SCALABLE THOMPSON SAMPLING VIA ENSEMBLE++

Anonymous authors
Paper under double-blind review

ABSTRACT

Thompson Sampling is a principled uncertainty-driven method for active exploration, but its real-world adoption is impeded by the high computational overhead of posterior maintenance in large-scale or non-conjugate settings. Ensemble-based approaches offer partial remedies, but often require a large ensemble size. This paper proposes the Ensemble++, a scalable agent that sidesteps these limitations by a shared-factor ensemble update architecture and a random linear combination scheme. We theoretically justify that in linear bandits, Ensemble++ agent only needs an ensemble size of $\Theta(d \log T)$ to achieve regret guarantees comparable to exact Thompson Sampling. Further, to handle nonlinear rewards and complex environments. we introduce a neural extension that replaces fixed features with a learnable representation, preserving the same underlying objective via gradient-based updates. Empirical results confirm that Ensemble++ agent excel in both sample efficiency and computational scalability across linear and nonlinear environments, including GPT-based contextual bandits for automated content moderation – a safety-critical foundation model online decision-making task.

1 INTRODUCTION

Balancing *exploration* and *exploitation* is a core challenge in sequential decision-making problems, with applications ranging from online recommendation systems, automated content moderation to robotics, personalized healthcare and computer-using agent. A prominent Bayesian solution is *Thompson Sampling* (TS) (Thompson, 1933; Russo et al., 2018), which maintains a posterior – the uncertainty estimates – over unknown parameters (or reward functions). At each step, it samples a model hypothesis from this posterior and selects the action appearing optimal under that hypothesis, elegantly balancing exploration of uncertain actions and exploitation of seemingly high-reward ones. Despite its elegant theory and strong empirical performance in simpler (conjugate) bandit scenarios, TS encounters serious *scalability* hurdles in modern settings with high-dimensional or non-conjugate (e.g., neural) models. Maintaining exact posterior samples can be computationally prohibitive, often requiring iterative approximation methods (Laplace, MCMC, or variational inference) that become expensive as the time horizon T grows. These methods may also yield biased uncertainty estimates, undermining TS's exploration.

Ensemble-Based Approximate Sampling. A widely adopted alternative to full Bayesian updates is *ensemble sampling*, which keeps M model replicas in parallel and randomly picks one each round to act, thus approximating Thompson Sampling's "draw from the posterior" step (Osband & Van Roy, 2015; Osband et al., 2016; 2019). However, the only existing theoretical result that *matches* TS's optimal regret in linear bandits requires $M = O(T \cdot |\mathcal{X}|)$ (Qin et al., 2022) where $\mathcal{X}$ is the action space. This large-M requirement may be infeasible in high-dimensional or long-horizon tasks. Moreover, many ensembles demand either repeated retraining or large architectural overhead, raising practical concerns in real-time or resource-constrained environments. Recent methods like *Ensemble+* (Osband et al., 2018; 2019) and *EpiNet* (Osband et al., 2023a;b) refine ensemble-based exploration in deep networks by adding randomized prior functions or "epistemic indices." They typically rely on fairly large ensembles or architectural overhead, and offer no rigorous understanding. This highlights the gap between empirical feasibility and theoretical understanding.

1.1 KEY CONTRIBUTIONS

In this paper, we propose *Ensemble++* agent for scalable approximate Thompson Sampling that obviates the need for large ensemble sizes or costly per-step retraining:

- *Linear Ensemble++ Sampling.* It incrementally updates a *single shared* ensemble matrix factor to capture posterior-like uncertainty and approximate TS via random linear combinations of the ensembles. We show that, in d-dimensional linear bandits, Ensemble++ with ensemble size $M = O(d \log T)$ suffices to *match the regret order* of exact TS in various decision set settings. It dramatically improves prior analysis (Qin et al., 2022), which requires $M = O(T \cdot |\mathcal{X}|)$.
- *Neural Extension.* We extend these ideas by incorporating neural networks, replacing fixed linear features with a learnable neural feature extractor while keeping the same incremental-update principle. This yields a flexible approach for complex, high-dimensional reward functions.
- *Empirical Validation.* Through comprehensive experiments on synthetic and real-world benchmarks–including large-scale neural tasks involving GPTs–we demonstrate that Ensemble++ achieves superior regret-*vs*-computation trade-offs compared to leading baselines such as Ensemble+ and EpiNet; and validate the theoretical results of linear Ensemble++ sampling.

This work both closes a longstanding theoretical gap in linear ensemble sampling and provides a flexible framework for deeper models. We describe linear Ensemble++ sampling and neural extension in Section 3, provide full theoretical analysis of our linear scheme in Appendix D, and present empirical evaluations in Appendix B, building on the foundational concepts in Section 2.

2 Background and Related Work

2.1 Motivation: Content Moderation in Real-Time

Modern social-media platforms handle a vast volume of user-generated content every second, creating a *critical need* for automated moderation (Gorwa et al., 2020; Roberts, 2019). Historically, human reviewers manually inspected each post to detect policy violations. However, as platforms like Facebook (Meta, 2024), Twitter (Corp., 2024), and Reddit (Reddit, 2024) expanded to hundreds of millions of users, *fully manual moderation* became infeasible. Consequently, *AI-driven moderation systems* emerged, often leveraging *foundation models* (Weng et al., 2023) (large pretrained language or vision models) for *real-time* filtering.

Despite robust performance on the distributions seen during training, these large models often face **high uncertainty** in *novel* or *rare* content: emergent slang, subtle or borderline hate speech, or newly formed harassment styles (Markov et al., 2023). A purely *deterministic* policy (e.g., the model's single best guess) can err severely by

- **over-blocking** benign content (harmful to user experience), or
- **under-blocking** hateful material (a safety hazard).

Hence, **human feedback** remains indispensable for correcting the system, especially on ambiguous or boundary cases. The key dilemma is *when* to rely on human (which yields better learning but increases workload) versus *auto-removing* content (which saves labor but risks higher error).

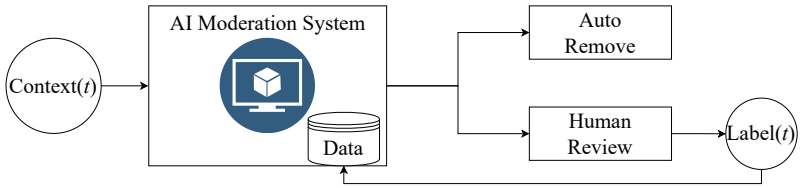


Figure 1: Real-time decision-making pipeline for content moderation. At each time t, the AI moderation system receives a post x_t, decides to **auto-remove** or **request human review**, then obtains feedback (if reviewed) to update its policy. This setup inherently involves uncertainty about *borderline* or *novel* content.

Human-AI Collaboration. Figure 1 depicts a *human-in-the-loop* moderation pipeline:

1. A new post x_t arrives.
2. The AI system either *auto-removes* it or *requests a human review*.
3. If reviewed, a moderator provides a corrective label y_t (e.g., "hate" or "benign"), and the AI system updates its internal policy.
4. Over time, decisions become more accurate, reducing human intervention.

Balancing *exploitation* (avoiding unnecessary reviews) and *exploration* (gathering feedback on uncertain content) is central to improving moderation quality while minimizing reviewer workload. This tension aligns well with a *contextual bandit* formulation.

Contextual Bandit Formulation We model moderation as a **contextual bandit** problem (Wang et al., 2005; Langford & Zhang, 2007):

- **Context** x_t: the textual (or multimedia) representation of the post at time t.
- **Action set** $\mathcal{A}_t = \{\text{auto-remove}, \text{ human-review}\}$.
- **Reward** $r_t \in \mathbb{R}$: quantifying correctness vs. cost. For example:
  - $+1$ for correctly publishing benign content,
  - -0.5 for inadvertently publishing hateful or disallowed content,
  - $+0.5$ for blocking any post (safer fallback, but potentially suboptimal if content was benign).

At each step, the agent chooses an action based on the context $\{x_1, \ldots, x_{t-1}\}$ and partial knowledge gained so far. Crucially, the agent must *explore* suspicious or uncertain contexts (requesting reviews) to learn from human labels, while *exploiting* confident predictions (auto-removing) to conserve human effort.

This *exploration-exploitation* trade-off, typical of contextual bandits, poses a significant challenge for large-scale moderation pipelines. As we show next, *foundation models alone* are not sufficient to address this adaptivity, motivating the need for an ensemble-based approach like *Ensemble++*.

2.2 Challenges of Foundation Models in Online Decision-Making

Large foundation models (e.g. GPT series) have shown remarkable generalist capability but lack intrinsic **uncertainty modeling and adaptive exploration** (Krishnamurthy et al., 2024). Indeed, even top-tier LLMs can fail in multi-armed bandit or contextual-bandit scenarios if not provided with explicit "memory" or "sampling" mechanisms (Krishnamurthy et al., 2024). Hence, **foundation models alone** often struggle in large-scale, real-time moderation because:

- **Uncertainty Estimation.** Large models do not, by default, provide robust estimates of *how uncertain* they are on out-of-distribution content. As a result, they can incur high misclassification rates for novel forms of hate speech, rapidly changing memes, or new harassment tactics.
- **Incremental Adaptation at Scale.** The moderation stream is both continuous and high-volume. We need an approach that updates quickly (in near-constant or modest cost per step) to keep pace with new data, while preserving strong overall performance.

In short, to address *rare* or *emerging* forms of hateful content, a model must **actively explore** uncertain contexts and incorporate *human corrections* with minimal overhead.

2.3 Thompson Sampling and the Scalability Dilemma

Thompson Sampling (TS) addresses the exploration-exploitation dilemma but its implementation with complex models such as foundation models can become prohibitively expensive in practice.

Principled exploration with Thompson Sampling. A popular Bayesian approach to sequential decision-making is *Thompson Sampling* Thompson (1933); Russo et al. (2018). Given a posterior distribution over an unknown reward function f^* (or unknown parameters θ^*), Thompson Sampling operates as follows at each time t:

1. *Sample* a hypothesis θ_t from the current posterior,
2. *Select* the action $X_t = \arg\max_{x \in \mathcal{X}_t} f_{\theta_t}(x)$ under that hypothesis,
3. *Observe* the reward Y_t,
4. *Update* the posterior distribution given (X_t, Y_t).

By sampling from its posterior, TS naturally balances *exploration* of uncertain regions with *exploitation* of apparently high-reward arms.

The scalability dilemma. When environment model belongs to a *conjugate* family (e.g., linear–Gaussian), posterior updates remain analytically tractable. However, in high-dimensional or *non-conjugate* cases (e.g., quadratic functions or neural nets),

- *Exact* Bayesian updates become expansive or even intractable (Russo et al., 2018),

- *Approximate* methods (Laplace, MCMC, variational inference) can introduce large computational overheads and/or biased uncertainty estimates MacKay (1992); Welling & Teh (2011); Blei et al. (2017); Xu et al. (2022).

This tension—retaining Thompson Sampling's conceptual appeal versus keeping per-step overhead manageable—motivates scalable approaches to maintaining approximate posteriors in large-scale, complex environments.

2.4 Local Perturbation and Ensemble Methods

Local perturbation. An elegant technique for *linear–Gaussian* bandits is *local perturbation* Papandreou & Yuille (2010), which updates a "perturbed" parameter in $\mathcal{O}(d^2)$ per step to emulate posterior samples. Concretely, suppose rewards follow $Y_s = X_s^\top \theta^* + \omega_s^*, \quad \omega_s^* \sim \mathcal{N}(0,1)$, and $\theta^* \sim \mathcal{N}(\mu_0, \Sigma_0)$. After t observations, naively sampling from $\mathcal{N}(\mu_t, \Sigma_t)$ would cost $\mathcal{O}(d^3)$ due to matrix factorizations. Instead, local perturbation *incrementally* maintains: $\tilde{A}_t = \Sigma_t \Big(\Sigma_{t-1}^{-1} \tilde{A}_{t-1} + X_t Z_t\Big)$, where $\tilde{A}_0 \sim \mathcal{N}(0, \Sigma_0)$ and $Z_s \sim \mathcal{N}(0,1)$. *If* $\{X_s\}$ *were fixed (i.e. non-adaptive)*, $\tilde{A}_t \mid \{X_s\}_{s \le t} \sim \mathcal{N}(0, \Sigma_t)$, so $\mu_t + \tilde{A}_t$ reproduces the exact posterior draw. Crucially, each update costs only $\mathcal{O}(d^2)$. See details in Appendix A.4.

Sequential dependency pitfall. However, once actions are chosen adaptively, the chosen actions $\{X_t\}$ depend on prior parameters $\{\tilde{A}_s\}_{s<t}$ due to the interplay between sequential action selection based on recursively updated models. This *sequential dependency* causes the conditional distribution $\tilde{A}_t \mid \{X_s\}_{s \le t}$ to no longer match $\mathcal{N}(0, \Sigma_t)$ due to the break down of independence assumptions, leading to *biased* draws (details in Appendix A.5).

One potential workaround is to *resample* fresh random perturbations $\{Z_s\}_{s \le t}$ and re-fit from scratch each step (e.g., storing all historical data) to restore independence, but this is computationally and memory expensive in practice (Osband et al., 2019; Kveton et al., 2020a) and defeats the purpose of a cheap incremental method. This challenge motivates ensemble-based approaches, hoping to mitigate sequential dependency without full resampling.

Ensemble sampling. A popular approximate-sampling strategy is *ensemble sampling* (Osband & Van Roy, 2015; Osband et al., 2016; Lu & Van Roy, 2017), which maintains M models, each evolving through independent local perturbations. Each round, one model is selected uniformly at random to choose the action, mimicking the posterior-draw step of TS. Empirically, moderate ensemble sizes (e.g., $M = 20 \sim 100$) often perform well. However, from a theoretical standpoint, Qin et al. (2022) show the only known approach that *matches* exact TS's $\sqrt{T}$-type regret in linear bandits demands $M = O(T \cdot |\mathcal{X}|)$. Although this does not necessarily mean large M is *always* needed, it leaves a *major gap* in the existing theory: can we achieve TS regret with a smaller M in linear or high-dimensional settings?

