8±0.7**	147.0±0.5	149.8±0.4	150.5±1.2	4.2 / 71

The bound decreases strictly with K; Appendix E gives the accumulated bound for any $K' > K$. Figure 2 therefore combines exact reuse of the learned per-arm encoder with a bounded, decaying change in the comparison stage. The held-out counts in Table 1 follow the same smooth regret trend as the trained counts.

Long-Horizon Deployment and Handoff. The recurrent architecture in Section 2 can continue operating beyond its training horizon. Equation (8) bounds regret at each finite H. To characterize its long-horizon behavior, we ask whether the actual per-round decision error vanishes.

Let $\hat{a}_t$ be the action selected at round t, define its pseudo-regret $\delta_t = \max_k p_\theta(k \mid x_t) - p_\theta(\hat{a}_t \mid x_t)$, and write $p_t = \Pr(\delta_t > 0)$ and $g_t = \mathbb{E}[\delta_t \mid \delta_t > 0]$. Then

$$\mathbb{E}[\delta_t] = p_t g_t, \qquad R_H(\hat{\pi}) = \sum_{t=1}^{H} p_t g_t. \tag{15}$$

If $p_t \geq p > 0$ and $g_t \geq g > 0$, then $R_H(\hat{\pi}) \geq Hpg$. Against a comparator $\mathcal{A}$ with a valid bound $R_H(\mathcal{A}) \leq CK\Delta_{\text{gap}}^{-1} \log H$, where $C > 0$ is the bound constant and $\Delta_{\text{gap}} > 0$ is the comparator's reward gap, the sufficient crossover obeys

$$H^* = \Theta(\Xi \log \Xi) \quad \text{as } \Xi \to \infty, \qquad \Xi = \frac{CK}{pg\Delta_{\text{gap}}}. \tag{16}$$

The crossover therefore moves later when mistakes become rarer or less costly. Handoff is the asymptotic remedy only in this persistent-error regime. In the nonlinear synthetic suites tested here, we observe no crossover through T=1000 (Table 2); we give the full result in Appendix C.

4 EXPERIMENTAL RESULTS

We test one fixed checkpoint across arm counts, horizons, and real classification streams, then evaluate target-conditioned handoff. A controlled ablation varies only the AGP used to generate training trajectories.

Evaluation Protocol Online experiments report $R_T = \sum_{t=1}^{T} [\max_k p_\theta(k \mid x_t) - p_\theta(a_t \mid x_t)]$. Methods are paired on the same tasks, contexts, and reward draws. We report mean R_T, its standard error, and the fraction of paired rollouts won by BANDHA. We call validation regret divided by uniform-policy regret on the same held-out trajectories the *validation norm*; 1 is random-level and lower is better. Appendix L specifies all baselines and protocols.

Arm-Count Transfer We hold the model fixed, vary K, and measure online R_{200}. Training uses $K \in \{3, 5, 10\}$; evaluation at $K \in \{7, 13, 15, 20\}$ measures transfer to arm counts absent from training.

BANDHA has the lowest mean cumulative regret at every evaluated arm count. Its relative regret reduction versus LinUCB decreases from 33.6% at K=3 to 4.2% at K=20. The held-out counts

Table 2: Cumulative regret at $T=1000$. Methods are paired on the same rollouts. The $K=3, 5$ results use between 100 and 200 environments and 3 rollout seeds; the $K=10$ result uses 600 environments and 2 rollout seeds.

K	BANDHA	LinUCB	Reduction
3	**127.9**	280.9	54%
5	**206.3**	386.2	47%
10	**460.0**	586.4	21.6%

Table 3: Selected zero-shot results from 17 classification datasets ($T=200$, 30 shuffles). Fixed standardization and principal-component analysis (PCA) map each dataset to context dimension $d_{\text{ctx}}=3$; the model is not fine-tuned. Full results appear in Appendix G.

Dataset	K	BANDHA	LinUCB	LinTS	Win vs. LinTS
Iris	3	**19.7**	40.0	55.0	100%
Page-blocks	5	**24.2**	79.9	102.0	100%
Shuttle	7	**41.3**	72.0	112.6	100%
Segment	7	**90.4**	116.5	143.2	100%
Covertype	7	**125.0**	139.1	157.3	97%
Texture	11	**135.9**	146.8	172.6	100%
Volkert	10	145.6	**144.3**	171.5	97%
Letter	26	**187.7**	189.8	192.0	70%
Vehicle	4	122.3	**120.7**	134.3	87%
15/17 lower mean regret vs. LinUCB; 17/17 vs. LinTS.					

$K=7, 13, 15, 20$ follow the same smooth trend as the trained counts, supporting transfer to arm counts absent from training. The one-run fixed-K comparison in Table 13 is consistent with that picture: the mixed-K checkpoint has 18%, 24%, and 35% lower validation norm at $K=3, 5, 10$. At $K=20$, uniform regret is only 6.4% above BANDHA, and LinTS is only 0.5% below uniform, leaving less room for improvement than at smaller arm counts.

Long-Horizon Evaluation The checkpoint was trained on 200-round episodes. We then run it for 1000 rounds against LinUCB, using the same synthetic prior family.

At 1000 rounds, regret remains 54% below LinUCB at $K=3$, 47% below at $K=5$, and 21.6% below at $K=10$ (Table 2). The late-window slopes in Appendix C.1 are mixed, so this result establishes the advantage through the tested horizon rather than asymptotically.

Real-Data Transfer Class labels become arms and a correct choice receives reward one, so regret counts mistakes over 200 examples. Results average 30 shuffled streams after the fixed feature mapping.

The zero-shot transfer is strongest on small and medium label sets. Out of 17 datasets, the synthetic-trained model has lower mean regret than LinUCB on 15 and lower mean regret than LinTS on all 17 (Table 3). Every dataset is first standardized and projected to $d_{\text{ctx}}=3$, so this is a test of the update rule after a fixed feature map, not a learned representation interface. The misses are informative: LinUCB wins on `volkert` and `vehicle`, and on `letter` the gap is only 2.1 mistakes.

Target-Conditioned Handoff In the handoff experiment, LinUCB is not cold-started. It inherits the prefix log produced by BANDHA and then runs until $T=1000$. We compare prefixes trained without a successor target against prefixes trained to reduce a successor's tail regret. The Greedy-target rows are a mismatch test: they are trained for a greedy successor but handed to LinUCB.

At $H=400$, the best targeted prefix cuts regret by 10.964 ± 0.412 relative to the untargeted prefix and by 40.858 ± 0.676 relative to LinUCB alone (Table 4). One useful surprise is that the Greedy-targeted prefix is 1.954 ± 0.386 better than the matched LinUCB target. Target awareness matters, but the exact successor model is not the only way to produce a useful interaction prefix. We show

Table 4: LinUCB handoff at $K=10$, $T=1000$ (2000 environments, 2 rollout seeds; $n=4000$). All learned rows use a BANDHA prefix and the same downstream LinUCB evaluator. Lower regret is better.

Prefix objective	H	Regret	Win vs. LinUCB
LinUCB alone	none	586.989 ± 1.417	0.0%
Untargeted	400	557.095 ± 1.574	77.4%
Greedy target, $\alpha=0.5$	400	546.413 ± 1.570	85.7%
Greedy target, $\alpha=1$	400	$\mathbf{546.131 \pm 1.571}$	86.1%
LinUCB target, $\alpha=0.25$	400	548.086 ± 1.588	85.8%
Untargeted	200	577.358 ± 1.447	64.0%
Greedy target, $\alpha=0.5$	200	572.065 ± 1.456	69.8%

in Appendix D that, once the successor has enough coverage, additional exploration increases prefix regret without reducing tail regret.

AGP-Selection Ablation This ablation changes only the trajectories used for training. With one initialization at $K=5$, the outer controller improves validation norm from 0.400 for random trajectories to 0.344. Fixed LinTS and LinUCB trajectories both land at 0.491, even though those policies are stronger online than random sampling. The mixed-K adaptive run reaches 0.280, but it changes the arm-count schedule as well, so we treat it as a separate reference rather than the controlled comparison. Because the controlled run uses one initialization and the gap is on the scale of cross-run variation, we use it as directional evidence. The contrast shows that a policy's online regret does not determine the value of the trajectories it generates for training (Appendix H.8).

5 DISCUSSION

Related Work PFNs amortize inference over synthetic task priors Müller et al. (2022); Hollmann et al. (2023). The difference here is that the in-context observations are chosen by the learner and then censored. Related in-context decision models learn from trajectories by imitating actions or by conditioning on returns Lee et al. (2023); Laskin et al. (2023); Chen et al. (2021); Sinii et al. (2024). Here the model predicts arm reward distributions instead. NeuralUCB and NeuralTS fit inside each episode Zhou et al. (2020); Zhang et al. (2021); meta-bandits and warm-start methods transfer priors, interaction data, or baselines Riquelme et al. (2018); Kveton et al. (2021); Cella et al. (2020); Oetomo et al. (2023); Cheung & Lyu (2024); Kazerouni et al. (2017). BANDHA transfers a different object: an update rule for a short censored interaction prefix.

Limitations PFNs rely on a synthetic task prior that captures structure relevant to the deployment tasks Müller et al. (2022); Hollmann et al. (2023). BANDHA inherits this dependence and also uses the generator's full-feedback probabilities as training labels. If the DGP is misspecified, those labels can teach a confident but systematically wrong update rule. The real-data results test transfer beyond the synthetic suites, but do not certify that the prior is appropriate for a new deployment.

Our experiments are limited to fixed-dimensional tabular contexts. The real datasets use fixed standardization and PCA to $d_{\mathrm{ctx}}=3$, and we have not trained or evaluated the non-tabular extensions outlined in Appendix J. The controlled AGP comparison also uses one training initialization, while the $T=1000$ experiments establish a finite horizon advantage rather than a regret certificate.

Conclusion BANDHA amortizes early learning across related contextual-bandit instances by training a recurrent, permutation-equivariant update rule on censored trajectories with synthetic full feedback. One checkpoint reduces $T=200$ pseudo-regret relative to LinUCB by 4.2% to 33.6% across seven arm counts, including four absent from training, and remains 21.6% to 54% lower than LinUCB at $T=1000$. It also transfers without fine-tuning to 17 real classification datasets, with lower mean regret than LinUCB on 15 and LinTS on all 17. When LinUCB takes over, target-conditioned prefixes further reduce total regret. Together, these results support amortized inference for cold start, followed by handoff when a deployment requires a separately certified long-horizon learner.

Ethics Statement

This work trains on synthetic contextual bandits and evaluates on public tabular benchmarks. Potential downstream risks arise if amortized bandit models are deployed in high-stakes domains without uncertainty calibration or fallback policies.

Reproducibility Statement

The supplement contains the data-generation procedure (Appendix I), training losses and model sizes (Appendix K), AGP ablations (Appendix H), and evaluation protocol (Appendix L). An anonymized repository containing code and result summaries is available at `https://anonymous.4open.science/r/bandha-4D3D/README.md`.

AI Use Statement

We used generative AI tools during manuscript preparation to review and edit prose, check mathematical and internal consistency, and suggest corrections. We reviewed and verified all AI-assisted work, including the resulting text, mathematical claims, citations, and artifacts, and take responsibility for the final content of this submission.

References

Alekh Agarwal, Sarah Bird, Markus Cozowicz, Luong Hoang, John Langford, Stephen Lee, Jiaji Li, Dan Melamed, Gal Oshri, Oswaldo Ribas, Siddhartha Sen, and Alex Slivkins. Making contextual decisions with low technical debt. arXiv:1606.03966, 2016. URL https://arxiv.org/abs/1606.03966.

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

Fernando Amat, Ashok Chandrashekar, Tony Jebara, and Justin Basilico. Artwork personalization at Netflix. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 487–488, 2018. doi: 10.1145/3240323.3241729.

Leonardo Cella, Alessandro Lazaric, and Massimiliano Pontil. Meta-learning with stochastic linear bandits. In *International Conference on Machine Learning*, 2020.

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

Wang Chi Cheung and Lixing Lyu. Online bandits with (biased) offline data: Adaptive learning under distribution mismatch. arXiv:2405.02594, 2024. URL https://arxiv.org/abs/2405.02594.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=cp5PvcI6w8_.

Alexandra Iacob, Bogdan Cautis, and Silviu Maniu. Contextual bandits for advertising campaigns: A diffusion-model independent approach. In *SIAM International Conference on Data Mining*, 2022. URL https://arxiv.org/abs/2201.05231.

Abbas Kazerouni, Mohammad Ghavamzadeh, Yasin Abbasi-Yadkori, and Benjamin Van Roy. Conservative contextual linear bandits. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1611.06426.

Branislav Kveton, Mikhail Konobeev, Manzil Zaheer, Chih-Wei Hsu, Martin Mladenov, Craig Boutilier, and Csaba Szepesvari. Meta-thompson sampling. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 5884–5893. PMLR, 2021. URL https://proceedings.mlr.press/v139/kveton21a.html.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, D. J. 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=VwN8wNNv4c.

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*, 2023.

Huitian Lei, Yangyi Lu, Ambuj Tewari, and Susan A. Murphy. An actor-critic contextual bandit algorithm for personalized mobile health interventions. arXiv:1706.09090, 2017. URL https://arxiv.org/abs/1706.09090.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *International Conference on World Wide Web*, 2010.

Jérémie Mary, Romaric Gaudel, and Philippe Preux. Bandits warm-up cold recommender systems. arXiv:1407.2806, 2014. URL https://arxiv.org/abs/1407.2806.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

Bastian Oetomo, R. Malinga Perera, Renata Borovica-Gajic, and Benjamin I. P. Rubinstein. Cutting to the chase with warm-start contextual bandits. *Knowledge and Information Systems*, 65(9): 3533–3565, 2023. doi: 10.1007/s10115-023-01861-2.

Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang, and Martin Zinkevich. Data management challenges in production machine learning. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 1723–1726, 2017. doi: 10.1145/3035918.3054782.

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.

Viacheslav Sinii, Alexander Nikulin, Vladislav Kurenkov, Ilya Zisman, and Sergey Kolesnikov. In-context reinforcement learning for variable action spaces. In *International Conference on Machine Learning*, pp. 45773–45793, 2024. URL https://proceedings.mlr.press/v235/sinii24a.html.

Diane Tang, Ashish Agarwal, Deirdre O'Brien, and Mike Meyer. Overlapping experiment infrastructure: More, better, faster experimentation. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 17–26, 2010. doi: 10.1145/1835804.1835810.

Weitong Zhang, Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural thompson sampling. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tkAtoZkcUnm.

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with UCB-based exploration. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 11492–11502. PMLR, 2020. URL https://proceedings.mlr.press/v119/zhou20a.html.

A Full-Feedback Supervision

An online bandit observes only the reward of the selected arm. During synthetic training, an AGP generates the same censored interaction prefix, but the teacher also exposes the sampled task's full-feedback conditional reward vector after generating that prefix. We derive the resulting posterior target and formalize the variance reduction from soft supervision.

A.1 Posterior Target

Let $\theta \sim \mathcal{P}$ denote a latent environment. At round t, the learner observes $x_t \in \mathbb{R}^d$ and the censored interaction prefix

$$h_t = (x_1, a_1, r_1, \ldots, x_{t-1}, a_{t-1}, r_{t-1}). \tag{17}$$

The generator supplies $p_\theta(\cdot \mid x_t) \in \Delta^{K-1}$ and samples $Y_t \sim \mathrm{Categorical}(p_\theta(\cdot \mid x_t))$. The observed reward is $r_t = \mathbf{1}[a_t = Y_t]$, hence

$$\mathbb{E}[r_t \mid \theta, x_t, a_t = k] = p_\theta(k \mid x_t). \tag{18}$$

The model outputs $q_\phi(\cdot \mid h_t, x_t) \in \Delta^{K-1}$.

Proposition 1 (Soft-label posterior recovery)**.** *For every fixed (h_t, x_t), the conditional population minimizer of the unsharpened soft cross-entropy risk is*

$$m_t = \mathbb{E}[p_\theta(\cdot \mid x_t) \mid h_t, x_t]. \tag{19}$$

The greedy action $\arg\max_k m_t(k)$ minimizes one-step posterior expected pseudo-regret.

Proof. Fix (h_t, x_t). Averaging the soft cross-entropy over the posterior gives $-\sum_k m_t(k) \log q(k)$, which is minimized at $q = m_t$. Moreover, for any action a,

$$\mathbb{E}[\max_k p_\theta(k \mid x_t) - p_\theta(a \mid x_t) \mid h_t, x_t] = \mathbb{E}[\max_k p_\theta(k \mid x_t) \mid h_t, x_t] - m_t(a). \tag{20}$$

The first term is action-independent, so maximizing $m_t(a)$ minimizes the conditional pseudo-regret. □

A.2 Information Lost to Censoring

The previous proposition gives the population minimizer of full-feedback cross-entropy. We next compare the information in a full sampled outcome with the censored observation available to an online bandit.

Proposition 2 (Censored feedback is a coarsening)**.** *Fix (h_t, x_t), draw $A_t \sim \beta(\cdot \mid h_t, x_t)$ independently of θ conditional on (h_t, x_t), and let $R_t = \mathbf{1}[A_t = Y_t]$. Then $\theta \to (A_t, Y_t) \to (A_t, R_t)$ is a Markov chain and*

$$I(\theta; A_t, R_t \mid h_t, x_t) \le I(\theta; A_t, Y_t \mid h_t, x_t). \tag{21}$$

The two observations are equivalent when K=2. For $K > 2$, the inequality can be strict.

Proof. The reward R_t is a deterministic function of (A_t, Y_t), so the conditional data-processing inequality gives the result. When K=2, (A_t, R_t) also determines Y_t, making the two observations equivalent. When $K > 2$, $R_t = 0$ pools at least two possible values of Y_t; any prior that distinguishes their probabilities gives a strict instance. □

The proposition concerns the observation channel and requires no parametric model. A local calculation describes its rate near a categorical model.