Neural ensembles. Ensemble sampling extends readily to neural function approximators by training each member on perturbed or bootstrapped data. Methods like *Ensemble+* (Osband et al., 2018; 2019) and *EpiNet* (Osband et al., 2023a) refine how uncertainty is injected into deep networks—e.g., through random prior functions or "epistemic indices" concatenations. Despite feasibility in practice, these approaches often maintain large ensembles or architectural overhead, and lack of theoretical understanding. See Appendix A for a more comprehensive review.

Need for a scalable approach. In summary, existing incremental-sampling schemes either (i) become biased under adaptive data (local perturbation), or (ii) require large ensembles and extra architectural components. In neural or high-dimensional contexts, these costs can be prohibitive. Our method, *Ensemble++*, addresses these issues.

3 Ensemble++ Agent for Scalable Thompson Sampling

We now introduce *Ensemble++* agent, a *unified* and *scalable* approach to approximate Thompson Sampling in both *linear* and *nonlinear* bandit environments. The key technical novelty is to maintain a *shared ensemble factor* incrementally, thereby approximating posterior covariance (in the linear case) or capturing epistemic uncertainty (in the neural case) without requiring a large ensemble size or repeated retraining from scratch. We begin with *Linear Ensemble++ Sampling*

(Section 3.1), describing its incremental matrix-factor updates and explaining how it approximates Thompson Sampling with only $M \approx d \log T$ ensemble directions. We then extend these ideas to general *Ensemble++* (Section 3.3), using the same *symmetrized regression* principle (Section 3.2) but replacing linear features with a trainable neural representation.

3.1 Linear Ensemble++ Sampling

Consider a *linear contextual bandit* where each action $X_t \in \mathbb{R}^d$ and reward

$$Y_t \;=\; \langle \theta^*,\, X_t \rangle + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0,1).$$

Let (μ_t, Σ_t) be the usual ridge-regression posterior updates:

$$\Sigma_t^{-1} = \Sigma_{t-1}^{-1} + X_t X_t^\top, \quad \mu_t = \Sigma_t \Big(\Sigma_{t-1}^{-1} \mu_{t-1} + X_t Y_t \Big). \tag{1}$$

Naively sampling from $\mathcal{N}(\mu_t, \Sigma_t)$ each step requires $\mathcal{O}(d^3)$ matrix factorizations. *Linear Ensemble++ Sampling* avoids this by maintaining an ensemble matrix $\mathbf{A}_t \in \mathbb{R}^{d \times M}$ that approximates $\Sigma_t^{1/2}$ incrementally:

Initialization. Construct $\mathbf{A}_0 = \frac{1}{\sqrt{M}} \big[\tilde{A}_{0,1}, \ldots, \tilde{A}_{0,M} \big]$ with each $\tilde{A}_{0,m} \sim \mathcal{N}(0, \Sigma_0)$.

Per-step procedure $(t = 1, \ldots, T)$:

1. *Action selection*: Sample a "reference" vector $\zeta_t \in \mathbb{R}^M$ from P_ζ (e.g., Gaussian). Form
$$\theta_t(\zeta_t) \;=\; \mu_{t-1} \;+\; \mathbf{A}_{t-1}\, \zeta_t, \tag{2}$$
via a random linear combination of the matrix columns, then choose $X_t = \arg\max_{x \in \mathcal{X}_t} \langle x,\, \theta_t(\zeta_t) \rangle$.
2. Observe reward Y_t, sample a "perturbation" vector $\mathbf{z}_t \in \mathbb{R}^M$ from $P_{\mathbf{z}}$, and *update* μ_t via Equation (1) and:
$$\mathbf{A}_t = \Sigma_t \Big(\Sigma_{t-1}^{-1} \mathbf{A}_{t-1} + X_t\, \mathbf{z}_t^\top \Big), \tag{3}$$

Approximate Posterior Sampling. With $M = \widetilde{\mathcal{O}}(d \log T)$, our analysis (see Appendix D) shows that $\frac{1}{2}\, \Sigma_t \;\preccurlyeq\; \mathbf{A}_t\, \mathbf{A}_t^\top \;\preccurlyeq\; \frac{3}{2}\, \Sigma_t, \quad \forall\, 1 \le t \le T$, with high probability. Hence, for $\zeta \sim \mathcal{N}(0, I_M)$, the random vector $\mu_t + \mathbf{A}_t \zeta$ serves as an *approximate* sample from $\mathcal{N}(\mu_t, \Sigma_t)$, enabling near-Thompson Sampling performance while only storing $M \approx d \log T$ ensemble directions.

Advantages (Linear Setting).

- *Small Ensemble Size*: $M \simeq d \log T$ suffices, far less than naive $M = \Omega(|\mathcal{X}|T)$ from prior ensemble sampling analysis (Qin et al., 2022) that matches regret order of TS. Each step in Ensemble++ costs $\mathcal{O}(d^2\, M)$.
- *Near-optimal Regret*: We prove that Linear Ensemble++ Sampling matches the regret order of exact TS (Appendix D).

We emphasize that P_ζ and $P_{\mathbf{z}}$ can be chosen from distributions like Gaussian, uniform-on-sphere, coordinate or cube, each with different performance (Appendices B and D).

3.2 A Symmetrized Ridge-Regression View

An alternative perspective derives Ensemble++ from a single ridge-regression objective. First, note:

- The *base* parameter μ_t solves the usual *ridge regression* objective $\min_b \sum_{s=1}^t \big[Y_s - \langle b, X_s \rangle \big]^2 + \lambda \|b\|^2$.
- Each column in $\mathbf{A}_t$ can be seen as a *perturbed* ridge solution that includes random offsets $\mathbf{z}_{s,m}$ for each data.

Combining them yields a single objective for all parameters:

$$\min_{b, \{\theta_m\}} \sum_{s=1}^{t} \Big(Y_s - \langle b,\, X_s \rangle \Big)^2 \;+\; \sum_{m=1}^{M} \Big(\mathbf{z}_{s,m} - \langle \theta_m,\, X_s \rangle \Big)^2 \;+\; \lambda \Big(\|b\|^2 + \sum_{m=1}^{M} \|\theta_m\|^2 \Big). \tag{4}$$

The closed-form solution $(b_t^*, \{\theta_{t,m}^*\})$ coincides with the incremental updates in **??** and Equation (3) when we identify

$$\mu_t = b_t^*, \quad \mathbf{A}_{t,m} = \theta_{t,m}^* + \tilde{\theta}_{0,m}, \tag{5}$$

assuming $\mu_0 = 0$, $\Sigma_0 = \frac{1}{\lambda} I$ and $\tilde{\theta}_{0,m} = \frac{1}{\sqrt{M}} \tilde{A}_{0,m}$.

Symmetrized Loss. Finally, from an random linear combination view (c.f. Equation (2)) and Equation (5), define the ensemble prediction function

$$f_\theta^{\text{linear}}(x, \zeta) \;=\; \Big\langle x,\; b \;+\; \sum_{m=1}^{M} \zeta_m \left(\theta_m + \tilde{\theta}_{0,m}\right) \Big\rangle, \tag{6}$$

and let $D = \{(X_s, Y_s, \mathbf{z}_s)\}_{s=1}^t$ with $\mathbf{z}_s = (\mathbf{z}_{s,1}, \ldots, \mathbf{z}_{s,M})$. A *symmetrized* objective $L(\theta; D, f)$ includes both $(+\mathbf{z}_{s,m})$ and $(-\mathbf{z}_{s,m})$ for each data point:

$$\sum_{m=1}^{M} \sum_{s \in D} \sum_{\beta \in \{\pm 1\}} \Big(Y_s + \beta \mathbf{z}_{s,m} - f\big(X_s,\; \beta e_m\big)\Big)^2 \;+\; \lambda \|\theta\|^2. \tag{7}$$

Minimizing Equation (7) recovers the same solution as Equation (4) since the symmetrized slack variable β cancels out the cross-term. This "two-sided" perturbation perspective extends naturally to *Ensemble*++, as shown next.

3.3 Ensemble++ Agent for Nonlinear Bandits

Real-world tasks frequently require *nonlinear* function approximators (e.g., neural networks) for high-dimensional inputs or complex reward structures f^*. *Ensemble*++ retains the same "shared ensemble factor" principle but replaces linear features with a learnable network.

Model Architecture. We generalize Equation (6) by letting $h(x; w)$ be a *neural* feature extractor:

$$f_\theta(x, \zeta) \;=\; \big\langle h(x; w),\, b + \sum_{m=1}^{M} \zeta_m \left(\theta_m + \tilde{\theta}_{0,m}\right) \big\rangle,$$

where $\theta = (w, b, \{\theta_m\})$ are learnable parameters, and $\{\tilde{\theta}_{0,m}\}$ are fixed random "prior" directions. The only difference from Linear Ensemble++ Sampling is that we no longer have a closed-form update for $b, \{\theta_m\}$.

Symmetrized Loss and SGD. Define the same symmetrized objective $L(\theta; D, f_\theta)$ as in Equation (7), except that f_θ is now a neural mapping. At time step t, we store $(X_t, Y_t, \mathbf{z}_t)$ in a FIFO buffer D (capacity C), then run a fixed number G of SGD steps to update θ:

$$\theta \;\leftarrow\; \theta \;-\; \eta \nabla_\theta L\big(\theta; D, f_\theta\big).$$

Algorithm 1 summarizes: by capping C and G, the agent ensures constant-time updates even as t grows. Though we lack a formal regret proof for *nonlinear* rewards, empirical evidence (see Appendix B) shows that Ensemble++ exhibit strong performance in practice, even for complex, high-dimensional reward functions. See implementation details in Appendix C.

Algorithm 1 Ensemble++ Agent

Initialize $\theta = (w, b, \{\theta_m\})$, prior ensemble $\{\theta_{0,m}\}$
Initialize FIFO buffer D of capacity C
for $t = 1$ to T **do**
  Sample $\zeta_t \sim P_\zeta$; $\quad X_t \;=\; \arg\max_{x \in \mathcal{X}_t} f_\theta(x,\, \zeta_t)$.
  Observe reward Y_t; sample $\mathbf{z}_t \sim P_{\mathbf{z}}$
  Add $\big(X_t,\, Y_t,\, \mathbf{z}_t\big)$ to buffer D (pop oldest if $|D| > C$)
  Perform SGD w.r.t. $L\big(\theta; D,\; f_\theta\big)$ up to G steps
end for

4 ENSEMBLE++ IN FOUNDATION MODEL ONLINE DECISION-MAKING

This section demonstrates how *Ensemble++* can be integrated with large *foundation models* (e.g. GPTs) to address real-time decision-making under uncertainty. We focus on the high-stakes domain of **content moderation** on social media platforms, where rare or borderline hateful content arises frequently. By fusing GPT-style feature extraction with Ensemble++, uncertainty-driven sampling selectively allocates human review to ambiguous posts. This yields a scalable, adaptive pipeline that reduces moderator workload while improving overall safety and accuracy.

4.1 GPT-ENSEMBLE++ FOR CONTENT MODERATION

We now introduce **GPT-Ensemble++**, adapting the *Ensemble++* agent (cf. Section 3.3) to text-based moderation scenarios with a foundation model backbone.

Feature Extractor. We define $\phi(x; w)$, mapping a post x into $\mathbb{R}^d$ using a GPT-2 (or Pythia14m) backbone, with w either *frozen* or *partially finetuned*. This captures context and semantic cues.

Ensemble++ Decision Head. For each action a, we define a base parameter $b^a \in \mathbb{R}^d$ and an ensemble factor $\mathbf{A}^a \in \mathbb{R}^{d \times M}$. At each time step, we sample a random index $\zeta \sim P_\zeta \subset \mathbb{R}^M$ (e.g. Gaussian). Then the *action-value* is:

$$f_\theta(x, \zeta)[a] \;=\; \big\langle \phi(x; w),\, b^a \big\rangle \;+\; \big\langle \mathrm{sg}[\phi(x; w)],\, \mathbf{A}^a\, \zeta \big\rangle.$$

Hence, the agent picks $\arg\max_a f_\theta(x, \zeta)[a]$. Drawing ζ each round fosters *randomized* (Thompson-like) exploration around uncertain or borderline posts.

Incremental Updates. If the system chooses `human-review` for a post x_t, we obtain a corrective label y_t (hate vs. free) which implies a reward r_t. As describe in Algorithm 1, we then update $\theta = \{(b^a, \mathbf{A}^a),\ w\}$ using the *symmetrized* objective with bounded gradient steps. This step yields a fast, incremental refinement of the policy, allowing GPT-Ensemble++ to adapt quickly whenever new borderline cases arise in production.

4.2 EXPERIMENTS: HATE-SPEECH DETECTION

Dataset and Setup. We employ a hate-speech dataset[1] of about 135k posts, each assigned a continuous "hate" score. Thresholding at 0.5 yields "hate" vs. "free." At round t, the agent sees x_t (text), chooses **publish** ($A_t = 1$) or **block** ($A_t = 2$), and receives:

$$\begin{cases} +1 & \text{if publishes a free post,} \\ -0.5 & \text{if publishes a hate post,} \\ +0.5 & \text{if blocks any post.} \end{cases}$$

We embed text with GPT-2 or Pythia14m in either frozen or partially finetuned mode, then feed into Ensemble++ or baselines.

Comparative Baselines. We consider:

1. **Greedy**: A single LLM-based classifier with no ensemble factor, i.e. $\mathbf{A}^a = 0$,
2. **Ensemble+** (Osband et al., 2018): multiple ensemble heads jointly upated,
3. **Ensemble++** (ours): separated ensemble updates plus partial or full LLM finetuning.

We also vary **frozen vs. finetuned** embeddings w for GPT-based models.

4.2.1 RESULTS AND ANALYSIS

Uncertainty-Aware Gains. Figure 2 shows that *Ensemble++* significantly outperforms Greedy in cumulative reward, clarifying borderline expressions faster and reducing error variance.

Frozen vs. Finetuned. In Figure 3, together with Ensemble++, full finetuning of GPT-based features yields further gains compared to frozen embeddings. This suggests that *active adaptation* of the LLM backbone is crucial for handling evolving content.

Reduced Human Overhead. Although not depicted, Ensemble++ quickly pinpoints which posts are certain vs. borderline, leading to $\sim 80\%$ fewer "human-review" actions after 10^4 steps compared to naive or deterministic triggers (e.g., Greedy).