Proposition 3 (Local information rate)**.** *Let $q = \mathrm{softmax}(z)$ and $v_a = e_a - q$. Let $\mathcal{I}_{\mathrm{full}}$ be the Fisher information about z in $Y \sim \mathrm{Categorical}(q)$. Let $\mathcal{I}_{\mathrm{cens}}$ be the information in (A, R) when $A \sim \beta$ independently of z and $R = \mathbf{1}[A = Y]$. Then*

$$\mathcal{I}_{\mathrm{full}} = \sum_{a=1}^{K} q_a v_a v_a^\top = \mathrm{diag}(q) - qq^\top, \tag{22}$$

$$\mathcal{I}_{\mathrm{cens}} = \sum_{a=1}^{K} \beta_a \frac{q_a}{1 - q_a} v_a v_a^\top, \tag{23}$$

and

$$\beta_{\min}\mathcal{I}_{\text{full}} \preceq \mathcal{I}_{\text{cens}} \preceq \frac{\beta_{\max}}{1-q_{\max}}\mathcal{I}_{\text{full}}. \tag{24}$$

Under a uniform AGP and $q_{\max} \le 1 - c$ for fixed $c > 0$, censored feedback therefore carries a factor of order K less local information.

Proof. The categorical score is $e_Y - q$, giving $\mathcal{I}_{\text{full}} = \mathrm{Cov}(e_Y)$. Conditional on $A = a$, R is Bernoulli(q_a) and $\nabla_z q_a = q_a v_a$, so its information is $q_a(1-q_a)^{-1}v_a v_a^\top$. Averaging over A yields the second identity. The matrix bounds follow termwise from $\beta_{\min} \le \beta_a \le \beta_{\max}$ and $1 \le (1 - q_a)^{-1} \le (1 - q_{\max})^{-1}$. □

Proposition 3 is local and does not by itself imply a finite-sample regret or neural-optimization bound. The censored objective used in our matched comparison discards still more information: under a uniform AGP, $\Pr(R = 1) = 1/K$, so the positive-only loss $-R \log q_A$ has zero gradient on a $1 - 1/K$ fraction of positions. This is a property of that estimator, not of censoring itself. The $K{=}2$ boundary in Proposition 2 separates the two effects: the channel loses no information there, while the positive-only estimator still discards negative observations.

Proposition 4 (Rao-Blackwellized supervision)**.** *Fix (θ, x), write $p = p_\theta(\cdot \mid x)$, let $Y \sim \mathrm{Categorical}(p)$, and let $q = \mathrm{softmax}(z)$. The sampled one-hot loss and the soft full-feedback loss satisfy*

$$\mathbb{E}[-\log q_Y \mid \theta, x] = -\sum_{k=1}^{K} p_k \log q_k, \tag{25}$$

$$\mathbb{E}[\nabla_z(-\log q_Y) \mid \theta, x] = q - p, \tag{26}$$

$$\mathrm{Cov}(\nabla_z(-\log q_Y) \mid \theta, x) = \mathrm{diag}(p) - pp^\top, \tag{27}$$

$$\mathrm{tr}\,\mathrm{Cov}(\nabla_z(-\log q_Y) \mid \theta, x) = 1 - \|p\|_2^2. \tag{28}$$

The soft-label gradient is exactly $q - p$ and has zero conditional outcome variance.

Proof. For a one-hot vector e_Y, the hard-label gradient is $q - e_Y$ and $\mathbb{E}[e_Y \mid \theta, x] = p$. The displayed loss and gradient identities follow by conditioning on Y; the covariance is the covariance of a categorical one-hot vector. □

Thus one soft label removes exactly the outcome noise left after replacing the teacher distribution by a sampled one-hot label. The removed total gradient variance $1 - \|p\|_2^2$ vanishes as the teacher becomes one-hot and is largest at the uniform distribution. This quantity, rather than Shannon entropy, is the exact prior-family diagnostic implied by the proposition.

This proposition isolates softness from full-feedback access. Suppose an AGP draws $A \sim \beta(\cdot \mid h, x)$ and observes only $R = \mathbf{1}[A = Y]$. The naive positive-only categorical loss has conditional risk

$$\mathcal{L}_{\text{obs}}(q; h, x) = -\sum_{k=1}^{K} \beta(k \mid h, x) m_t(k) \log q_k, \tag{29}$$

and therefore has population minimizer

$$q_k^{\text{obs}} = \frac{\beta(k \mid h, x) m_t(k)}{\sum_j \beta(j \mid h, x) m_t(j)}. \tag{30}$$

The AGP can consequently change the greedy action. For example, $m_t = (0.4, 0.6)$ and $\beta = (0.9, 0.1)$ give $q^{\text{obs}} = (0.857, 0.143)$. This is an AGP-induced target tilt, not causal confounding: conditional on (h, x), the policy's action is ignorable.

If $0 < \beta_{\min} \le \beta_k \le \beta_{\max}$, let $a^{\text{B}} = \arg\max_k m_t(k)$ and $a^{\text{obs}} = \arg\max_k \beta_k m_t(k)$. Optimality of a^{obs} for the tilted score implies

$$m_t(a^{\text{obs}}) \ge \frac{\beta_{a^{\text{B}}}}{\beta_{a^{\text{obs}}}} m_t(a^{\text{B}}) \ge \frac{\beta_{\min}}{\beta_{\max}} m_t(a^{\text{B}}), \tag{31}$$

and hence

$$m_t(a^{\mathrm{B}}) - m_t(a^{\mathrm{obs}}) \le 1 - \frac{\beta_{\min}}{\beta_{\max}}. \tag{32}$$

The bound vanishes for a uniform AGP. More generally, the full-feedback teacher makes m_t independent of the current AGP propensities, leaving β to control the distribution of prefix-context pairs. This makes AGP selection a design over trajectories for a fixed estimand; under the naive loss it would also move the estimand itself.

With full support, the inverse-propensity loss

$$-\frac{R}{\beta(A \mid h, x)} \log q_A \tag{33}$$

is an unbiased estimator of the aligned soft cross-entropy. Unlike direct access to p, it is censored and carries action-sampling and outcome noise that grows at small propensities. Thus the synthetic teacher removes β from the conditional target while the AGP still controls which prefix-context pairs enter the training distribution. The matched ablation separates these mechanisms by comparing soft full feedback, sampled one-hot full feedback, and naive censored supervision.

For the sharpened target $\widetilde{p}_{t,k} \propto p_\theta(k \mid x_t)^\gamma$, the conditional minimizer of sharpened cross-entropy is

$$\mathbb{E}[\widetilde{p}_t \mid h_t, x_t]. \tag{34}$$

At $\gamma = 1$, this equals m_t exactly; for other values, posterior averaging and sharpening need not preserve the same greedy action.

B Finite-Horizon Regret Accounting

Proposition 1 shows that the posterior mean reward vector m_t is both the aligned full-feedback target and the Bayes-greedy decision target. We extend the accounting to any supervision scheme whose population target is a probability vector.

Warmup Conditions. Let $\mathcal{S}$ denote a supervision scheme, let $m^{\mathcal{S}}(h, x) \in \Delta^{K-1}$ be its conditional population target, and write $m_t^{\mathcal{S}} = m^{\mathcal{S}}(h_t, x_t)$. We assume the average KL-fit condition

$$\frac{1}{H} \sum_{t=1}^{H} \mathbb{E}_{d_t^\beta} \mathrm{KL}(m_t^{\mathcal{S}} \| \hat{q}_t) \le \epsilon_{\mathcal{S}}^2. \tag{35}$$

This condition concerns the learned predictor rather than the objective used to obtain it. It must be checked separately because the reported objective also contains a regret surrogate and omits early positions.

Let $e_{\mathcal{S}}(h, x) = \|q_\phi(\cdot \mid h, x) - m^{\mathcal{S}}(h, x)\|_\infty$. The pointwise-target-error shift from training to deployment is

$$\bar{s}_{\mathcal{S}} = \frac{1}{H} \sum_{t=1}^{H} \left[\mathbb{E}_{d_t^{\bar{\pi}}} e_{\mathcal{S}} - \mathbb{E}_{d_t^\beta} e_{\mathcal{S}} \right]_+ . \tag{36}$$

Here $[u]_+ = \max(u, 0)$. Unlike total variation between full trajectory distributions, this discrepancy is zero when pointwise target error transfers despite different action sequences.

Define the rollout-averaged target bias

$$\bar{b}_{\mathcal{S}} = \frac{1}{H} \sum_{t=1}^{H} \mathbb{E}_{d_t^{\bar{\pi}}} [\|m_t^{\mathcal{S}} - m_t\|_\infty]. \tag{37}$$

It is zero for aligned soft or sampled one-hot full feedback. The positive-only censored loss has a propensity-tilted population minimizer and therefore also has zero target bias for a uniform AGP, but not generally for a nonuniform AGP (Appendix A.1).

For the reported sharpened scheme, $m^{\mathcal{S}} = \mathbb{E}[s_\gamma(p_\theta(\cdot \mid x)) \mid h, x]$. Its bias $\bar{b}_{\mathcal{S}}$ is exactly zero at $\gamma = 1$. As $\gamma \to \infty$, assuming a unique maximizer almost surely, $m^{\mathcal{S}}(k)$ approaches the posterior

probability that arm k is best. Larger exponents therefore target a different objective rather than a uniformly better one. For $\gamma \neq 1$, the transform preserves each task's maximizing arm but can change the maximizing arm after posterior averaging. The analysis quantifies this distortion at the reported γ=2 but does not establish that exponent as optimal.

For $\gamma \geq 1$, Jensen's inequality and the tower property first give the γ-dependent, generator-computable bound

$$\bar{b}_{\mathcal{S}} \leq \mathbb{E}_{\theta,x}[\|s_\gamma(p_\theta(\cdot \mid x)) - p_\theta(\cdot \mid x)\|_\infty]. \tag{38}$$

At γ=2 and K=5, the right-hand side is 0.068 to 0.070 across the three held-out prior families (Table 11). A looser bound eliminates even the dependence on γ. Let $k^* \in \arg\max_k p_k$. Sharpening cannot reduce the total mass on the set of maximizers, and hence

$$\|s_\gamma(p) - p\|_\infty \leq 1 - p_{k^*}. \tag{39}$$

Indeed, the increase of a maximizing coordinate is at most $1 - p_{k^*}$; the total sharpened mass outside the maximizing set is at most its unsharpened mass, which bounds every decrease or increase of a nonmaximizing coordinate. Consequently,

$$\bar{b}_{\mathcal{S}} \leq \mathbb{E}_{\theta,x}[1 - \max_k p_\theta(k \mid x)]. \tag{40}$$

The second bound can be too loose to make the regret bound non-vacuous, especially at larger K. Even the tighter first bound is an upper bound on, not an estimate of, posterior-conditioned $\bar{b}_{\mathcal{S}}$.

The AGP β determines which censored trajectories support the training distribution. A sufficient broad-coverage condition is

$$\beta(k \mid h_t, x_t) \geq \frac{1}{CK}, \tag{41}$$

for all interaction prefixes, contexts, and arms and some $C \geq 1$. Random and mixed AGPs provide this support directly; the outer-loop controller chooses among broad and focused AGPs using validation improvement.

KL Fit.

Proposition 5 (KL fit implies L_1 fit)**.** *Under Eq.* (35)*,*

$$\frac{1}{H}\sum_{t=1}^{H} \mathbb{E}_{d_t^\beta}[\|\hat{q}_t - m_t^{\mathcal{S}}\|_1] \leq \sqrt{2}\,\epsilon_{\mathcal{S}}. \tag{42}$$

Proof. Pinsker's inequality gives $\|m_t^{\mathcal{S}} - \hat{q}_t\|_1 \leq \sqrt{2\,\mathrm{KL}(m_t^{\mathcal{S}} \| \hat{q}_t)}$. Taking the time-averaged expectation and applying Jensen's inequality proves the claim. □

The validation quantity "val norm" is an empirical proxy for decision quality, not the population KL fit, so its relationship to $\epsilon_{\mathcal{S}}$ is correlational rather than an equality.

Action Gap. Let $\hat{a}_t = \arg\max_k \hat{q}_t(k)$ and $a_t^{\mathrm{B}} = \arg\max_k m_t(k)$. Their excess one-step posterior expected pseudo-regret is

$$m_t(a_t^{\mathrm{B}}) - m_t(\hat{a}_t) \leq 2\|\hat{q}_t - m_t\|_\infty \tag{43}$$

$$\leq 2\|\hat{q}_t - m_t^{\mathcal{S}}\|_\infty + 2\|m_t^{\mathcal{S}} - m_t\|_\infty. \tag{44}$$

The first inequality inserts and subtracts $\hat{q}_t$ at both actions; its middle term is non-positive because $\hat{a}_t$ maximizes $\hat{q}_t$.

AGP Coverage. Under the uniform random AGP, arm k is pulled $n_k \sim \mathrm{Binomial}(T, 1/K)$ times. A Chernoff bound gives

$$\Pr[n_k < T/(2K)] \leq \exp(-T/(8K)), \tag{45}$$

and a union bound gives probability at least $1 - K\exp(-T/(8K))$ that every arm receives at least $T/(2K)$ pulls. This is the formal reason the random AGP gives broad evidence. The empirical AGP ablations show why broad evidence alone is insufficient: the random AGP covers arms evenly but need not match deployment trajectories.

Warmup Bound.

Theorem 1 (Quantified warmup regret bound)**.** *Fix a supervision scheme $\mathcal{S}$ with population target $m^{\mathcal{S}}$. Let d_t^{β} be the round-t interaction-prefix distribution induced by the AGP. Let $R_H(\pi^{\mathrm{B}})$ denote the pseudo-regret of the posterior-mean greedy action, evaluated counterfactually along prefixes induced by $\hat{\pi}$. If Eq.* (35) *holds, then*

$$R_H(\hat{\pi}) \le R_H(\pi^{\mathrm{B}}) + 2\sqrt{2}\, H\epsilon_{\mathcal{S}} + 2H\bar{b}_{\mathcal{S}} + 2H\bar{s}_{\mathcal{S}}. \tag{46}$$

The comparator π^{B} is a one-step posterior-mean rule, not the full-horizon Bayes-optimal explorer; the decomposition does not separate that exploration gap.

Proof. Conditioned on (h_t, x_t), the conditional optimal-reward term $\mathbb{E}[\max_k p_\theta(k \mid x_t) \mid h_t, x_t]$ is common to both actions. Therefore the difference between $\hat{\pi}$ and π^{B} is exactly $m_t(a_t^{\mathrm{B}}) - m_t(\hat{a}_t)$. Applying the action-gap inequality and summing gives

$$R_H(\hat{\pi}) \le R_H(\pi^{\mathrm{B}}) + 2\sum_{t=1}^{H} \mathbb{E}_{d_t^{\hat{\pi}}}[\|\hat{q}_t - m_t^{\mathcal{S}}\|_\infty] + 2H\bar{b}_{\mathcal{S}}. \tag{47}$$

By Eq. (36), the sum of deployment target errors is at most their sum under d_t^{β} plus $H\bar{s}_{\mathcal{S}}$. Each infinity norm is at most the corresponding one norm. Proposition 5 completes the proof. □

The terms give distinct diagnostics: the prior and induced prefix distributions determine $R_H(\pi^{\mathrm{B}})$, average KL fit determines $\epsilon_{\mathcal{S}}$, the supervision target determines $\bar{b}_{\mathcal{S}}$, and the AGP-to-deployment change determines $\bar{s}_{\mathcal{S}}$. The bound becomes a certificate only when all four are controlled.

C Horizon Boundary

Theorem 1 says amortization is useful over a finite horizon when its Bayes, KL-fit, target-bias, and pointwise-target-error-shift terms are smaller than the exploration cost paid by a prior-free online learner:

$$R_H(\pi^{\mathrm{B}}) + 2\sqrt{2}\, H\epsilon_{\mathcal{S}} + 2H\bar{b}_{\mathcal{S}} + 2H\bar{s}_{\mathcal{S}} < R_H(\mathcal{A}). \tag{48}$$

This is a finite-horizon diagnostic. It does not assert that the classical algorithms tested here are correctly specified for the nonlinear synthetic prior.

Long-Horizon Crossover.

Theorem 2 (Persistent-floor crossover)**.** *Let $\hat{\pi}$ be an amortized policy whose expected instantaneous pseudo-regret is at least $\rho > 0$ at every round. Then*

$$R_H(\hat{\pi}) \ge H\rho. \tag{49}$$

Consequently, any learner $\mathcal{A}$ with $R_H(\mathcal{A}) = o(H)$ as $H \to \infty$ eventually has lower regret. In particular, if

$$R_H(\mathcal{A}) \le CK\Delta_{\mathrm{gap}}^{-1} \log H \tag{50}$$

for a constant $C > 0$, arm count K, and reward gap $\Delta_{\mathrm{gap}} > 0$, then the sufficient crossover is the larger solution of

$$H\rho = CK\Delta_{\mathrm{gap}}^{-1} \log H. \tag{51}$$

Writing $\Xi = CK/(\rho\Delta_{\mathrm{gap}})$, this solution exists for $\Xi > e$ and obeys $H^* = \Theta(\Xi \log \Xi)$ as $\Xi \to \infty$, with absolute comparison constants for the scalar equation $H = \Xi \log H$.

Proof. Summing the assumed per-round pseudo-regret floor gives $R_H(\hat{\pi}) \ge H\rho$. A sublinear upper bound is eventually smaller than this linear lower bound. Under the displayed logarithmic bound, a sufficient crossover solves $H = \Xi \log H$; standard inversion gives the stated asymptotic form for the larger root. □

A lower bound of the form $\rho \geq p\Delta_{\text{gap}}$ follows if the policy selects a suboptimal arm with probability at least p and its conditional expected gap on those events is at least Δ_{gap}. Unlike the previously assumed one-hot model, the soft arm means used in the experiments need not have a uniform positive gap, so we state the result directly in terms of ρ.