[1] https://huggingface.co/datasets/ucberkeley-dlab/measuring-hate-speech

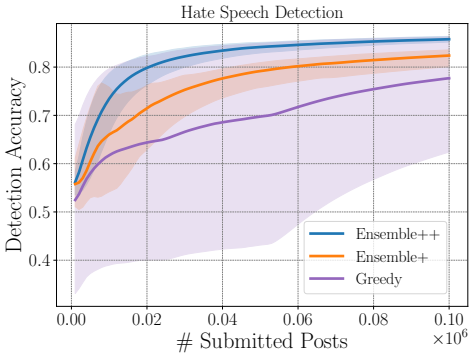
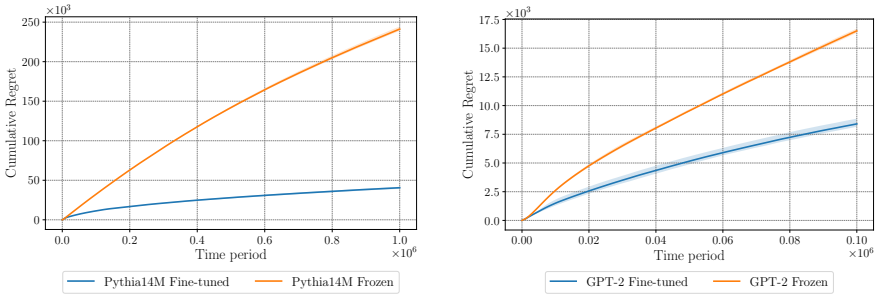


Figure 2: *Detection accuracy* over time in hateful vs. free moderation, averaged across random seeds, as the number of submitted posts increases. **Ensemble++** (blue) outperforms Greedy (purple) and Ensemble+ (orange) with lower variance.



(a) Pythia14m (b) GPT-2

Figure 3: Ablation in hateful-content moderation. (a,b) Fully finetuning yields stronger improvements in uncertain areas than freezing GPT (Pythia14m) backbone.

We show Ensemble++ can be integrated with *foundation models* like GPT-2 for large-scale content moderation—a domain rife with domain shifts, ambiguous inputs, and costly feedback and indicate:

- *Uncertainty quantification*: Ensemble++ better identifies borderline or novel forms of hate speech, enabling more selective human intervention.
- *Incremental adaptation*: The ensemble updates per step boundedly, even with partial finetuning.
- *Reduced moderator workload*: By focusing reviews on genuinely uncertain posts, Ensemble++ drastically cuts human oversight needs.

These results highlight *Ensemble++* as a powerful approach for real-time, uncertain tasks in industrial settings where foundation models alone lack uncertainty-awareness for adaptive exploration.

5 Concluding Remarks

Thompson Sampling's principled treatment of exploration has inspired extensive research into Bayesian methods for sequential decision-making. Yet large-scale and non-conjugate contexts remained a hurdle: exact posteriors are infeasible, and naive ensemble approximations often demand huge ensemble sizes or high retraining cost.

We introduced *Ensemble++*, showing how to:

- maintain a single shared matrix $\mathbf{A}_t$ in the *linear* case with $M = O(d \log T)$,
- achieve incremental $\mathcal{O}(d^2 M)$-time updates that approximate $\Sigma_t^{1/2}$ despite adaptive data,
- unify base and ensemble parameters in a *symmetrized regression objective* that admits a closed-form solution (linear) or an SGD solution (neural).

As a result, *Linear Ensemble++ Sampling* inherits Thompson-like $\sqrt{T}$ regret without the previous M-vs-T scalability conflict. The *neural* extension simply replaces fixed feature mappings with a trainable feature extractor under the same objective, enabling broad applicability in high-dimensional problems. Experiments provide strong empirical performance, superior to alternative ensemble methods, thanks to (i) a modest ensemble dimension; (ii) incremental, real-time updates.

IMPACT STATEMENT

Ethics Statement: This research was conducted in compliance with all applicable ethical guidelines and institutional regulations. Since the study did not involve human participants, animals, or sensitive data, no specific ethical approvals were required. All data used in this research were obtained from publicly available sources, ensuring full transparency and reproducibility of the results.

Reproducibility Statement: Detailed settings for the experiments can be found in Appendices B and H. We conduct the experiments on linear bandits using only CPUs, and the experiments on nonlinear bandits using P40 GPUs, except for those involving GPT-2, which were conducted on V100 GPUs.

For the baselines compared in the experiments, we reimplemented the following methods: *Ensemble+* following the repository https://github.com/google-deepmind/bsuite, *EpiNet* following the repository https://github.com/google-deepmind/neural_testbed. Additionally, we used the source code from the repository https://github.com/devzhk/LMCTS for LMCTS to obtain the credited results.

To reproduce the results of our proposed Ensemble++ agent, please refer to our codebase at https://anonymous.4open.science/r/EnsemblePlus2-1E54.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011a.

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Online least squares estimation with self-normalized processes: An application to bandit problems. *arXiv preprint arXiv:1102.2670*, 2011b.

Marc Abeille and Alessandro Lazaric. Linear thompson sampling revisited. *Electronic Journal of Statistics*, 11(2), 2017.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International conference on machine learning*, pp. 127–135. PMLR, 2013.

Arthur Asuncion, David Newman, et al. Uci machine learning repository, 2007.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

X Corp. The x rules: Safety, privacy, authenticity, and more. https://help.twitter.com/en/rules-and-policies/x-rules, 2024. Accessed: 2024-07-09.

Vikranth Dwaracherla, Xiuyuan Lu, Morteza Ibrahimi, Ian Osband, Zheng Wen, and Benjamin Van Roy. Hypermodels for exploration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryx6WgStPB.

Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5): 10143–10162, 1994.

Geir Evensen. The ensemble kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53:343–367, 2003.

Robert Gorwa, Reuben Binns, and Christian Katzenbach. Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society*, 7(1): 2053951719897945, 2020.

T. H. Gronwall. The gamma function in the integral calculus. *Annals of Mathematics*, 20(2):35–124, 1918. ISSN 0003486X. URL http://www.jstor.org/stable/1967180.

Lawrence Hollom and Julien Portier. Tight lower bounds for anti-concentration of rademacher sums and tomaszewski's counterpart problem. *arXiv preprint arXiv:2306.07811*, 2023.

David Janz, Alexander Litvak, and Csaba Szepesvari. Ensemble sampling for linear bandits: small ensembles suffice. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=SO7fnIFq0o.

Daniel M Kane and Jelani Nelson. Sparser johnson-lindenstrauss transforms. *Journal of the ACM (JACM)*, 61(1):1–23, 2014.

Akshay Krishnamurthy, Keegan Harris, Dylan J. Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context?, 2024.

Branislav Kveton, Csaba Szepesvári, Mohammad Ghavamzadeh, and Craig Boutilier. Perturbed-history exploration in stochastic linear bandits. In *Uncertainty in Artificial Intelligence*, pp. 530–540. PMLR, 2020a.

Branislav Kveton, Manzil Zaheer, Csaba Szepesvari, Lihong Li, Mohammad Ghavamzadeh, and Craig Boutilier. Randomized exploration in generalized linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 2066–2076. PMLR, 2020b.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in neural information processing systems*, 20, 2007.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Yingru Li. Probability tools for sequential random projection, 2024a. URL https://arxiv.org/abs/2402.14026.

Yingru Li. Simple, unified analysis of johnson-lindenstrauss with applications, 2024b. URL https://arxiv.org/abs/2402.10232.

Yingru Li and Zhi-Quan Luo. Prior-dependent analysis of posterior sampling reinforcement learning with function approximation. In *The 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

Yingru Li, Jiawei Xu, Lei Han, and Zhi-Quan Luo. Q-Star Meets Scalable Posterior Sampling: Bridging Theory and Practice via HyperAgent. In *Forty-first International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2024. URL https://arxiv.org/abs/2402.10228.

Xiuyuan Lu and Benjamin Van Roy. Ensemble sampling. *Advances in neural information processing systems*, 30, 2017.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. A holistic approach to undesired content detection in the real world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 15009–15018, 2023.

Meta. Facebook community standards. https://transparency.meta.com/policies/community-standards/, 2024. Accessed: 2024-07-09.

Gergő Nemes. Error bounds and exponential improvements for the asymptotic expansions of the gamma function and its reciprocal. *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, 145(3):571–596, 2015. doi: 10.1017/S0308210513001558.

Ian Osband and Benjamin Van Roy. Bootstrapped thompson sampling and deep exploration. *arXiv preprint arXiv:1507.00300*, 2015.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

Ian Osband, John Aslanides, and Albin Cassirer. Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Ian Osband, Benjamin Van Roy, Daniel J. Russo, and Zheng Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019. URL http://jmlr.org/papers/v20/18-339.html.

Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Xiuyuan Lu, Morteza Ibrahimi, Dieterich Lawson, Botao Hao, Brendan O'Donoghue, and Benjamin Van Roy. The neural testbed: Evaluating joint predictions. *Advances in Neural Information Processing Systems*, 35:12554–12565, 2022.

Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Epistemic neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=dZqcC1qCmB.

Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. Approximate thompson sampling via epistemic neural networks. *arXiv preprint arXiv:2302.09205*, 2023b.

George Papandreou and Alan L Yuille. Gaussian sampling by local perturbations. *Advances in Neural Information Processing Systems*, 23, 2010.

Chao Qin, Zheng Wen, Xiuyuan Lu, and Benjamin Van Roy. An analysis of ensemble sampling. *Advances in Neural Information Processing Systems*, 35:21602–21614, 2022.

Reddit. Automoderator guide. https://www.reddit.com/r/reddit.com/wiki/automoderator/, 2024. Accessed: 2024-07-09.

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.

Sarah T Roberts. *Behind the screen*. Yale University Press, 2019.

Donald B Rubin. The bayesian bootstrap. *The annals of statistics*, pp. 130–134, 1981.

Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. *Operations Research*, 66(1):230–252, 2018.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

Maciej Skorski. Bernstein-type bounds for beta distribution. *Modern Stochastics: Theory and Applications*, 10(2):211–228, 2023.

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

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019. doi: 10.1017/9781108627771.

Chih-Chun Wang, Sanjeev R Kulkarni, and H Vincent Poor. Bandit problems with side observations. *IEEE Transactions on Automatic Control*, 50(3):338–355, 2005.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688. Citeseer, 2011.

Lilian Weng, Vik Goel, and Andrea Vallone. Using gpt-4 for content moderation. August 2023. URL https://openai.com/index/using-gpt-4-for-content-moderation/. OpenAI.

Pan Xu, Hongkai Zheng, Eric V Mazumdar, Kamyar Azizzadenesheli, and Animashree Anandkumar. Langevin monte carlo for contextual bandits. In *International Conference on Machine Learning*, pp. 24830–24850. PMLR, 2022.

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pp. 11492–11502. PMLR, 2020.

A Additional Discussions on Related Works

This appendix provides further background and motivation for the techniques discussed in the main text, focusing on Thompson Sampling and its limitations, local perturbation for Gaussian posteriors, and ensemble-based methods. We also compare Ensemble++ with advanced neural architectures such as Ensemble+ (Osband et al., 2018; 2019) and EpiNet (Osband et al., 2023b).

A.1 Sequential Decision Making under Uncertainty

We consider a sequential decision-making problem over a discrete time horizon T. At each time step t:

- The agent observes a *decision set* $\mathcal{X}_t \subseteq \mathcal{X}$, which may change over time (e.g., due to evolving *context* or the appearance of new candidate actions).
- The agent selects an action $X_t \in \mathcal{X}_t$, based on its past experience $\mathcal{H}_t = \{\mathcal{X}_1, X_1, Y_1, \ldots, \mathcal{X}_{t-1}, X_{t-1}, Y_{t-1}, \mathcal{X}_t\}$.
- It then receives a noisy reward $Y_t = f^*(X_t) + \epsilon_t$, where f^* is an unknown reward function and ϵ_t is noise.

The agent's *cumulative regret* measures how much reward is lost by not always picking the best available action:

$$R(T) \;=\; \sum_{t=1}^{T} \Big[\max_{x \in \mathcal{X}_t} f^*(x) \;-\; f^*(X_t) \Big]. \tag{8}$$

A key challenge is *learning* unknown reward function f^* (exploration) while simultaneously *selecting* good actions from $\mathcal{X}_t$ (exploitation) in T time periods.

A.2 Thompson Sampling (TS)

Thompson Sampling is a Bayesian approach for balancing exploration and exploitation in bandit or sequential decision-making problems (Thompson, 1933; Russo et al., 2018; Li & Luo, 2024). It maintains a posterior distribution over unknown parameters (or functions) and selects actions by sampling from this posterior.

Methodology. At each time step t, with history $\mathcal{H}_t$, TS does:

1. **Sample a model:** Draw a parameter $\theta_t \sim P(\theta \mid \mathcal{H}_t)$.
2. **Select action:** $X_t = \arg\max_{x \in \mathcal{X}_t} f_{\theta_t}(x)$, where $f_\theta(\cdot)$ represents the expected reward under model θ.
3. **Observe reward:** Receive Y_t.
4. **Update posterior:** Incorporate (X_t, Y_t) into the posterior $P(\theta \mid \mathcal{H}_{t+1})$.

Because TS samples from a posterior that encodes the agent's uncertainty, it naturally allocates exploration to regions (actions) that are less well understood, while exploiting current knowledge of high-reward actions.

Conjugate Settings. In special "conjugate" scenarios, posterior updates are tractable:

- **Beta-Bernoulli Bandits:** A Beta prior for each arm remains Beta after seeing Bernoulli rewards.
- **Linear-Gaussian Bandits:** With Gaussian priors and Gaussian noise, the posterior remains Gaussian with updated mean and covariance.

Here, each TS update is fast. However, in high-dimensional or non-conjugate (e.g., neural network) reward models, exact posterior inference becomes intractable.

A.3 Challenges in Scaling Thompson Sampling

While Thompson Sampling has strong theoretical properties (e.g., near-optimal regret in finite action or linear bandit scenarios), it faces two main challenges when extended to large-scale or complex environments:

Non-Conjugate Models. Many real-world applications (e.g., deep neural networks, highly structured rewards) do not admit closed-form updates. Direct posterior sampling then requires approximate Bayesian techniques that can be expensive or unreliable.