The nonzero floor and the classical regret guarantee are explicit premises, not empirical claims about the reported LinUCB or LinTS runs. A floor can arise from prior mismatch, finite capacity, sharpening bias, or incomplete learning under censored feedback. Classical learners continue updating within an episode and may therefore overtake a frozen amortized rule; this is the structural motivation for handoff.

If the comparator also has a persistent floor $\rho_{\mathcal{A}} > 0$, as can happen when a linear learner is misspecified for the nonlinear synthetic prior, then the sublinear premise fails. When both regrets are asymptotically linear, their slopes determine the comparison: the amortized policy remains ahead when $\rho < \rho_{\mathcal{A}}$, and its advantage can grow with the horizon. The reported $T{=}200$ and $T{=}1000$ suites are therefore not a test of the certified-learner crossover theorem.

C.1 Observed Late-Window Slopes

We also estimate finite-window slopes directly from the saved cumulative pseudo-regret trajectories. Over rounds 100 to 200 in the Table 1 suite, BANDHA's slope is below LinUCB's at every tested $K \in \{3, 5, 7, 10, 13, 15, 20\}$, with all paired bootstrap confidence intervals for the difference below zero. The $K{=}10, T{=}1000$ evidence is not uniform: over rounds 500 to 1000, the main-method suite gives slopes 0.4147 versus 0.5633 (difference -0.1486, 95% CI $[-0.1545, -0.1427]$), whereas the horizon suite gives 0.5688 versus 0.5633 (difference 0.0055, 95% CI $[-0.0010, 0.0117]$). These finite-window diagnostics neither establish a universal BANDHA floor nor validate the asymptotic crossover premise.

D Handoff Experiments

If BANDHA acts for the first H rounds and a downstream algorithm $\mathcal{A}$ takes over afterward, total regret decomposes as

$$R_T(\text{PFN} \to \mathcal{A}) = R_H(\text{PFN}) + R_{H+1:T}(\mathcal{A} \mid h_H)\,. \tag{52}$$

A useful handoff prefix must reduce downstream tail regret as well as prefix regret. A greedy low-regret prefix can starve arms that a later learner must explore. Handoff quality depends on the downstream learner's regret-sensitivity operator $\Psi_{\mathcal{A}}$, i.e., which observations reduce its future regret. In the handoff teacher, the scalar α weights this downstream sensitivity term relative to the estimated one-step regret term.

For tail length J, let $R_J^{\text{frozen}}(X \mid s)$ be the remaining-horizon regret estimate obtained without allowing X to update and let $R_J^{\text{rolled}}(X \mid s)$ be the corresponding rolled estimate with online updates. We report the self-correction gap

$$\text{SC}_J(X) = \mathbb{E}_s\big[R_J^{\text{frozen}}(X \mid s) - R_J^{\text{rolled}}(X \mid s)\big]\,. \tag{53}$$

Table 5: Direct-policy and handoff comparison at $K{=}10$, $T{=}1000$ (600 environments, 2 rollout seeds; $n{=}1200$). Lower regret is better.

Policy	Regret
LinUCB alone	586.403 ± 2.575
BANDHA alone	$\mathbf{460.032 \pm 4.308}$
BANDHA $\to$ LinUCB, untargeted	554.935 ± 2.874
BANDHA for Greedy, $\alpha{=}1 \to$ LinUCB	542.776 ± 2.862

Greedy and ϵ-greedy self-correct weakly, so frozen tail estimates are close to the true downstream value. LinUCB and LinTS repair starved directions during the tail; using a frozen proxy for them overstates the value of additional coverage and can create over-exploration.

Table 6: Downstream sensitivity probe at K=10, T=1000, handoff horizon H=200 (100 environments). SC_L measures how much regret the downstream recovers by learning during the tail; large values require rolled-tail teachers rather than frozen proxies.

Downstream X	$\mathrm{SC}_L(X)$	Rolled tail	$\max \Psi_X$	Std. Ψ_X
Greedy	1.634	563.180	13.202	8.644
ϵ-greedy	1.216	564.190	10.596	6.864
LinUCB	23.613	484.086	9.810	5.923
LinTS	8.960	545.938	17.133	10.290

Table 7: Dense LinUCB handoff decomposition at K=10, T=1000, handoff horizon H=400 (400 environments, 2 rollout seeds). Tail delta and total delta are relative to LinUCB alone; negative is better.

Prefix policy	Prefix regret	Tail delta	Total regret	Total delta
Same-log	250.644±1.662	+18.886±0.656	609.454±3.183	+22.732±1.404
Greedy α=0.5	203.018±1.936	+2.459±0.599	545.400±3.483	−41.321±1.404
Greedy α=1	**202.986±1.894**	+0.789±0.615	**543.698±3.467**	**−43.023±1.459**
Greedy α=2	205.507±1.850	**−0.281±0.575**	545.150±3.449	−41.572±1.470

Table 7 separates the two pieces of handoff regret. Same-log handoff uses the direct warmup log and harms LinUCB's tail. The Greedy-targeted prefixes lower total regret by 41 to 43 points while leaving the LinUCB tail nearly unchanged, and α=2 slightly improves the tail. The evidence against a universal primer comes from this target sensitivity and from the no-target stress tests below.

Table 8: No-target and over-coverage prefixes can hurt. Excess coverage pressure handed to LinUCB can increase regret; lower is better.

Setting	Prefix + LinUCB	LinUCB alone	Change
K=10, T=250, α=0.5 coverage	172.770±0.751	159.281±0.898	+13.489
K=20, T=300, α=0.5 coverage	220.742±1.176	218.230±1.225	+2.512
K=10, T=300, no-target ϵ=1	204.653±1.158	188.360±1.389	+16.294
K=20, T=300, no-target ϵ=1	222.001±1.108	218.230±1.225	+3.771

Table 8 gives the no-target negative result. Coverage without downstream marginal value keeps paying the immediate regret cost after Ψ_X has saturated; the effect is largest in the short-tail K10 setting. The handoff rule is target-sensitive exploration: gather observations where the target learner's future regret is sensitive.

D.1 Successor Dependence of Prefix Value

Fix a total horizon T. Let $z_t = (s_t, x_t)$ denote the inherited interaction prefix together with the current context, and let $\delta_t(z_t, a)$ be the instantaneous pseudo-regret of action a. For a successor X whose action is measurable with respect to this prefix, define its finite-horizon value recursively by

$$V^X_{T+1}(z) = 0, \tag{54}$$

$$Q^X_t(z,a) = \delta_t(z,a) + \mathbb{E}[V^X_{t+1}(Z_{t+1}) \mid z, a], \tag{55}$$

$$V^X_t(z) = \mathbb{E}_{a\sim\pi_X(\cdot|z)}[Q^X_t(z,a)]. \tag{56}$$

Because the objective is regret, define the regret-reduction advantage $A^X_t(z,a) = V^X_t(z) - Q^X_t(z,a)$, so positive advantage lowers total regret. Let $\pi \to_H X$ follow prefix π for H rounds and X thereafter, and let $\mathcal{R}_T(\cdot)$ denote expected regret over all T rounds.

Proposition 6 (Prefix performance difference)**.** *For any H-round prefix π,*

$$\mathcal{R}_T(\pi \to_H X) - \mathcal{R}_T(X) = -\sum_{t=1}^{H} \mathbb{E}_{\pi\to_H X}[A^X_t(Z_t, A_t)]. \tag{57}$$

Proof. For a trajectory generated by $\pi \to_H X$, the Bellman definitions give

$$\mathbb{E}[\delta_t(Z_t, A_t) + V_{t+1}^X(Z_{t+1}) - V_t^X(Z_t)] = -\mathbb{E}[A_t^X(Z_t, A_t)]. \tag{58}$$

Summing from 1 to T telescopes, with $V_{T+1}^X = 0$ and $\mathbb{E}[V_1^X(Z_1)] = \mathcal{R}_T(X)$. After round H the hybrid policy follows X, for which the conditional expectation of A_t^X is zero. Only the first H terms remain, yielding Eq. (57). □

Corollary 1 (Exact imitation has zero value)**.** *If $\pi(\cdot \mid z_t) = \pi_X(\cdot \mid z_t)$ at every prefix round and reachable state, then $\mathcal{R}_T(\pi \to_H X) = \mathcal{R}_T(X)$.*

This corollary applies only to a prefix that exactly reproduces X; it does not identify an arbitrary untargeted prefix with the successor.

For a candidate prefix class Π_H, define

$$G_X(\pi) = \sum_{t=1}^{H} \mathbb{E}_{\pi \to_H X}[A_t^X(Z_t, A_t)]. \tag{59}$$

Proposition 6 gives the exact compatibility condition

$$\text{a common optimal prefix exists for } \mathcal{X} \quad \Longleftrightarrow \quad \bigcap_{X \in \mathcal{X}} \arg\max_{\pi \in \Pi_H} G_X(\pi) \neq \varnothing. \tag{60}$$

Distinct successors can satisfy this condition, so successor dependence alone does not prove that their optima differ. Without a compatibility assumption, however, no downstream-independent optimum is guaranteed.

Proposition 7 (No unconditional downstream-independent optimum)**.** *There exist two such successors for which the sets in Eq.* (60) *are disjoint.*

Proof. Take K=2, T=2, and H=1, with no context. Draw θ uniformly from $\{1, 2\}$ and set the reward of arm a to $\mathbf{1}[a = \theta]$. Thus either prefix action has expected immediate regret $1/2$ and reveals θ. Successor X_1 uses this observation only when arm 1 was pulled; otherwise it ignores the observation and selects arm 1. Successor X_2 symmetrically uses only an observation of arm 2 and otherwise selects arm 2. Pulling arm 1 gives total expected regret $1/2$ before X_1 and 1 before X_2; pulling arm 2 reverses these values. The unique optimal prefix action is therefore arm 1 for X_1 and arm 2 for X_2. □

At the final prefix round, Eq. (57) decomposes the exact advantage into immediate regret and the horizon-aligned change in X's continuation value. This motivates the two terms in the teacher score following Eq. (5). The implemented $\widehat{\Psi}_X$ is estimated from finite tail rollouts, so the identity neither fixes the fitted weight α nor certifies the learned prefix.

E Architecture Invariants

E.1 Permutation Equivariance

A K-agnostic bandit learner should relabel predictions when arm labels are relabelled. The generator makes the labeled prior exchangeable by applying a uniform random arm permutation to every environment, and the per-arm state table satisfies the same symmetry exactly.

Proposition 8 (Target and model equivariance)**.** *Let S_K be the set of permutations of $[K]$, and let $\sigma \in S_K$. For $v \in \mathbb{R}^K$, define $[\sigma v]_k = v_{\sigma^{-1}(k)}$. Define $h_t^\sigma = (x_1, \sigma(a_1), r_1, \ldots, x_{t-1}, \sigma(a_{t-1}), r_{t-1})$, and write $m(h, x) = \mathbb{E}[p_\theta(\cdot \mid x) \mid h, x]$. Then*

$$m(h_t^\sigma, x_t) = \sigma m(h_t, x_t), \qquad q_\phi(\cdot \mid h_t^\sigma, x_t) = \sigma q_\phi(\cdot \mid h_t, x_t) \qquad \forall\, \sigma \in S_K. \tag{61}$$

Proof. The uniform random relabeling makes the prior invariant under σ, and the likelihood of h_t^σ under a relabeled environment equals the likelihood of h_t under the original environment. Changing variables in the posterior expectation gives $m(h_t^\sigma, x_t) = \sigma m(h_t, x_t)$.

We verify equivariance at each stage of the architecture.

State update. The GRU cell and event projection use shared weights applied identically to every arm. Permuting arm labels permutes which state vector receives the update, but the update function itself is invariant: if arm a was pulled in the original prefix, arm $\sigma(a)$ is pulled in the permuted prefix, so $S^{\sigma}_{\sigma(a)} = S_a$ at every timestep. That is, the state table permutes as $S^{\sigma} = S \circ \sigma^{-1}$.

Query addition. The query $c_t = W_q x_t$ depends only on the context, not on arm labels. Adding c_t to every state preserves the permutation structure: $(S^{\sigma}_{\sigma(k)} + c_t) = (S_k + c_t)$.

Arm mixer. The Transformer encoder is permutation-equivariant over its input tokens (self-attention computes pairwise interactions without positional encoding). A permutation of input tokens produces the same permutation of output tokens.

Scoring head. The linear head $w^{\top}(\cdot)$ is applied identically to each arm token. Composing equivariant maps gives the result. □

Parameter sharing. For context dimension d and state width D, the initial state, two-layer event projection, GRU cell, and update normalization contain

$$P_{\mathrm{upd}} = 7D^2 + Dd + 13D \tag{62}$$

parameters. The reported $D{=}256, d{=}3$ model therefore uses $462{,}848$ parameters for this block. Replacing it with K independent arm-specific copies would add $(K-1)P_{\mathrm{upd}}$, equal to approximately 0.93M, 1.85M, and 4.17M parameters at $K{=}3, 5, 10$. Under uniform arm sampling, each independent copy receives T/K selected-arm transitions per episode in expectation, whereas the shared block receives all T. Under independent, equal-variance transition gradients, the variance of their empirical average therefore differs by a factor of K.

Set-size sensitivity. For a fixed query in one attention head, let z_{kj} be its score for arm j, v_j the corresponding value, and

$$o_k^{(K)} = \sum_{j=1}^{K} \omega_{kj}^{(K)} v_j, \qquad \omega_{kj}^{(K)} = \frac{\exp(z_{kj})}{\sum_{\ell=1}^{K} \exp(z_{k\ell})}. \tag{63}$$

After appending one arm while holding the existing inputs fixed,

$$o_k^{(K+1)} - o_k^{(K)} = \omega_{k,K+1}^{(K+1)} (v_{K+1} - o_k^{(K)}). \tag{64}$$

Consequently, if $\|v_j\| \le B$, then $\|o_k^{(K+1)} - o_k^{(K)}\| \le 2B\omega_{k,K+1}^{(K+1)}$. If the attention scores satisfy $z_{kj} \in [-\Lambda, \Lambda]$, then

$$\omega_{k,K+1}^{(K+1)} \le \frac{e^{2\Lambda}}{e^{2\Lambda} + K}, \tag{65}$$

so the perturbation is $O(K^{-1})$ as $K \to \infty$ for fixed B and Λ, uniformly over values and scores satisfying the displayed bounds. Normalized attention therefore prevents a change in set size from rescaling the representation solely because more rows are present. Writing $c = e^{2\Lambda}$ and applying the triangle inequality as arms are appended gives, for $K' > K$,

$$\|o_k^{(K')} - o_k^{(K)}\| \le 2Bc \sum_{j=K}^{K'-1} \frac{1}{c+j} \tag{66}$$

$$\le 2Bc \log\left(\frac{c + K' - 1}{c + K - 1}\right). \tag{67}$$

Thus the bound for each additional arm decreases strictly with K, and the accumulated layerwise drift grows logarithmically in K'/K once K exceeds c. This is a layerwise stability statement; Table 1 reports the corresponding end-to-end behavior.

E.2 Target Surrogate

For prediction $\hat{p}$ and soft teacher vector s, the surrogate

$$L_{\mathrm{reg}}(\hat{p}) = \max_k s_k - \sum_{k=1}^{K} \hat{p}_k s_k \tag{68}$$

is the expected shortfall of the randomized policy that samples $a \sim \hat{p}$ under the teacher target. It is zero exactly when $\hat{p}$ puts mass only on maximal-target arms. When $s = \widetilde{p}$ with $\gamma > 0$, its maximizing arms agree with those of the unsharpened expected reward vector p, although their gap sizes differ. The surrogate imposes a stricter zero-loss condition: all probability mass must lie on maximal-target arms. Zero greedy pseudo-regret only requires the selected argmax to maximize p. The surrogate is differentiable, so it can bias training toward decision-relevant errors without replacing the CE term whose population minimizer is the sharpened soft-label target. After posterior averaging, however, the maximizer of $\mathbb{E}[\widetilde{p} \mid h, x]$ need not equal the maximizer of $\mathbb{E}[p \mid h, x]$.

E.3 Mixed-K Evaluation

The architecture defines the same parameterized map at every arm count. Adding an unpulled arm appends a state row initialized to s_0; relabeling arms only permutes rows by Proposition 8. Thus the parameter shapes do not depend on K, and the same checkpoint can be evaluated on held-out arm counts without padding or surgery. Equivariance gives label and parameter-shape compatibility across arm counts. The held-out K experiments then measure whether the learned approximation remains accurate at unseen arm counts.

F Online Neural Baselines

NeuralUCB (Zhou et al., 2020) and NeuralTS (Zhang et al., 2021) train a 2-layer MLP online within each episode. At T=200, they cannot learn fast enough to compete in the matched neural-baseline sweep. The LinUCB column below is the matched baseline for this sweep.

K	LinUCB	Best NeuralUCB	Best NeuralTS
3	56.4	80.8 (α=0.1)	83.2 (σ=0.1)
5	82.5	103.6 (α=0.5)	106.3 (σ=0.1)
10	115.7	126.4 (α=0.1)	127.6 (σ=0.1)

Both lose to LinUCB at all K values (7% to 20% win rates), performing at epsilon-greedy level.

G Real-Data Results

Table 9 reports every dataset used in the zero-shot evaluation summarized in Section 4. The same mixed-K checkpoint and preprocessing are used throughout. BANDHA has lower mean regret than LinUCB on 15 of 17 datasets and than LinTS on all 17; Volkert and Vehicle are the two exceptions relative to LinUCB.

H Mechanism Ablations

This section separates the contributions of cross-K training, the per-arm architecture, access to censored feedback, model capacity, chunked scoring, and adaptive AGP selection. Validation regret norm is model pseudo-regret divided by uniform-policy pseudo-regret on the same held-out, AGP-generated trajectories. Lower is better, and 1 is random-level performance.