A.3.1 APPROXIMATE BAYESIAN INFERENCE

Motivation. To preserve the essence of TS (sampling from a distribution over plausible models), various approximate inference methods aim to produce posterior-like samples at each step. However, many of these approaches suffer from high computational overhead.

Prominent Approximate Methods.

- **Laplace Approximation** (MacKay, 1992): Approximates the posterior around its mode with a Gaussian. Scales poorly if the parameter dimension is large.
- **Variational Inference (VI)** (Blei et al., 2017): Uses a parametric distribution and minimizes a KL divergence. Can handle larger dimensions than Laplace but introduces biases based on the chosen family.
- **MCMC Methods** (Welling & Teh, 2011): Iteratively generate samples from the true posterior. Accurate but often expensive in high dimensions or real-time tasks.
- **Langevin Monte Carlo (LMC)** (Xu et al., 2022): A gradient-based MCMC approach adding noise to gradient steps. Its iterative nature can be costly in long horizons. More precisely, the per-step computation complexity grows linearly with the size of the history interaction data set.

Key Limitations.

- *Biased Uncertainty*: Approximate posteriors may misestimate uncertainty, harming TS's exploration.
- *Iterative Overheads*: Repeated passes over the entire history per step become impractical as T grows large.
- *Scalability*: Quadratic/cubic scaling in model dimension is prohibitive for large networks.

Thus, while approximate methods broaden TS's applicability, their computational or memory costs remain problematic in large-scale, non-conjugate settings.

A.4 GAUSSIAN SAMPLING VIA LOCAL PERTURBATION

An alternative for *linear–Gaussian* environments is **local perturbation** (Papandreou & Yuille, 2010), which incrementally updates posterior samples in $\mathcal{O}(d^2)$ per step—avoiding $\mathcal{O}(d^3)$ matrix factorizations.

Idea. Suppose $\theta^* \sim \mathcal{N}(\mu_0, \Sigma_0)$ and observations

$$Y_s \;=\; X_s^\top \theta^* + \omega_s^*, \quad \omega_s^* \sim \mathcal{N}(0,1).$$

Then the posterior after t observations is $\mathcal{N}(\mu_t, \Sigma_t)$. Rather than factor Σ_t at each step, local perturbation maintains

$$\tilde{A}_t \;=\; \Sigma_t\Big(\Sigma_0^{-1}\tilde{A}_0 \;+\; \sum_{s=1}^{t} X_s\, Z_s\Big),$$

with $\tilde{A}_0 \sim \mathcal{N}(0, \Sigma_0)$ and $Z_s \sim \mathcal{N}(0,1)$. Under a *fixed* (non-adaptive) design $\{X_s\}$, $\tilde{A}_t \sim \mathcal{N}(0, \Sigma_t)$, hence

$$\tilde{\theta}_t = \mu_t + \tilde{A}_t \quad \text{is an exact draw from } \mathcal{N}(\mu_t, \Sigma_t).$$

Both μ_t and $\tilde{A}_t$ update incrementally in $\mathcal{O}(d^2)$.

A.4.1 DISTRIBUTION MATCHING PROOF OUTLINE

For completeness, we briefly sketch why local perturbation yields an *exact* posterior draw in the non-adaptive case.

Let $D_t = \{(X_s, Y_s)\}_{s=1}^t$. Then:

$$\mathbb{E}[\tilde{A}_t \mid D_t] = 0, \quad \mathrm{Cov}(\tilde{A}_t \mid D_t) = \Sigma_t,$$

implying $\mu_t + \tilde{A}_t \sim \mathcal{N}(\mu_t, \Sigma_t)$. The key steps:

Mean argument. For each s, $Z_s \sim N(0,1)$ is independent of D_t, so $\mathbb{E}[Z_s|D_t] = 0$. Similarly, $\tilde{A}_0 \sim N(0, \Sigma_0)$ is independent of D_t and $\mathbb{E}[\tilde{A}_0 \mid D_t] = 0$. Hence,

$$\mathbb{E}[\tilde{A}_t \mid D_t] \;=\; \Sigma_t\Big(\Sigma_0^{-1}\mathbb{E}[\tilde{A}_0 \mid D_t] + \sum_{s=1}^{t} X_s\,\mathbb{E}[Z_s \mid D_t]\Big) = 0.$$

Covariance argument. Because $Z_s \sim N(0,1)$ i.i.d. and $\tilde{A}_0 \sim N(0, \Sigma_0)$,

$$\begin{aligned}
\mathrm{Cov}[\tilde{A}_t \mid D_t] &= \Sigma_t\Big(\Sigma_0^{-1}\,\mathrm{Cov}[\tilde{A}_0 \mid D_t]\,\Sigma_0^{-1} + \sum_{s=1}^{t} X_s X_s^\top \mathbb{E}[Z_s^2]\Big)\Sigma_t \\
&= \Sigma_t\Big(\Sigma_0^{-1}\,\Sigma_0\,\Sigma_0^{-1} + \sum_{s=1}^{t} X_s X_s^\top\Big)\Sigma_t \\
&= \Sigma_t\big(\Sigma_t^{-1}\big)\Sigma_t = \Sigma_t.
\end{aligned}$$

Thus $\tilde{A}_t \sim \mathcal{N}(0, \Sigma_t)$ *conditionally on* D_t. Therefore, $\tilde{\theta}_t := \mu_t + \tilde{A}_t$ *has mean* μ_t *and covariance* Σ_t and matches exactly $\mathcal{N}(\mu_t, \Sigma_t)$.

A.5 Recursive Randomized Least Squares (RRLS)

Motivation. Motivated by bounded per-step computation requirement, one could attempt to update the parameter vector θ_t in an incremental, *recursive* manner:

$$\theta_t \;=\; \Sigma_t\Big(\Sigma_{t-1}^{-1}\theta_{t-1} + X_t\,\big(Y_t + Z_t\big)\Big), \tag{9}$$

where $Z_t \sim \mathcal{N}(0,1)$ is a fresh random perturbation at each time t. This yields the *Recursive RLS* (RRLS) algorithm:

Algorithm 2 Recursive Randomized Least Squares (RRLS)

1: Initialize $\theta_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$
2: **for** $t = 1$ to T **do**
3: $\quad X_t = \arg\max_{x \in \mathcal{X}_t} \langle \theta_{t-1}, x \rangle$
4: $\quad$ Observe Y_t
5: $\quad$ Sample $Z_t \sim \mathcal{N}(0,1)$
6: $\quad$ Update θ_t via equation 9
7: **end for**

Sequential Dependency However, RRLS introduces *sequential dependency* because the action X_t chosen at time t depends on the previous parameter estimate θ_{t-1}, which itself depends on all past perturbations Z_s and past actions X_s for $s < t$. Due to this sequential dependency, the conditional expectation and covariance of θ_t no longer match those of the posterior distribution $\theta^* \mid D_t$. This is because when conditioning on D_t, the perturbations $Z_1, \ldots, Z_t$ are no longer independent and identically distributed (i.i.d.) as Normal random variables. This results in biased estimates and ineffective exploration, giving linear regret in some scenarios. This sequential dependency is illustrated in Figure 4.

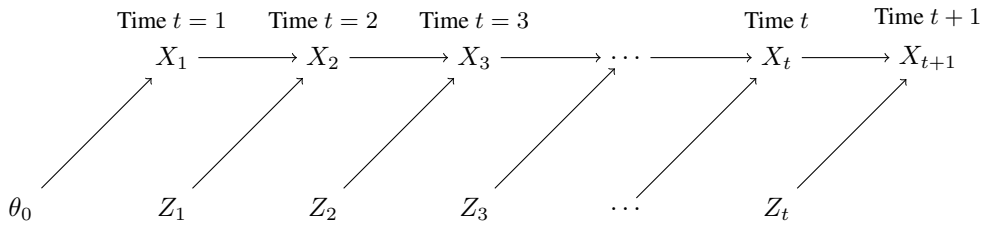


Figure 4: Sequential dependence due to the interplay between recursive updates and sequential decision-making. $Z_1, \ldots, Z_t$ are no longer independent and identically distributed (i.i.d.) when conditioned on the data X_{t+1}.

One potential workaround is to *resample* fresh random perturbations $\{Z_s\}_{s \le t}$ and re-fit from scratch each step (e.g., storing all historical data) to restore independence, but this is computationally and memory expensive in practice (Osband et al., 2019; Kveton et al., 2020a) and defeats the purpose of a cheap incremental method.

The next subsection describes an approach—*ensemble sampling*—that mitigates sequential dependency via multiple parallel parameter vectors, each incrementally updated.

A.6 Ensemble Sampling (ES)

Principle. Ensemble Sampling (Osband & Van Roy, 2015; Osband et al., 2016; Lu & Van Roy, 2017) keeps M independent parameter vectors (or models). At time t, it uniformly picks one model m_t to decide X_t and then updates all M models in a incremental, recursive manner. Intuitively, if these M vectors approximate M draws from the posterior, the overall policy resembles Thompson Sampling.

Algorithm Outline.

- *Initialization:* $\theta_{0,m} \sim \mathcal{N}(\mu_0, \Sigma_0)$ for $m = 1, \ldots, M$.
- *Action Selection:*

$$m_t \sim \text{Uniform}\{1, \ldots, M\}, \qquad X_t = \arg\max_{x \in \mathcal{X}_t} \langle \theta_{t-1,m_t}, x \rangle.$$

- *Model Updates:* Each $\theta_{t,m}$ is updated in an RRLS-like manner, but with fresh noise $Z_{t,m}$.

This balances memory usage ($M \ll T$) against sequential dependency.

Table 1: Comparison of methods for addressing sequential dependency.

Method	**Computation per Step**	**Memory Usage**	**Sequential Dependency**
RLS ($M = T$)	$O(T)$	High	None
ES ($M \ll T$)	$O(M)$	Moderate	Reduced
RRLS ($M = 1$)	$O(1)$	Low	High

Empirical Trade-offs. - *If $M = T$,* and each model is selected exactly once at time t (i.e., $m_t = t$), the Ensemble Sampling method becomes equivalent to original randomized least squares (RLS) with perturbations resampling and model retraining at each time step. This approach eliminates sequential dependency entirely but requires huge computation and memory overhead. - *If $M = 1$,* it degenerates to RRLS with minimal memory but strong sequential dependency. Hence, by choosing $M \ll T$, one often obtains good practical performance (Fig. 5). This suggest, empirically, ES with moderate M can achieve performance comparable to TS while only paying a factor of M overhead in memory and a moderate per-step cost.

Theoretical Limitations Qin et al. (2022) provide a rigorous regret analysis for linear ensemble sampling that could match the regret order of exact Thompson sampling but require $M = O(|\mathcal{X}|T)$

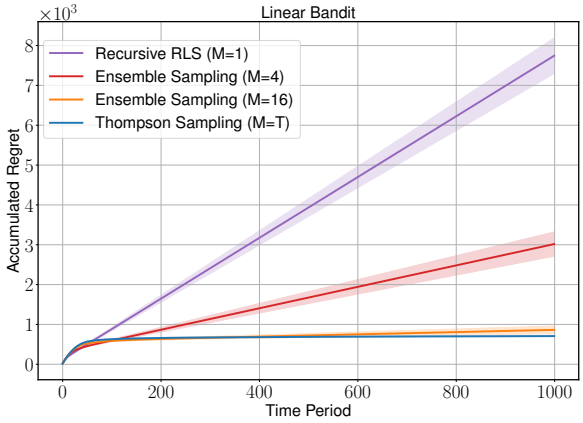


Figure 5: Ensemble Sampling (ES) with moderate M achieves near-TS performance. Setup: $|\mathcal{X}| = 10{,}000$ and dimension $d = 50$.

to maintain $\sqrt{T}$ scaling in Bayesian regret, a major barrier in practical large-scale problems. This suggests *naively* we might need an ensemble size that scales linearly with T or $|\mathcal{X}|$—infeasible for large action sets and long horizons tasks, contradicting with the empirical findings of a moderate size of ensembles.

Remark 1. *Qin et al. (2022) consider a d-dimensional linear bandit problem with an action set $\mathcal{X}$. When the true parameter follows a standard normal distribution $\theta^* \sim \mathcal{N}(0, I_d)$, the Bayesian regret is bounded by:*

$$BR(T) \leq C\sqrt{dT \log |\mathcal{X}|} + CT\sqrt{\frac{|\mathcal{X}| \log(MT)}{M}}(d \wedge \log |\mathcal{X}|)$$

where $C > 0$ is a universal constant. This bound has two significant limitations:

1. *To achieve the desired $\sqrt{T}$ scaling in Bayesian regret (ignoring constant and logarithmic factors), the ensemble size M must grow linearly with the time horizon T. This requirement undermines the computational efficiency that ensemble sampling aims to achieve.*
2. *To maintain a logarithmic dependence on $|\mathcal{X}|$ in the bound, the ensemble size M must scale linearly with the number of actions $|\mathcal{X}|$.*

These limitations become particularly problematic when dealing with compact action spaces. For instance, consider a bandit problem where $\mathcal{X} = \mathbb{B}_2^d$ (the d-dimensional unit ball). To achieve a small discretization error, we need approximately 2^{d-1} discrete actions. Consequently, following Qin et al.'s bound, the required ensemble size M would grow exponentially with dimension d.

Ensemble Sampling Beyond Linear Models For a general function class $\mathcal{F}$, Ensemble Sampling can be extended to approximate the posterior distribution of the optimal function $f^* \in \mathcal{F}$, e.g. *Bootstrapped Ensemble* (Osband et al., 2016) and *Ensemble+* (Osband et al., 2018; 2019). The agent maintains M models, each representing a hypothesis about f^* based on historical data. At each time step t, the agent samples a model m_t uniformly from $\{1, 2, \ldots, M\}$ and selects an action: $X_t = \arg\max_{x \in \mathcal{X}_t} f_{\theta_{t,m_t}}(x)$, where $f_{\theta_{t,m_t}}(x)$ is the prediction of ensemble member m_t for action x. After observing the reward Y_t, each ensemble member m updates its parameters $\theta_{t+1,m}$ by performing stochastic gradient descent on the loss (Equation (10)) starting from the previous iterate $\theta_{t,m}$:

$$L_m(\theta; D) = \sum_{s=1}^{t} \left(Y_s + Z_{s,m} - f_\theta(A_s)\right)^2 + \Psi(\theta) \tag{10}$$

where $D = \mathcal{H}_t$ and $Z_{s,m}$ are independent random perturbations added to encourage diversity among ensemble members, and $\Psi(\theta_{t+1,m})$ is a regularization term. This perturbed training procedure ensures that each ensemble member captures different aspects of the uncertainty in f^*, representing

different plausible hypotheses consistent with the history. The random perturbations $Z_{s,m}$ are independent across time index s and model index m. Once realized, $Z_{s,m}$ are fixed throughout the rest of the training, enabling incremental updates for real-time adaptation. This is a key computational feature compared to methods like Randomized Least Squares (RLS) or Perturbed History Exploration (PHE). In RLS and PHE, fresh independent perturbations for all historical data are introduced at each time t, and the model requires full retraining from scratch to ensure diverse exploration of different plausible hypotheses. Yet, it is important to note that the theoretical analysis of Ensemble Sampling beyond linear models remains an open research question.

A.7 Concluding Remarks and Forward Outlook

Local perturbation methods (RLS, RRLS) and ensemble-based approximations collectively aim to solve large-scale or non-conjugate posterior sampling in an *online* manner. Yet:

- **Recursive RLS (RRLS)** is cheap to update but suffers from *sequential dependency* bias, often giving linear regret in adaptive settings.
- **Ensemble Sampling** lessens sequential dependency empirically with moderate size of ensembles. How, current theory suggest ensemble sampling may require $M \propto T$ or $|\mathcal{X}|$ in worst-case analyses, which is computationally or memory-intensive. Moreover, maintaining M independent neural network ensembles is also computationally prohibitive for large models, even with moderate size $M = 10\ 100$.

We propose Ensemble++, which addresses these drawbacks by maintaining a single shared factor for covariance approximation, with incremental updates in $\mathcal{O}(d^2\, M)$ and a rigorous proof that $M \approx d \log T$ suffices to achieve near-optimal regret that matches exact Thompson sampling. Before concluding, we briefly compare with broader ensemble-based research, including Ensemble+ (Osband et al., 2018; 2019) and EpiNet (Osband et al., 2023a).

Future directions. Future directions of this work focus on providing theoretical understanding for the neural extension, integrating more advanced representation learning, and expanding these techniques to full reinforcement learning settings with large state-action spaces. By addressing the interplay between Bayesian exploration and scalable computation, Ensemble++ agent opens the door for more effective online decision-making in real-world systems with foundation models. Additional research directions include extending Ensemble++ to handle multimedia content (e.g., images, videos) for more comprehensive moderation, investigating its adaptability to adversarial attacks or adversarial examples in moderation tasks, and integrating it with scalable human-in-the-loop systems to further reduce human moderation costs.

A.8 Ensemble Methods in Broader Context

History of Ensemble Approaches. Ensemble methods date back to the *Ensemble Kalman Filter* (Evensen, 1994; 2003) or Bayesian bootstrap (Rubin, 1981). In modern literature, *Bootstrapped Ensemble* (Osband & Van Roy, 2015; Lu & Van Roy, 2017) introduced multiple models updated with random perturbations or "bootstrap" samples of data. *Ensemble+* (Osband et al., 2018; 2019) introduced the randomized prior ensembles to enhance the exploration efficiency.

Hypernetworks and EpiNet. Some architectures, like *Hypermodels* (Dwaracherla et al., 2020) or *Epistemic Neural Networks (EpiNet)* (Osband et al., 2023a), treat the ensemble index or random seed as additional network inputs, effectively learning a mapping from random "epistemic index" to parameter space. Although conceptually appealing, they typically lack any rigorous understanding and proven regret bounds and may suffer from large parameter counts, as we discuss next.

A.8.1 Detailed Comparison with EpiNet and Ensemble+

EpiNet Overview. EpiNet is designed to estimate epistemic uncertainty in neural networks by injecting an "epistemic index" $z \in \mathbb{R}^M$ into an MLP layer. Its final output is a combination of:

- A *base* prediction $\mu_\zeta(x)$ on the raw input x,
- An *epinet* MLP $\sigma_\eta^L([x, \tilde{x}, z])$ that processes the concatenation of raw input x, the hidden representation $\tilde{x} \in \mathbb{R}^D$ of x and the random index z,

- A *fixed prior* $\sigma^P(x, z)$, typically a collection of M small MLPs with raw input x as the input, each producing a per-class offset and combined with random z as the output.

Hence, the final model is

$$f_{\text{EpiNet}}(x, z) \;=\; \mu_\zeta(x) \;+\; \sigma_\eta^L\big([x, \tilde{x}, z]\big) \;+\; \sigma^P(x, z).$$

This architecture can learn uncertainty-aware predictions but often suffers from a large parameter footprint (due to multiple MLPs) and lacks any proven regret guarantees in bandit settings. Additionally, because both the *epinet* MLP and the *fixed prior* must take the raw input x, it is challenging to apply EpiNet to more complex networks such as Transformers. Therefore, we do not compare EpiNet in the Hate Speech Detection task.

Ensemble+ Overview. Ensemble+ extends *ensemble sampling* to deep neural networks using a *randomized prior* approach (Osband et al., 2018; 2019). Concretely:

- A *shared* feature extractor processes inputs x into some hidden representation $\tilde{x}$. There are M heads $\{\theta_m\}$, each a simple linear layer that predicts the reward from $\tilde{x}$.
- Additionally, a *fixed prior network* is maintained as a separate feature extractor: $x \mapsto \hat{x}$. Also, there are M unique random prior heads that fixed after random initialized, each predicting the additive prior reward from $\hat{x}$

This design helps capture model uncertainty by mixing learned features with a distinct randomized prior in each ensemble head. However, as with EpiNet, Ensemble+ can become large in parameter count (due to separate prior modules) and currently lacks theoretical regret bounds in deep or high-dimensional bandits.

Parameter Counts. We compare the number of parameters of each method. Assuming the number of parameters of the hidden feature extractor is H, we analyze how many additional parameters each method allocates beyond a single hidden feature extractor network.

- **EpiNet:**
  - The *epinet* MLP has hidden layers that receive $[x, \tilde{x}, z] \in \mathbb{R}^{d+D+M}$ as input and output $\mathbb{R}^{M\times C}$ (for C classes or outputs). Following Osband et al. (2023b;a), we use 2-layer MLPs with 15 units and bias to construct this epinet MLP. Therefore, we count the parameters of this part as:

$$\begin{aligned}
&(d + D + M + 1) \times 15 + (15 + 1) \times 15 + (15 + 1) \times (M + C)\\
&= 15(d + D + M + 1) + 16 \times 15 + 16 \times (M + C)\\
&= 15d + 15D + 31M + 15 + 240 + 16C\\
&= 15d + 15D + 31M + 255 + 16C.
\end{aligned}$$

  - The fixed prior σ^P is composed of M small MLPs, each adding parameters. Following Osband et al. (2023b;a), we use 2-layer MLPs with 5 units and bias to construct this prior network. It takes the raw input $x \in \mathbb{R}^d$ and each MLP outputs $\mathbb{R}^C$. Therefore, we count the parameters of this part as:

$$M\times\big((d{+}1)\times5{+}(5{+}1)\times5{+}(5{+}1)\times C\big) = M\times\big(5(d{+}1){+}30{+}6C\big) = M\times(5d{+}5{+}30{+}6C) = M\times(5d{+}35{+}6C).$$

  - Together, EpiNet can have a large overhead as M small prior MLPs or the epinet's hidden size grow. We can calculate the total parameters as:

$$H + 15d + 15D + 31M + 255 + 16C + M \times (5d + 35 + 6C).$$

- **Ensemble+:**
  - M *linear* heads, each taking the hidden representation $\tilde{x} \in \mathbb{R}^D$ as input, produce the main ensemble predictions $\mathbb{R}^C$, and each head has the same random prior network. Therefore, we count the parameters of this part as:

$$2 \times M \times ((D + 1) \times C) = 2MDC + 2MC.$$

  - A *separate* feature extractor for the M linear prior network heads to form the prior offset. Therefore, we count the parameters of this part as H.

- This leads to approximately $2M$ last-layer transforms (main + prior), plus the potential duplication of feature extractors. We can calculate the total parameters as:

$$2H + 2 \times M \times ((D+1) \times C) = 2H + 2MDC + 2MC.$$

- **Ensemble++:**
  - There are M *linear* heads without bias for the main ensemble for uncertainty estimation, each mapping $\mathbb{R}^D \to \mathbb{R}^C$ and equipped with the same prior networks. Therefore, we calculate the parameters of this part as:

$$2 \times M \times D \times C.$$

  - One more *base* linear head with bias to estimate the mean. The parameters of this part are $(D+1) \times C$.
  - In total, this results in $(2M+1)$ linear layers of dimension $\mathbb{R}^D \to \mathbb{R}^C$, but each is relatively lightweight. We can calculate the total parameters as:

$$H + 2 \times M \times D \times C + (D+1) \times C = H + (2M+1)DC + C.$$

Computational Efficiency.

- **EpiNet:** Concatenates $[x, \tilde{x}, z]$ of dimension $(d + D + M)$, driving up the input size for the epinet MLP. The fixed prior σ^P also has multiple small MLPs. Training/inference cost grows significantly with M.
- **Ensemble+**: Combines a main network and a separate prior network, each with M linear heads. While each head is relatively cheap, maintaining two feature extractors can be more expensive than Ensemble++'s single shared representation.
- **Ensemble++:** Each ensemble head is just a $\mathbb{R}^D \to \mathbb{R}^C$ linear map, combined additively with a base head. Training/inference overhead remains modest, as backprop only flows through linear heads plus one shared feature extractor. The stop-gradient trick can further reduce overhead.

Practical Implications. Empirical studies (Li et al., 2024) show that EpiNet's parameter overhead often slows training and can degrade exploration. Likewise, Ensemble+ can be parameter-heavy if the prior network is large or if M grows. By contrast, Ensemble++ uses a single shared representation with relatively simple linear heads (for both ensemble and prior), yielding a smaller parameter footprint and faster training. Crucially, *Ensemble++* also provides a theoretical foundation guaranteeing near-optimal linear-bandit regret with $M = \widetilde{O}(d \log T)$, whereas EpiNet and Ensemble+ currently lack proven regret bounds.

Conclusion. In summary, EpiNet and Ensemble+ push ensemble-based methods toward richer neural function approximation but face large parameter counts and no *a priori* theoretical guarantees. Ensemble++ uses lightweight linear heads on top of a shared feature extractor—much more efficient in large-scale or real-time settings—and *does* come with rigorous regret analyses for the linear bandit case. Extending those theoretical insights to deep bandits is an ongoing research direction, but empirical results (§B) show strong performance of *Ensemble++* relative to EpiNet and Ensemble+.

B EXPERIMENTS

In this section, we investigate the efficiency and scalability of Ensemble++ in varying contextual bandit as introduced in Appendix A.1. To fully support the theoretical insights, we first consider linear bandit environments.

B.1 EMPIRICAL STUDY ON LINEAR ENSEMBLE++ SAMPLING

We construct the *Finite-action Linear Bandit* environment guided by prior research (Russo & Van Roy, 2018). In this task, we construct the finite decision set $\mathcal{X}$ by uniformly sampling from the range $[-1/\sqrt{5}, 1/\sqrt{5}]^d$ where d is the ambient dimension of the linear reward function perturbed by an additive Gaussian noise term. We provide a detailed implementation of this task in Appendix H.1.

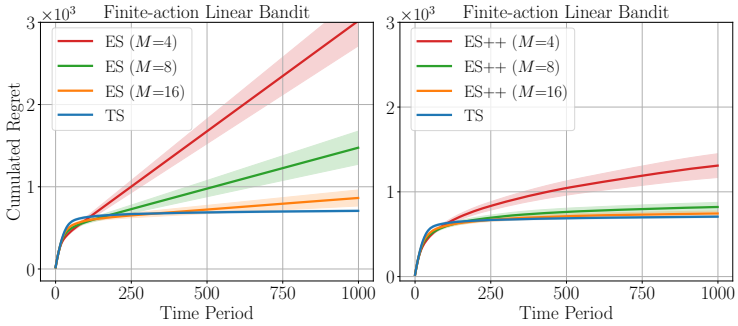


Figure 6: Comparison results in Finite-action Linear Bandit with $d = 50$ and $|\mathcal{X}| = 10,000$. We use ES to refer to Linear Ensemble Sampling, ES++ to refer to Linear Ensemble++ Sampling, and TS to refer to Thompson Sampling for clarity.

Advantage over Ensemble Sampling. We consider a special case of Linear Ensemble++ Sampling with a coordinate reference distribution, which essentially performs uniform sampling among symmetrized ensemble members, similar to vanilla Linear Ensemble Sampling (Lu & Van Roy, 2017). We compare the regret of Linear Ensemble++ Sampling with Linear Ensemble Sampling across varying ensemble sizes M in Figure 6. For a fair comparison, we use the same spherical perturbation distribution in both methods. The results suggest that Linear Ensemble++ Sampling with a Gaussian reference distribution significantly outperforms Linear Ensemble Sampling across varying ensemble sizes M. Notably, Linear Ensemble++ Sampling can nearly match the performance of TS with $M = 8$, saving $2\times$ computation cost compared to Linear Ensemble Sampling.

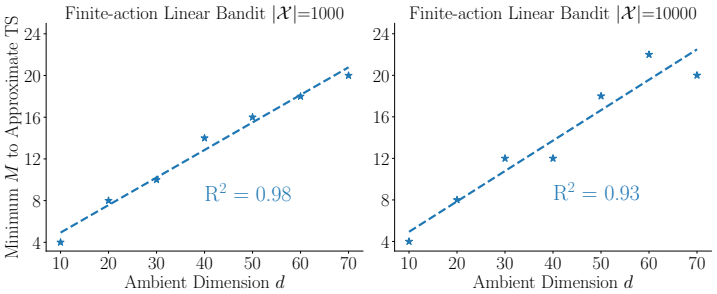


Figure 7: Minimum ensemble size M required to match TS.