H.1 Supervision Targets

We isolate access to full feedback using the same cross-entropy training interface on both sides. The full-feedback model receives a fresh categorical sample e_Y, $Y \sim \mathrm{Categorical}(p)$, for all arms. The

Table 9: Complete real-world zero-shot evaluation (17 datasets, one mixed-K per-arm model, principal-component analysis (PCA) to $d_{\text{ctx}} = 3$, 30 shuffles, T=200). Regrets are mean ± standard error across shuffles where shown.

Dataset	K	BANDHA	LinUCB	LinTS	Win vs LinTS
Iris	3	**19.7±0.7**	40.0	55.0±1.6	100%
Wine	3	**12.9±0.4**	17.3	31.3±0.9	100%
Balance	3	**50.2±1.7**	56.9	68.2±1.4	93%
CMC	3	**120.2±1.7**	122.8	126.3±1.5	73%
Page-blocks	5	**24.2±1.1**	79.9	102.0±4.0	100%
Satimage	6	**57.9±1.5**	62.7	104.2±1.6	100%
Shuttle	7	**41.3±1.3**	72.0	112.6±3.5	100%
Segment	7	**90.4±2.3**	116.5	143.2±1.5	100%
Covertype	7	**125.0±2.0**	139.1	157.3±2.0	97%
Texture	11	**135.9±2.3**	146.8	172.6±1.4	100%
Yeast	10	**139.9±1.8**	153.2	172.4±1.3	100%
Pendigits	10	**123.0±2.2**	135.7	164.3±1.2	100%
Volkert	10	145.6±2.7	**144.3**	171.5±1.0	97%
Optdigits	10	**126.1±2.2**	132.1	166.5±1.5	100%
Letter	26	**187.7±0.8**	189.8	192.0±0.5	70%
Vehicle	4	122.3±1.6	**120.7**	134.3±1.6	87%
Jannis	4	**107.1±1.7**	109.2	125.3±1.6	90%

censored model receives only the selected arm's binary outcome. Generated episodes, fixed-random AGP, architecture, mask, and optimization budget are held fixed at K=5 and T=200. Each of three independently trained checkpoints is evaluated on the same 450 held-out environment and rollout pairs.

Table 10: Matched full-versus-censored supervision at T=200. Entries are means ± standard errors across three training runs. Lower regret and higher optimal-arm accuracy are better.

Training target	Regret	Optimal-arm accuracy
Sampled full feedback	63.92 ± 0.56	0.531 ± 0.004
Censored binary feedback	69.01 ± 1.20	0.502 ± 0.007

The paired full-minus-censored difference is -5.09 ± 1.35 regret and $+0.029 \pm 0.008$ optimal-arm accuracy, with standard errors over training-run means. Full feedback has lower regret in all three runs. This comparison isolates the feedback channel; it does not test an advantage of soft over sampled one-hot full-feedback labels or of sharpening.

Separately, Table 11 reports target-side quantities on the same held-out suite. The Rao-Blackwell variance reduction is largest for formula tasks, yielding a directional prediction for the B versus C comparison. The pointwise sharpening distortion is similar across families; it is not the prefix-conditioned posterior bias in Appendix B.

Table 11: No-training target diagnostics on the held-out K=5 supervision suite. The last column is a generator-only upper bound on pointwise sharpening distortion, not an estimate of posterior bias.

Prior	$1 - \|p\|_2^2$	$\|s_2(p) - p\|_\infty$	$\mathbb{E}[1 - \max_k p_k]$
Formula	0.314	0.070	0.255
GP	0.210	0.070	0.149
MLP	0.205	0.068	0.145

H.2 MIXED-K VALIDATION

The same checkpoint performs similarly across random, Thompson-sampling, and greedy validation trajectories. Performance degrades smoothly as K grows, and the held-out K=7 and K=15 rows do not show a discontinuity relative to neighboring trained arm counts.

Table 12: Mixed-K val norm (256-dimensional hidden states, 12 layers, 1000 epochs). K=7 and K=15 are held out.

Policy	K=3	K=5	K=7*	K=10	K=15*
vRnd	0.242	0.399	0.549	0.516	0.693
vTS	0.233	0.397	0.547	0.526	0.688
vGrd	0.294	0.404	0.560	0.530	0.701

Table 13: Best-checkpoint validation norm from one training run per configuration for the shared per-arm model and the earlier fixed-K v9 specialist. Positive improvement favors the per-arm model; training-run uncertainty is not estimated.

K	Per-arm (D=256, 12L)	v9 specialist	Improvement
3	0.242	0.295	+18%
5	0.399	0.522	+24%
10	0.516	0.788	+35%

H.3 Per-Arm Models

The shared per-arm architecture improves validation norm at every trained arm count, with the largest relative gain at K=10. This comparison changes the architecture as well as its ability to share parameters across arm counts, so it supports the combined design rather than isolating equivariance alone.

H.4 Feedback Ablation

Zeroing out bandit feedback reduces accuracy from 32.2% to 9.2% (random chance at K=10 is 10%). The model depends entirely on partial feedback for its predictions.

H.5 Capacity Ablation

12-layer mixer improves over 6-layer by 5% to 10% at all K. 512-dimensional hidden states provide marginal improvement over 256-dimensional hidden states (0.351 vs. 0.399 at K=5) at 4× compute. The 256-dimensional, 12-layer model is the practical sweet spot.

H.6 Chunked Scoring

Chunked Transformer scoring reduces K=10 epoch time from 320s to 19s (16.8× speedup) without changing scorer semantics.

H.7 Seed Variance

Three seeds at K=10 (512d checkpoints): mean PFN regret 96.4 ± 2.4 vs. LinUCB 115.7. Win rates range from 85% to 88% across seeds.

H.8 Adaptive AGP

The arm generation policy controls what evidence the model sees during pretraining. Table 14 separates the controlled same-initialization comparison from reference runs using a different initialization. In the single-initialization K=5 comparison, the Thompson Sampling outer loop improves best validation norm from 0.400 under the fixed random AGP to 0.344. The mixed-K adaptive run reaches 0.280, the lowest measured value, but it also changes the mixed-K schedule and is therefore reported separately rather than used as the controlled estimate of the adaptive-AGP gain.

AGP quality differs from online bandit quality. Pure LinTS and pure LinUCB AGPs both underperform the random AGP in this experiment. They concentrate evidence too early, while the random

Table 14: Adaptive AGP selection vs. fixed AGPs. Lower validation norm is better. The first block holds the training initialization fixed; the second block uses a separate initialization and is reported only as a reference.

Protocol	AGP	Best K=5 val norm
Controlled K=5	Fixed random	0.400
Controlled K=5	Adaptive single-K	**0.344**
Controlled K=5	Fixed LinTS	0.491
Controlled K=5	Fixed LinUCB	0.491
Mixed-K adaptive	Adaptive mixed-K	**0.280**
Reference	Fixed random	0.322
Reference	Fixed mixture	0.334

AGP covers all arms but does not match later deployment trajectories. The adaptive controller is useful because AGP quality is nonstationary: broad trajectories are useful early, and focused trajectories become useful after the model has learned enough structure.

Table 15: Multi-policy validation for K=5 AGP choices. The random AGP is best on random-policy validation, while the mixture AGP reduces deployment-trajectory mismatch for ϵ-greedy and LinUCB-style trajectories.

Training AGP	vRnd	vUCB	vTS	vEps	vGrd
Random	**0.407**	0.423	**0.399**	0.712	**0.445**
Mixture	0.411	**0.401**	0.417	**0.609**	0.452

Table 15 shows why the outer loop has a real decision to make. Changing the AGP changes which deployment trajectories the model is good at: the mixture AGP improves vEps by 14.5% and vUCB by 5.2%, while the random AGP retains slightly better random-policy and Thompson-sampling validation.

I Synthetic Task Generation

We adapt the synthetic-prior construction used by PFNs and TabPFN (Müller et al., 2022; Hollmann et al., 2023): sample latent functions, evaluate them on tabular contexts, and convert their outputs into bandit reward distributions. We specify the prior mixture, soft arm targets, AGPs that generate censored trajectories, and sharded train/validation/test splits used in our experiments.

I.1 Prior Mixture

Each training environment is sampled from one of three prior families, chosen uniformly at random under the *mixed* prior setting.

Formula Trees Following the PFN prior (Müller et al., 2022), we sample a random expression tree over the context features. Internal nodes are drawn from $\{+, -, \times, \div, \sin, \cos, \exp\}$ and leaves are context dimensions or scalar constants. The tree is evaluated to produce a latent scalar $f(x) \in \mathbb{R}$ for each context x.

Random MLPs A 2-layer MLP with random weights is sampled: hidden dimension drawn from $\{16, 32, 64\}$, weights $\sim \mathcal{N}(0, 1/\text{fan_in})$, ReLU activation. The output is a scalar $f(x)$.

Gaussian Processes A sample from a GP with an RBF kernel: $k(x, x') = \exp(-\|x - x'\|^2/2\ell^2)$ where the lengthscale ℓ is drawn uniformly from $[0.1, 2.0]$.