Optimal Ensemble Size Scaling. To justify our theoretical prediction of $M = O(d \log T)$, we investigate the minimal ensemble size M required to match the performance of TS. We compute the minimal M using criterion: $M = \min\left\{M : \frac{|\text{Regret}(\textit{Ensemble++}(M),T) - \text{Regret}(\textit{TS},T)|}{T} \leq 0.02\right\}$ and evaluate Linear Ensemble++ Sampling across varying decision set sizes $|\mathcal{X}|$ and ambient dimensions d. As shown in Figure 7, the minimal M exhibits a *linear* relationship with d and nearly remains *unaffected* by $|\mathcal{X}|$.

B.2 Enmseble++ for Nonlinear Bandits

To evaluate Ensemble++ (c.f. Algorithm 1), we consider several nonlinear contextual bandit environments: (1) *Quadratic Bandit*: Adapted from Zhou et al. (2020), the reward function is expressed as $f(x) = 10^{-2}(x^\top \Theta \Theta^\top x)$. Here, $x \in \mathbb{R}^d$ represents the action feature, while $\Theta \in \mathbb{R}^{d \times d}$ is a matrix filled with random variables from $\mathcal{N}(0, 1)$. (2) *Neural Bandit*: This is a binary classification problem adapted from Osband et al. (2022; 2023a). We use 2-layer MLPs with 50 units and ReLU activations to build the neural network with two logit outputs. The Bernoulli reward $r \in \{0, 1\}$ is sampled according to the probabilities obtained from applying softmax to the logits. (3) *UCI Shuttle*: Following prior works Riquelme et al. (2018); Kveton et al. (2020b), we build contextual bandits with N-class classification using the UCI Shuttle dataset Asuncion et al. (2007). (4) *Online*

Hate Speech Detection: We leverage a language dataset[2] to build this task. The agent must decide whether to publish or block content. Blocking any content yields a reward of 0.5. Publishing "free" content earns a reward of 1, while publishing "hate" content incurs a penalty of -0.5.

A detailed description of these nonlinear bandits is provided in Appendix H.2. For all algorithms, we apply 2-layer MLPs with 64 units as the hidden network backbone in the first three tasks, and GPT-2[3] in the last one. Detailed implementation for each algorithms can be found in Appendix A.8.1.

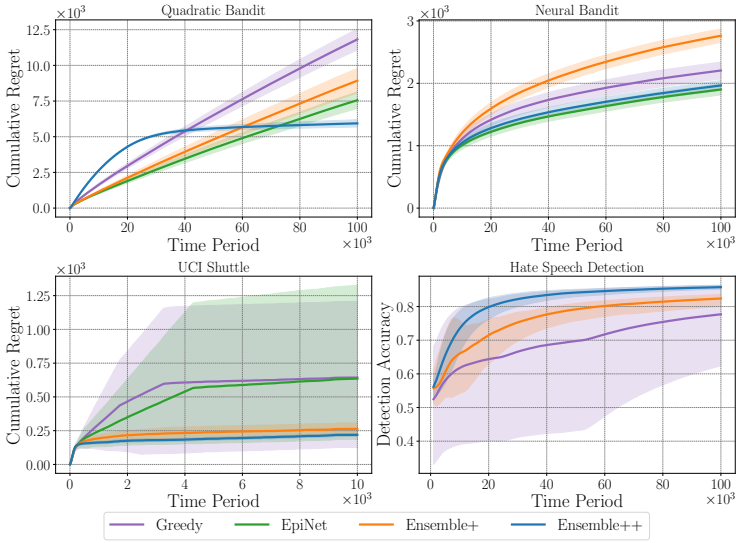
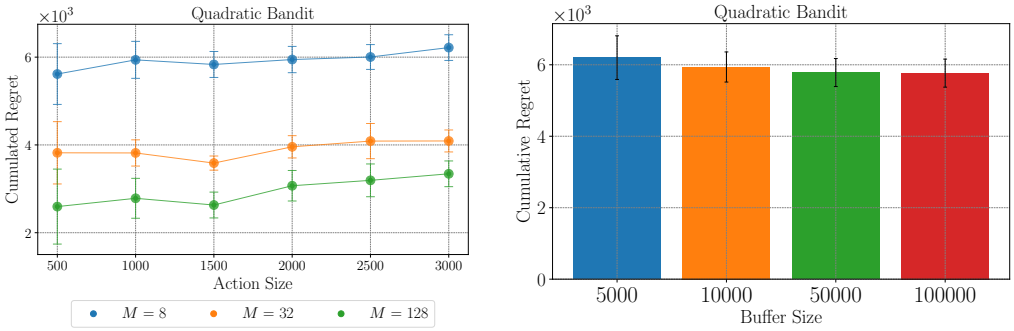


Figure 8: Comparison results across various nonlinear bandits.

Comparison Results. We consider Ensemble+ (Osband et al., 2018) and EpiNet (Osband et al., 2023a) as baselines and include Greedy to demonstrate the exploration requirements of each task. The comparison results in Figure 8 demonstrate that Ensemble++ consistently achieves sublinear regret and higher accuracy. Notably, in the Quadratic Bandit, Ensemble++ achieves fast convergence while other baselines still exhibit linear regret. In the Hate Speech Detection task, Ensemble++ outperforms Ensemble+ by 5%, underscoring its scalability in dealing with more complex networks, such as Transformers. Additionally, the framework of the Hate Speech Detection task can be extended to a wide range of applications, from recommendation systems to online content moderation, as discussed in Section 4, demonstrating the promising utility of Ensemble++ in real-world applications. Due to the implementation details of EpiNet discussed in the Appendix A.8.1, we are unable to apply it to the Hate Speech Detection task. Furthermore, we compare Ensemble++ to LMCTS (Xu et al., 2022), on extensive nonlinear bandits in Appendix H.2, where Ensemble++ consistently achieves sublinear, smaller regret with bounded and lower per-step computation costs.



(a) (b)

Figure 9: Ablation results on Quadratic Bandit: (a) Evaluation of the scalability of Ensemble++ with varying decision set sizes. (b) Performance of Ensemble++ under varying buffer sizes.

[2] https://huggingface.co/datasets/ucberkeley-dlab/measuring-hate-speech
[3] https://huggingface.co/openai-community/gpt2

Regret v.s. Computation Trade-off. We have demonstrated that Ensemble++ can achieve sub-linear regret with moderate computation cost in the Quadratic Bandit, as shown in **??**. In this experiment, we use the number of network parameters to measure computation cost and evaluate all methods in in the Quadratic Bandit with feature dimension $d = 100$ and candidate decision set size $|\mathcal{X}| = 1000$. Additional comparison results in the Neural Bandit, as shown in Figure 17, also support the same finding: across a range of ensemble sizes M, Ensemble++ outperforms baselines such as EpiNet and Ensemble+ in the regret–compute frontier, reaffirming that random linear combinations plus a shared base are quite cost-effective. A detailed discussion on the relationship between ensemble size and the network parameter size of Ensemble++ and other baselines is provided in Appendix A.8.1.

Ablation Studies on Scaling and Storage Requirement. We have demonstrated that the regret performance of Linear Ensemble++ Sampling is *not* affected by the decision set sizes in linear bandits. We extend this analysis to nonlinear bandits. As shown in Figure 9(a), Ensemble++ achieves similar performance under varying candidate decision set sizes. This finding further confirms our theoretical insights in Appendix D. We also observe that larger ensemble sizes M bring additional benefits, consistent with findings in linear bandits. Additionally, as introduced in Section 3.3, Ensemble++ does *not* require storing the entire history of data for training. To examine this, we compare different buffer sizes over a fixed period of 100,000 time steps. As shown in Figure 9(b), using a smaller buffer size results in only a slight performance drop. Nevertheless, Ensemble++ achieves comparable performance even with a buffer size smaller than the total time period. Furthermore, We provide guidance on choosing the distribution of index in Appendix H.2.

C Ensemble++ Algorithm Details

Here we provide detailed derivations and design choices for the Ensemble++ algorithm. Let $x \in \mathcal{X}$ denote the input, and $h(x; w)$ be the shared feature extractor parameterized by w. The extracted features are denoted by

$$\tilde{x} = h(x; w).$$

The base network $\psi(\tilde{x}; b)$, parameterized by b, estimates the mean prediction based on the shared features. The ensemble components $\{\psi(\mathrm{sg}(\tilde{x}); \theta_m)\}_{m=1}^{M}$, parameterized by θ_m, capture the uncertainty in the prediction. The stop-gradient operator $\mathrm{sg}(\cdot)$ prevents gradients from flowing through $\tilde{x}$ when computing gradients with respect to θ_m, effectively decoupling the ensemble components from the shared layers. The prior ensemble components $\{\psi(\tilde{x}; \theta_{0,m})\}_{m=1}^{M}$ are fixed throughout the learning process, incentivizing diverse exploration with prior variations in the initial stage where the data region is under-explored. Put all together, $\theta = \{w, b, \theta_1, \ldots, \theta_M, \theta_{0,1}, \ldots, \theta_{0,M}\}$ are the model parameters. By default, we choose ψ as a linear function:

$$\psi(\tilde{x}; \theta) = \langle \tilde{x}, \theta \rangle.$$

the Ensemble++ model predicts via random linear combinations of the base network and ensemble components, with the prior ensemble components fixed throughout the learning process. The model is defined as:

$$f_\theta^{++}(x, \zeta_t) = \psi(\tilde{x}; b) + \psi(\mathrm{sg}(\tilde{x}); \sum_{m=1}^{M} \zeta_{t,m}\theta_m) + \mathrm{sg}(\psi(\tilde{x}; \sum_{m=1}^{M} \zeta_{t,m}\theta_{0,m})), \tag{11}$$

where $\zeta_t = (\zeta_{t,1}, \ldots, \zeta_{t,M})^\top$ is a random vector sampled from an index distribution P_ζ.

Loss Function Derivation Starting from the loss function $L(\theta; D)$ with symmetric auxiliary variables:

$$\frac{1}{2M} \sum_{m=1}^{M} \sum_{s=1}^{N} \sum_{\beta \in \{1,-1\}} (Y_s + \beta \mathbf{z}_{s,m} - \psi(\tilde{x}_s; b) - \beta\psi(\mathrm{sg}(\tilde{x}_s); \theta_m) - \beta\, \mathrm{sg}(\psi(\tilde{x}_s; \theta_{0,m})))^2 + \Phi(\theta). \tag{12}$$

Expanding the square and summing over β:

$$
\begin{aligned}
&\sum_{\beta\in\{1,-1\}} \left(Y_s + \beta \mathbf{z}_{s,m} - \psi(\tilde{x}_s; b) - \beta\psi(\mathrm{sg}(\tilde{x}_s); \theta_m) - \beta\,\mathrm{sg}(\psi(\tilde{x}_s; \theta_{0,m}))\right)^2 \\
=&\sum_{\beta\in\{1,-1\}} \left((Y_s - \psi(\tilde{x}_s; b)) + \beta(\mathbf{z}_{s,m} - \mathrm{sg}(\psi(\tilde{x}_s; \theta_{0,m})) - \psi(\mathrm{sg}(\tilde{x}_s); \theta_m))\right)^2 \\
=&\,2\left((Y_s - \psi(\tilde{x}_s; b))^2 + (\mathbf{z}_{s,m} - \mathrm{sg}(\psi(\tilde{x}_s; \theta_{0,m})) - \psi(\mathrm{sg}(\tilde{x}_s); \theta_m))^2\right),
\end{aligned}
$$

since the cross terms cancel out due to summing over $\beta \in \{1, -1\}$. This leads to the simplified loss function $L(\theta; D)$:

$$
\frac{1}{M}\sum_{m=1}^{M}\sum_{s=1}^{N}\left[\frac{1}{2}\left(Y_s - \psi(\tilde{x}_s; b)\right)^2 + \frac{1}{2}\left(\mathbf{z}_{s,m} - \mathrm{sg}(\psi(\tilde{x}_s; \theta_{0,m})) - \psi(\mathrm{sg}(\tilde{x}_s); \theta_m)\right)^2\right] + \Phi(\theta). \quad (13)
$$

Gradient Computations The gradients with respect to the shared parameters (w, b) are derived solely from the base network loss:

$$
\nabla_w L(\theta; D) = \sum_{s=1}^{N}\left(\psi(\tilde{x}_s; b) - Y_s\right)\nabla_{\tilde{x}_s}\psi(\tilde{x}_s; b)\nabla_w h(A_s; w), \quad (14)
$$

$$
\nabla_b L(\theta; D) = \sum_{s=1}^{N}\left(\psi(\tilde{x}_s; b) - Y_s\right)\nabla_b \psi(\tilde{x}_s; b). \quad (15)
$$

The gradients with respect to the ensemble parameters θ_m are independent of the base network:

$$
\nabla_{\theta_m} L(\theta; D) = \sum_{s=1}^{N}\left(\psi(\mathrm{sg}(\tilde{x}_s); \theta_m) - Z_{s,m}\right)\nabla_{\theta_m}\psi(\mathrm{sg}(\tilde{x}_s); \theta_m). \quad (16)
$$

Note that due to the stop-gradient operator $\mathrm{sg}(\cdot)$, the ensemble components do not contribute to the gradients of shared parameters.

Classification Loss Function For classification tasks, we use the cross-entropy loss function instead of the squared loss function in Equation (12):

$$
L(\theta; D) = \frac{1}{2M}\sum_{m=1}^{M}\sum_{s=1}^{N}\sum_{\beta\in\{1,-1\}} \mathrm{CE}\left(f^{++}(X_s, \beta e_m), [Y_s, 1 - Y_s]\right) + \Phi(\theta) \quad (17)
$$

where $\mathrm{CE}(X, Y) = -\sum_j Y_j(X_j - \log\sum_i \exp X_i)$ is the cross-entropy loss function, and $[Y_s, 1 - Y_s]$ is the one-hot encoding of the label Y_s.

C.1 Design of Reference Distributions

The choice of index distribution P_ζ significantly impacts the exploration behavior of Ensemble++. We consider five distribution designs, each offering unique properties for different aspects of the algorithm:

1. **Gaussian Distribution** ($\zeta_t \sim \mathcal{N}(0, I_M)$):
   - Promotes diversity through natural covariance sampling
   - Provides strong theoretical guarantees
2. **Sphere Distribution** ($\zeta_t \sim \sqrt{M} \cdot \mathcal{U}(\mathbb{S}^{M-1})$):
   - Maintains perfect isotropy through rotational invariance
   - Ensures uniform exploration in all directions
   - Controls exploration magnitude with fixed norm

3. **Cube Distribution** ($\zeta_t \sim \mathcal{U}(\{1,-1\}^M)$):
   - Offers discrete exploration with binary choices
   - Provides strong anti-concentration properties
   - Computationally efficient for implementation
4. **Coordinate Distribution** ($\zeta_t \sim \mathcal{U}(\sqrt{M}\{\pm e_1, \dots, \pm e_M\})$):
   - Enables axis-aligned exploration
   - Minimizes interference between dimensions
   - Particularly useful for feature selection
5. **Sparse Distribution** (s-sparse random vectors):
   - Suitable for high-dimensional problems
   - Adjustable sparsity level for different settings

For sampling algorithms and the detailed theoretical analysis of the properties of these distributions (isotropy, concentration, and anticoncentration), see Appendix G.

D Theoretical Analysis

In this section, we provide a detailed theoretical analysis of *Linear Ensemble++ Sampling* in the linear contextual bandit setting (c.f. **??**). We show that, with an ensemble size $M = O(d \log T)$, Linear Ensemble++ Sampling achieves a near-optimal regret bound matching linear Thompson Sampling, while only incurring $O(d^3 \log T)$ computation per step. This closes a longstanding gap in scalable ensemble-based exploration.

W.L.O.G., we impose the following mild assumption.

Assumption 1. *The random noise ε_t satisfies*

$$\mathbb{E}\left[\exp\{s\,\varepsilon_t\} \mid \mathcal{H}_t,\, X_t\right] \;\le\; \exp\!\Big(\tfrac{s^2}{2}\Big), \quad \forall\, s \in \mathbb{R},$$

where $\mathcal{H}_t$ is the history up to time t. In addition, all actions satisfy $\|x\|_2 \le 1$ for $x \in \mathcal{X}$.

D.1 Key Lemma: Covariance Tracking under Sequential Dependence

A critical step in analyzing Linear Ensemble++ Sampling is to ensure that its incremental updates accurately track the true posterior covariance $\boldsymbol{\Sigma}_t$. Specifically, recall the Ensemble++ update for the matrix $\mathbf{A}_t$, which aims to approximate $\boldsymbol{\Sigma}_t^{1/2}$ even when actions X_t are chosen adaptively based on prior $\{\mathbf{A}_s\}_{s<t}$. The following lemma establishes that, provided M is on the order of $d \log T$, we obtain high-probability bounds ensuring $\mathbf{A}_t\mathbf{A}_t^\top$ remains an approximation to $\boldsymbol{\Sigma}_t$.

Lemma 1 (Covariance Tracking under Sequential Dependence)**.** *Let $\{\mathbf{z}_t\}_{t=1}^T$ be unit-norm $\sqrt{1/M}$-sub-Gaussian vectors in $\mathbb{R}^M$, adapted to a filtration $\{\mathcal{F}_t\}$. Define $\mathbf{A}_t$ via the recursive update equation 3 in Linear Ensemble++, and let $\boldsymbol{\Sigma}_t$ be the exact ridge posterior covariance. Suppose*

$$\begin{aligned} M \;&\ge\; 320\Big(d\,\log\Big(\tfrac{2+\frac{96}{s_{\min}}\sqrt{s_{\max}^2+T}}{\delta}\Big) \;+\; \log\Big(1+\tfrac{T}{s_{\min}^2}\Big)\Big) \\ &\simeq\; d\Big(\log\tfrac{1}{\delta} + \log T\Big), \end{aligned} \tag{18}$$

where $s_{\min}^2 = \inf_{\|a\|=1} a^\top \boldsymbol{\Sigma}_0^{-1} a$ and $s_{\max}^2 = \sup_{\|a\|=1} a^\top \boldsymbol{\Sigma}_0^{-1} a$. Then with probability at least $1-\delta$,

$$\boxed{\forall\, t \le T: \quad \tfrac{1}{2}\,\boldsymbol{\Sigma}_t \;\preccurlyeq\; \mathbf{A}_t\,\mathbf{A}_t^\top \;\preccurlyeq\; \tfrac{3}{2}\,\boldsymbol{\Sigma}_t.}$$

Significance. Lemma 1 ensures that the Ensemble++ "covariance factor" $\mathbf{A}_t\mathbf{A}_t^\top$ tracks the true posterior $\boldsymbol{\Sigma}_t$ to within constant factors, *uniformly* for all $t \le T$. Thus, the variance estimates used by Ensemble++ remain trustworthy at each decision point, despite the *sequential* dependencies in how actions are chosen.

Technical Innovation. The primary challenge is dealing with the *sequential* dependencies among *adaptively chosen* actions X_t and the high-dimensional random vectors $\mathbf{z}_t$. Our proof uses:

- A *variance-aware discretization* scheme that *avoids* super-linear growth in the required ensemble size M. Naive discretizations can require $M = \Omega(d\,T^2 \log T)$. Our approach cuts down the dimension requirement to $\widetilde{O}(d \log T)$, keeping computation manageable.
- A reduction to *Sequential Johnson–Lindenstrauss (JL)* arguments Li (2024a), which handle time-varying, high-dimensional data under adaptivity.

Proof Sketch. Rewriting the update rule $\Sigma_t^{-1}\mathbf{A}_t = \Sigma_{t-1}^{-1}\mathbf{A}_{t-1} + X_t\,\mathbf{z}_t^\top$, we analyze one-dimensional projections $a^\top \mathbf{A}_t$ by leveraging the sequential JL lemma on discritized directions a. *1. Simpler case*: when X_t is a standard basis vector, $\boldsymbol{\Sigma}_t$ is diagonal, simplifying the updates; the proof then boils down to bounding the diagonal entries via a direct sequential JL approach. *2. General feature vectors*: we still focus on $a^\top \mathbf{A}_t$ for unit vectors a, and show concentration around $a^\top \Sigma_t^{-1} a$. Combining this with a covering argument (discretizing the unit sphere with a covariance-aware weighted norm) and a union bound yields the matrix inequalities. Formal details are deferred to Appendix E.

D.2 Regret Bound for Linear Ensemble++ Sampling

Building on Lemma 1, we now show that *Linear Ensemble++ Sampling* attains near-optimal regret comparable to linear Thompson Sampling. Let P_ζ be the *reference distribution* used in sampling ζ_t for action selection.

Theorem 1 (Distribution-dependent Regret)**.** *Suppose Assumption 1 holds, and let $\boldsymbol{\Sigma}_0$ be the prior covariance. If Lemma 1 applies (i.e., M satisfies equation 18), then Linear Ensemble++ Samplingachieves the following regret bound with probability at least $1-\delta$:*

$$\mathrm{Regret}(T) \;\le\; \frac{\rho(P_\zeta)}{p(P_\zeta)}\,\beta\,\sqrt{T\,d\,\log\!\Big(1+\frac{T}{\lambda\,d}\Big)},$$

where $\beta = \sqrt{\lambda}\,\|\theta^*\|_2 + \sqrt{2\log\!\big(\frac{1}{\delta}\big)}$, and $\rho(P_\zeta), p(P_\zeta)$ are distribution-dependent constants described below. See Appendix F for full proof details.

Reference Distribution. A crucial design element in Ensemble++ is the choice of sampling distribution P_ζ.

Example 1 (Reference Distribution Choices)**.** *We can sample $\zeta_t \sim P_\zeta$ from, e.g., Gaussian distribution $\mathcal{N}(0,\, I_M)$, Sphere distribution $\sqrt{M}\cdot\mathcal{U}(\mathbb{S}^{M-1})$, Cube distribution $\mathcal{U}(\{\pm 1\}^M)$, Coordinate distribution $\mathcal{U}\big(\{\pm e_1,\,\ldots,\,\pm e_M\}\big)$ or Sparse distributions (s-sparse random vectors). Each design has different isotropy and anti-concentration properties that affect $\rho(P_\zeta)$ and $p(P_\zeta)$; see Table 2 for examples and Appendix G for formal definitions.*

Table 2: Representative values of $\rho(P_\zeta)$ and $p(P_\zeta)$ for typical distributions. The ratio $\frac{\rho(P_\zeta)}{p(P_\zeta)}$ appears in Theorem 1 and influences the final regret constant. Notation: $\rho_1 = O\big(\sqrt{M\log(M/\delta)}\big)$, $\rho_2 = O(\sqrt{M})$, and $\rho_3 = O\big(\sqrt{\log(|\mathcal{X}|/\delta)}\big)$.

P_ζ	$\mathcal{N}(0, I_M)$	$\sqrt{M}\cdot\mathcal{U}(\mathbb{S}^{M-1})$	$\mathcal{U}(\{\pm 1\}^M)$	$\mathcal{U}(\{\pm e_i\})$	Sparse
$\rho(P_\zeta)$	$\rho_1 \;\wedge\; \rho_3$	$\rho_2 \;\wedge\; \rho_3$	$\rho_2 \;\wedge\; \rho_3$	ρ_2	ρ_2
$p(P_\zeta)$	$\frac{1}{4\sqrt{e\pi}}$	$\frac{1}{2} - \frac{e^{1/12}}{\sqrt{2\pi}}$	$\frac{7}{32}$	$\frac{1}{2M}$	N/A

Discussion of Reference Distributions. Continuous-support distributions (e.g., Gaussian or uniform on the sphere) often yield a more favorable ratio $\rho(P_\zeta)/p(P_\zeta)$ than discrete distributions (e.g., uniform on cube or coordinate vectors). Consequently, continuous P_ζ provides tighter regret constants and improved exploration efficiency. Furthermore, for finite action sets, an additional $\log|\mathcal{X}|$ factor may appear in ρ_3, but this still remains within a $\tilde{O}(\sqrt{T\log|\mathcal{X}|})$ regret scaling, matching the best known linear TS bounds.

D.3 Comparisons and Implications

Table 3 places Ensemble++ in the context of related algorithms and analyses. Notably, it is the *first* approximate-TS method to achieve:

- **Scalable per-step updates** of $\Theta(d^3 \log T)$, rather than depending on T or $|\mathcal{X}|$,
- **Near-optimal regret matching linear TS** across all decision set setups (finite set or compact set, time-invariant or time-varying).

This addresses the longstanding computational–statistical trade-off in ensemble-based exploration, surpassing prior methods such as LMCTS Xu et al. (2022), which require $\mathcal{O}(d^2T)$ cost per step, or earlier ensemble sampling analyses that need $M \propto T$ Qin et al. (2022).

Table 3: Regret upper bounds for representative algorithms in linear bandits under different action-set setups. For references: (1) Qin et al. (2022) covers Bayesian regret in time-invariant *finite* actions; (2) Janz et al. (2024) assumes time-invariant or continuous sets; (3) Abeille & Lazaric (2017) and Agrawal & Goyal (2013) analyze linear Thompson Sampling (TS); Our method, Ensemble++, is the first approximate TS algorithm to handle all four setups with $O(d^3 \log T)$ per-step complexity.

	Inv. & Compact	**Var. & Compact**	**Inv. & Finite**	**Var. & Finite**
linear TS	$O(d^{3/2}\sqrt{T}\log T)$	$O(d^{3/2}\sqrt{T}\log T)$	$O(d\sqrt{T\,\log\lvert\mathcal{X}\rvert}\,\log T)$	$O(d\sqrt{T\,\log\lvert\mathcal{X}\rvert}\,\log T)$
Qin et al. (2022)	N/A	N/A	$O(\sqrt{dT\,\log\lvert\mathcal{X}\rvert\log\frac{\lvert\mathcal{X}\rvert T}{d}})$	N/A
Janz et al. (2024)	$O\big((d\log T)^{5/2}\sqrt{T}\big)$	$O\big((d\log T)^{5/2}\sqrt{T}\big)$	N/A	N/A
Ensemble++	$O\big(d^{3/2}\sqrt{T}\,(\log T)^{3/2}\big)$	$O\big(d^{3/2}\sqrt{T}\,(\log T)^{3/2}\big)$	$O\big(d\sqrt{T\,\log\lvert\mathcal{X}\rvert}\,\log T\big)$	$O\big(d\sqrt{T\,\log\lvert\mathcal{X}\rvert}\,\log T\big)$

Remark 2 (Efficiency)**.** *Linear Ensemble++ Sampling provides an* exponential improvement *in the T-dependence of computational cost relative to prior works Qin et al. (2022); Xu et al. (2022) that require $O(T)$ or $O(d^2T)$ per-step overhead to achieve near-TS regret. It also refines concurrent ensemble sampling bounds Janz et al. (2024), improving by $O(d(\log T)^2)$ in regret. These gains realize a more practical method for large-scale or long-horizon tasks. For Frequentist analysis, we choose an inflated version of linear Ensemble++ sampling, detailed in Appendix F.*

Remark 3 (Flexibility)**.** *Unlike prior analyses specialized to* compact *action sets (e.g., Abeille & Lazaric (2017); Xu et al. (2022); Janz et al. (2024)) or* finite *action sets only (e.g., Qin et al. (2022)), our results apply seamlessly to* both *scenarios and remain valid when $\mathcal{X}_t$ is time-varying or remains fixed over t. This broad applicability underscores the generality of Ensemble++ as a scalable approximation to Thompson sampling in linear bandits.*

Overall, these results confirm that *Linear Ensemble++ Sampling* matches the exploration quality of TS without incurring large ensemble sizes or per-step costs.

E Technical Details for Lemma 1

Proof Sketch Notice the recursive update rule Equation (3) can be rewritten as

$$\Sigma_t^{-1}\mathbf{A}_t = \Sigma_{t-1}^{-1}\mathbf{A}_{t-1} + X_t\mathbf{z}_t^\top. \tag{19}$$

We first consider a simpler setting where the feature vectors X_t are from the standard basis, reducing the problem to a multi-armed bandit setting. In this case, $\boldsymbol{\Sigma}_t$ is a diagonal matrix, and Equation (19) reduces to Equation (20) in example 2 where σ_t^i is the i-th diagonal element of $\boldsymbol{\Sigma}_t$. Then the proof goal of Lemma 1 share the exact same goal (c.f. Equation (21)) in example 2, where the sequential Johnson-Lindenstrauss theorem (introduced later in Theorem 2) can be applied to show that the incremental uncertainty estimates remain accurate even with *sequential dependence*.