I.2 Soft Arm Targets

Given the latent function f, we discretize into K arms using calibrated quantile thresholds. For a batch of n context evaluations $\{f(x_i)\}_{i=1}^n$, we compute the k/K quantiles for $k = 1, \ldots, K-1$ to define arm boundaries.

Soft arm target probabilities are computed via Gaussian CDF binning:

$$p_\theta(k \mid x) = \Phi\left(\frac{\tau_k - f(x)}{\sigma}\right) - \Phi\left(\frac{\tau_{k-1} - f(x)}{\sigma}\right), \tag{69}$$

where $\tau_0 = -\infty$, $\tau_K = +\infty$, and σ is a smoothing parameter derived from the inter-quantile range. Each environment then draws a uniform random permutation of the K arm labels and applies it to the probability coordinates. The resulting prior over labeled arms is exchangeable even though the bins are constructed in quantile order. For each context, the environment samples $Y_t \sim \text{Categorical}(p_\theta(\cdot \mid x_t))$ and returns $r_t = \mathbf{1}[a_t = Y_t]$. Hence $p_\theta(k \mid x_t)$ is both the conditional expected reward of arm k and the full-feedback teacher target. The optimal arm is $a^*_\theta(x_t) = \arg\max_k p_\theta(k \mid x_t)$.

I.3 AGP Trajectories

Each environment exposes the soft full-feedback tensor to the training scaffold. Training trajectories are then generated by the AGP used for the run; the uniform random AGP is the simplest fixed choice. At each step t, the selected arm a_t is recorded, and the binary reward $r_t = \mathbf{1}[a_t = Y_t]$ is observed. The model's input at step t consists of context x_t and the censored-feedback event (a_{t-1}, r_{t-1}) from the previous step. In the per-arm state-table architecture, the event encoder and GRU cell process this feedback to update only the pulled arm's state; the remaining arms retain their states from the previous step.

I.4 Sharded Splits

For large-scale training, we pre-generate environments into sharded files on disk. Each shard contains N environments with tensors: `contexts` $[N, T, d]$, `arm_means` $[N, T, K]$ storing the soft target vectors, and `true_arms` $[N, T]$ storing the sampled labels Y_t. Shards are organized by prior type, K, d, and T, with separate train/val/test splits using disjoint seed ranges. Training streams shards via an `IterableDataset` with per-worker shard splitting and within-shard shuffling.

J General PFN Construction

We outline how the construction could be extended beyond tabular contexts. Let a PFN task generator sample a latent task θ, a sequence of contexts $x_{1:T}$, and a full-feedback teacher map $y_\theta(x_t, \cdot)$ for each timestep. Training trajectories are generated by an AGP that observes only the censored bandit stream

$$h_t = (x_1, a_1, r_1, \ldots, x_{t-1}, a_{t-1}, r_{t-1}).$$

The supervised target is the posterior mean teacher object

$$\mathbb{E}[y_\theta(x_t, \cdot) \mid h_t, x_t],$$

estimated by the synthetic teacher from its full-feedback access to y_θ after the trajectory has been generated. The soft cross-entropy loss in the main text is the discrete probability-vector instantiation of this idea. Other reward spaces would replace it with an appropriate proper scoring rule or regression loss for the teacher's full-feedback object. At deployment, the teacher is removed and the policy receives only (x_t, a_t, r_t) feedback.

For a new PFN domain, the required replacements are modular. The context encoder must match the domain, e.g., a vision encoder for images, a language encoder for text, or a schema-aware encoder for arbitrary tables. The task generator must sample realistic latent tasks and expose full-feedback teacher targets during training. The arm architecture must respect the action space: a per-arm equivariant state table is appropriate for finite exchangeable arms, while continuous or structured actions require a scorer over action features or a generative action proposal mechanism. The censored-feedback objective itself is unchanged: train an amortized map from censored interaction prefixes

and the current context to posterior teacher estimates, then connect those estimates to the deployment reward for the domain.

K Training Details

K.1 Execution and Model Sizes

Section 2 and Figure 2 define the per-arm architecture. Here we give its training execution and exact model sizes.

During training, episodes are teacher-forced: the full AGP-generated trajectory is processed in a single forward pass with the GRU applied sequentially over time steps. A chunked scoring variant collects state snapshots across time and scores them in batches, reducing epoch time by $16.8\times$ at $K{=}10$.

Table 16: Model configurations.

Config	d_{model}	Layers	Heads	FF mult
Small	128	6	4	4
Main	256	12	8	4
Large	512	12	8	4

K.2 Optimization

- Optimizer: AdamW with lr $= 3 \times 10^{-4}$, weight decay 10^{-4}.
- Scheduler: cosine annealing over the full training run.
- Gradient clipping: max norm 1.0.
- Mixed precision: bfloat16 for the large model, float32 for the small model.
- Gradient accumulation: 4 micro-batches for the large model.
- Activation checkpointing: enabled for the large model.

K.3 Loss Positions

Reported checkpoints use the composite objective in Eq. (2): soft cross-entropy on the teacher target after fixed-exponent sharpening, plus the expected-regret surrogate with $\lambda{=}0.5$. Unless a run is explicitly marked full-loss, training computes this objective after the earliest quarter of each episode. This mask prevents the objective from being dominated by nearly observation-free positions; online regret is still evaluated from the first round onward.

K.4 Reported Runs

Table 17: Training configurations for reported results. The $K{=}20$ scaling row is a separate run and is not the $K \in \{3, 5, 10\}$ checkpoint reported in Table 1.

Setting	Model	Epochs	Batch	T	Notes
Mixed-K 6L	Main ($D{=}256$)	600	32	200	$K \in \{3, 5, 10\}$ round-robin
Mixed-K 12L	Main ($D{=}256$)	1000	32	200	depth expansion from 6L
Adaptive AGP	Main ($D{=}256$)	500	32	200	TS over AGPs
$K{=}20$ scaling	Main ($D{=}256$)	400	32	200	$K \in \{3, 5, 10, 20\}$

L Evaluation Protocols

L.1 Online Evaluation

Unless a table caption states otherwise, an evaluation run creates $N_{\text{env}} = 100$ fresh environments from the specified prior and is repeated over $N_{\text{seed}} = 3$ rollout seeds. Table 1 instead uses 100

environments and 10 independent rollout seeds, as stated in its caption. For each environment and seed:

1. Sample contexts $x_1, \ldots, x_T$, compute the soft target $p_\theta(\cdot \mid x_t)$, and draw $Y_t \sim \mathrm{Categorical}(p_\theta(\cdot \mid x_t))$.
2. Run each method sequentially for T rounds: at step t, the method selects arm a_t, observes reward $r_t = \mathbf{1}[a_t = Y_t]$, and updates its state.
3. Record cumulative pseudo-regret $R_t = \sum_{s=1}^{t} [\max_k p_\theta(k \mid x_s) - p_\theta(a_s \mid x_s)]$ at checkpoints $t \in \{10, 20, 50, 100, 200\}$.

BANDHA uses greedy arm selection: $a_t = \arg\max_k f_\theta(k \mid h_t, x_t)$, where f_θ is the model's output at position t conditioned on the self-generated interaction prefix h_t. The model runs autoregressively, feeding its own actions and rewards back as input.

Baselines

- **LinUCB** (Li et al., 2010): $\alpha = 1.0$. Maintains per-arm ridge regression with UCB exploration bonus.
- **LinTS** (Agrawal & Goyal, 2013): $v = 1.0$. Samples from the posterior of per-arm linear models.
- **ϵ-greedy**: $\epsilon = 0.1$. Maintains empirical mean rewards per arm.
- **Random**: uniform arm selection.

All baselines receive the same contexts and reward realizations as BANDHA. For Table 1, the expected-uniform column analytically averages the selected arm on each of the same 100 environments: $\sum_t [\max_k p_\theta(k \mid x_t) - 1/K]$. Its standard error is across environments; no random-action Monte Carlo is used.

Win Rate For each paired environment and rollout-seed unit, we compare cumulative regret at T=200. The win rate is the fraction of units where BANDHA achieves strictly lower regret than the baseline.

L.2 OFFLINE EVALUATION

Offline evaluation measures the model's prediction quality on held-out AGP-generated trajectories (teacher-forced, not autoregressive). We report:

- **Accuracy**: fraction of positions where $\arg\max_k q(k \mid h_t, x_t) = \arg\max_k p_\theta(k \mid x_t)$.
- **Accuracy F20**: accuracy on the last 20% of positions (where the model has the most preceding observations).
- **Regret norm**: ratio of model regret to random-policy regret on the same environments. A value of 1.0 indicates random-level performance; lower is better.

L.3 HANDOFF EVALUATION

The handoff protocol evaluates the deployment scenario where BANDHA runs for H warmup rounds, then a conventional algorithm takes over for the remaining $T - H$ rounds. The conventional algorithm inherits the full interaction prefix generated by BANDHA.

We test handoff points $H \in \{50, 100, 200, 400\}$ with total horizon $T = 1000$, comparing:

- BANDHA $\to$ LinUCB at each H
- BANDHA $\to$ LinTS at each H
- Pure LinUCB for all T rounds
- Pure LinTS for all T rounds