Example 2 (Approximate Posterior in a Multi-Armed Bandit)**.** *Consider a multi-armed bandit with K independent arms, each having an unknown mean reward θ_i^*. A Gaussian prior is placed on each arm's mean $\theta_i^* \sim \mathcal{N}(\mu_0^i, \sigma_0^2)$. At time t, the algorithm pulls an arm X_t. The posterior variance $(\sigma_t^i)^2$ of arm i is updated as*

$$\frac{1}{(\sigma_t^i)^2} = \frac{1}{(\sigma_{t-1}^i)^2} + \mathbf{1}_{\{X_t = e_i\}},$$

and $(\sigma_t^i)^2$ is left unchanged for unchosen arms.

Ensemble++ produces posterior samples *relies on storing a high-dimensional factor* $m_t^i \in \mathbb{R}^M$ *approximating* $(\sigma_t^i)^2$ *through its square norm* $\|m_t^i\|^2$. *We draw* $\zeta \sim \mathcal{N}(0, I_M)$ *and form* $\mu_t^i + \langle m_t^i, \zeta \rangle$ *as an approximate posterior sample. To maintain* m_t^i *efficiently, an* incremental update *is used (reduced from Equation* (19)*):*

$$\frac{1}{(\sigma_t^i)^2}\, m_t^i \;=\; \frac{1}{(\sigma_{t-1}^i)^2}\, m_{t-1}^i \;+\; \mathbf{1}_{\{X_t = e_i\}}\, \mathbf{z}_t, \tag{20}$$

where $\mathbf{z}_t$ *are fresh random vectors at each step. For initialization, set* $m_0^i = \sigma_0\, \mathbf{z}_0^i$ *so that* $\|m_0^i\|^2 = \sigma_0^2$.

Note that X_t *depends on* all past data *and thus on* $\mathbf{z}_0, \ldots, \mathbf{z}_{t-1}$. *Denoting* $x_t := \mathbf{1}_{\{X_t = e_i\}}$, *we see that each* x_t *is adaptive to* $\{\mathbf{z}_s\}_{s<t}$ *and* $\mathbf{z}_t$ *is the fresh perturbation at each step. Rewriting Equation* (20) *for a fixed arm* i*:*

$$\frac{1}{(\sigma_t^i)^2}\, m_t^i \;=\; \sum_{s=0}^{t} x_s\, \mathbf{z}_s, \quad \textit{while} \quad \frac{1}{(\sigma_t^i)^2} \;=\; \sum_{s=0}^{t} x_s^2.$$

Hence, we want:

$$\Big\|\sum_{s=0}^{t} x_s \mathbf{z}_s\Big\|^2 \;\approx\; \sum_{s=0}^{t} x_s^2 \quad \textit{uniformly over } t \in \{0, \ldots, T\}. \tag{21}$$

Standard JL arguments break under such sequential dependence *between* x_t *and* $\{\mathbf{z}_s\}_{s<t}$, *motivating our sequential-JL theorem, as described later in Theorem* 2.

For the general case where X_t are arbitrary bounded feature vectors, we still leverage the matrix recursion structure in Equation (19) but investigate its projection onto a single direction a:

$$a^\top \Sigma_t^{-1} \mathbf{A}_t = a^\top \Sigma_{t-1}^{-1} \mathbf{A}_{t-1} + a^\top X_t \mathbf{z}_t^\top,$$

a form proven to be concentrated around $a^\top \Sigma_t^{-1} a$ using the sequential Johnson-Lindenstrauss theorem. The proof then proceeds by carefully selecting representative directions to discretize the unit sphere and applying a union bound to extend the concentration results to the entire continuous space $\mathbb{S}^{d-1}$. Immediately, we can convert the guarantee about $\{a^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{A}_t \mathbf{A}_t^\top \boldsymbol{\Sigma}_t^{-1} a, a \in \mathbb{S}^{M-1}\}$ to the desired result about $\{a^\top \mathbf{A}_t \mathbf{A}_t^\top a, \forall a \in \mathbb{S}^{M-1}\}$.

In the later subsection, we rigorously formalize each of these steps.

E.1 Fundamental probability tools: Sequential Johnson-Lindenstrauss

First, we state the preliminary tools of sequential Johnson-Lindenstrauss (JL) for completeness, which is adapted form (Li, 2024a). This tool was used to prove incremental posterior approximation argument of HyperAgent in tabular RL setup (Li et al., 2024). As the tool in (Li, 2024a) works only for the scalar process, we need additional technical innovations to deal with high-dimensional vector process. Thus, we make a novel utilization of this tool in the linear function approximation setting for the first time, by a non-trivial discretization argument in Appendix E.3.

We define some important concept that would be useful in the analysis. Let $(\Omega, \mathcal{F}, \mathbb{F} = (\mathcal{F}_t)_{t\in\mathbb{N}}, \mathbb{P})$ be a complete filtered probability space. We first consider the measurable properties within the filtered probability space.

Definition 1 (Adapted process)**.** *For an index set* I *of the form* $\{t \in \mathbb{N} : t \geq t_0\}$ *for some* $t_0 \in \mathbb{N}$, *we say a stochastic process* $(X_t)_{t\in I}$ *is adapted to the filtration* $(\mathcal{F}_t)_{t\in I}$ *if each* X_t *is* $\mathcal{F}_t$*-measurable.*

Definition 2 ((Conditionally) σ-sub-Gaussian)**.** *A random variable* $X \in \mathbb{R}$ *is* σ*-sub-Gaussian if*

$$\mathbb{E}[\exp(\lambda X)] \leq \exp\left(\frac{\lambda^2 \sigma^2}{2}\right), \qquad \forall \lambda \in \mathbb{R}.$$

Let $(X_t)_{t\geq 1} \subset \mathbb{R}$ *be a stochastic process adapted to filtration* $(\mathcal{F}_t)_{t\geq 1}$. *Let* $\sigma = (\sigma_t)_{t\geq 0}$ *be a stochastic process adapted to filtration* $(\mathcal{F}_t)_{t\geq 0}$. *We say the process is* $(X_t)_{t\geq 1}$ *is conditionally* σ*-sub-Gaussian if*

$$\mathbb{E}[\exp(\lambda X_t) \mid \mathcal{F}_{t-1}] \leq \exp\left(\frac{\lambda^2 \sigma_{t-1}^2}{2}\right), \quad a.s. \quad \forall \lambda \in \mathbb{R}.$$

Specifically for the index $t+1$, we can say X_{t+1} is ($\mathcal{F}_t$-conditionally) σ_t-sub-Gaussian. If σ_t is a constant σ for all $t \geq 0$, then we just say (conditionally) σ-sub-Gaussian.

For a random vector $X \in \mathbb{R}^M$ or vector process $(X_t)_{t\geq 1} \subset \mathbb{R}^M$ in high-dimension, we say it is σ-sub-Gaussian is for every fixed $v \in \mathbb{S}^{M-1}$ if the random variable $\langle v, X\rangle$, or the scalarized process $(\langle v, X_t\rangle)_{t\geq 1}$ is σ-sub-Gaussian.

Definition 3 (Almost sure unit-norm)**.** *We say a random variable X is almost sure unit-norm if $\|X\|_2 = 1$ almost surely.*

Remark 4. *When talking about the perturbation distribution $P_{\mathbf{z}}$, we scale all specific distribution discussed in Appendix G by $\sqrt{\frac{1}{M}}$. Then the spherical distribution $\mathcal{U}(\mathbb{S}^{M-1})$ and uniform over scaled cube $\mathcal{U}(\frac{1}{\sqrt{M}}\{1,-1\}^M)$ satisfy the sub-Gaussian condition in Definition 2 with parameter $\sigma = \frac{1}{\sqrt{M}}$ and also satisfy the unit-norm condition in Definition 3 according to the discussion in Appendix G.*

Now, we are ready to state the important tool that is fundamental to our analysis.

Theorem 2 (Sequential Johnson–Lindenstrauss (Li, 2024a))**.** *Fix $\varepsilon \in (0,1)$, and let $\{\mathcal{F}_t\}_{t\geq 0}$ be a filtration. Consider random vectors $\{\mathbf{z}_t\}_{t\geq 0} \subset \mathbb{R}^M$ adapted to $\{\mathcal{F}_t\}_{t\geq 0}$, satisfying:*

- *$\mathbf{z}_0$ is $\mathcal{F}_0$-measurable, $\mathbb{E}[\|\mathbf{z}_0\|^2] = 1$, and $\left|\|\mathbf{z}_0\|^2 - 1\right| \leq \varepsilon/2$ almost surely.*
- *For $t \geq 1$, the process $\{\mathbf{z}_t\}_{t\geq 1}$ is conditionally $\sqrt{c_0/M}$-sub-Gaussian and each $\|\mathbf{z}_t\| = 1$ almost surely.*

Let $\{x_t\}_{t\geq 1} \subset \mathbb{R}$ be adapted to $\{\mathcal{F}_{t-1}\}_{t\geq 1}$ and satisfy $x_t^2 \leq c_x$ a.s. For a fixed $x_0 \in \mathbb{R}\setminus\{0\}$, if

$$M \;\geq\; \frac{16\, c_0\,(1+\varepsilon)}{\varepsilon^2}\left(\log\!\left(\tfrac{1}{\delta}\right) \;+\; \log\!\left(1 + \tfrac{c_x\, T}{x_0^2}\right)\right),$$

then with probability at least $1-\delta$,

$$\forall\, t = 0,\dots,T: \quad (1-\varepsilon)\sum_{i=0}^{t} x_i^2 \;\;\leq\;\; \left\|\sum_{i=0}^{t} x_i\,\mathbf{z}_i\right\|^2 \;\;\leq\;\; (1+\varepsilon)\sum_{i=0}^{t} x_i^2.$$

E.2 Reduce Lemma 1 to Sequential Johnson-Lindenstrauss (Theorem 2)

Without loss of generality, let us consider the compact set $\mathbb{S}^{d-1}$ define the feature space of all actions. First, we define a fine-grained good event for desired approximation error $\varepsilon \in (0,1)$: the approximate posterior variance $a^\top \mathbf{A}_t\mathbf{A}_t^\top a$ is ε-close to the true posterior variance $a^\top \boldsymbol{\Sigma}_t a$ for direction a at time $t \in \mathcal{T} := \{0,1,\dots,T\}$, i.e.,

$$\mathcal{G}_t(a,\varepsilon) = \left\{|a^\top \mathbf{A}_t\mathbf{A}_t^\top a - a^\top \boldsymbol{\Sigma}_t a| \leq \varepsilon a^\top \boldsymbol{\Sigma}_t a\right\}, \tag{22}$$

and corresponding joint event over the set $\mathbb{S}^{d-1}$,

$$\mathcal{G}_t(\varepsilon) = \bigcap_{a\in\mathbb{S}^{d-1}} \mathcal{G}_t(a,\varepsilon). \tag{23}$$

The good event at time priod t defined in Lemma 1 is indeed $\mathcal{G}_t(1/2)$.

A reduction. To fully utilize the probability tool for Sequential Johnson-Lindenstrauss in Theorem 2, we make use of the following reduction from vector process to scalar process. For a fixed $a \in \mathbb{S}^{d-1}$, we let $\mathbf{s}(a) = a^\top \boldsymbol{\Sigma}_0^{-1/2}\mathbf{Z}_0, s(a)^2 = a^\top \boldsymbol{\Sigma}_0^{-1} a$. Further define short notation $\mathbf{z}_0 := \mathbf{s}(a)^\top / s(a)$ and $x_0 := s(a)$. and $x_t = a^\top X_t$ for all $t \in [T]$, then we can relate the incremental update in Equation (20)

$$a^\top \boldsymbol{\Sigma}_t^{-1}\mathbf{A}_t = \underbrace{a^\top \boldsymbol{\Sigma}_0^{-1/2}\mathbf{Z}_0}_{\mathbf{s}(a)=\mathbf{z}_0^\top x_0} + \sum_{i=1}^{t} \underbrace{a^\top (X_i)}_{x_i}\mathbf{z}_i^\top, \quad a^\top \boldsymbol{\Sigma}_t^{-1} a = \underbrace{a^\top \boldsymbol{\Sigma}_0^{-1} a}_{x_0^2} + \sum_{i=1}^{t} \underbrace{a^\top (X_i)(X_i)^\top a}_{x_i^2}$$

to the scalar sequence $(x_t)_{t\geq 0}$ and the vector sequence $(\mathbf{z}_t)_{t\geq 0}$ that would be applied in Theorem 2.

Recall that $\mathcal{H}_t$ the σ-algebra generated from history $(\mathcal{X}_1, X_1, Y_1, \dots, \mathcal{X}_{t-1}, X_{t-1}, Y_{t-1}, \mathcal{X}_t)$. Denote $\mathcal{Z}_1 = \sigma(\mathbf{Z}_0)$ and $\mathcal{Z}_t = \sigma(\mathbf{Z}_0, \mathbf{z}_1, \dots, \mathbf{z}_{t-1})$ for $t \geq 2$. We observe the following statistical relationship, which is further demonstrated in Figure 10

- $\mathbf{z}_t \perp\!\!\!\perp (\mathcal{H}_t, X_t, \mathcal{Z}_t)$, X_t is dependent on $\mathcal{H}_t, \mathcal{Z}_t$,
- $\mathbf{A}_{t-1} \in \sigma(\mathcal{H}_t, \mathcal{Z}_t)$,
- $\mu_{t-1}, \boldsymbol{\Sigma}_{t-1} \in \mathcal{H}_t$.

For all $t \geq \mathbb{N}$, let us define the sigma-algebra $\mathcal{F}_t = \sigma(\mathcal{H}_{t+1}, \mathcal{Z}_{t+1}, X_{t+1})$. We can verify $\mathcal{F}_k \subseteq \mathcal{F}_l$ for all $k \leq l$. Thus $\mathbb{F} = (\mathcal{F}_t)_{t\in\mathbb{N}}$ is a filtration. Now, we could verify $(\mathbf{z}_t)_{t\geq 0}$ is adapted to $(\mathcal{F}_t)_{t\geq 0}$ and $(x_t)_{t\geq 1}$ is adapted to $(\mathcal{F}_t)_{t\geq 0}$, satisfying the conditions in Theorem 2.

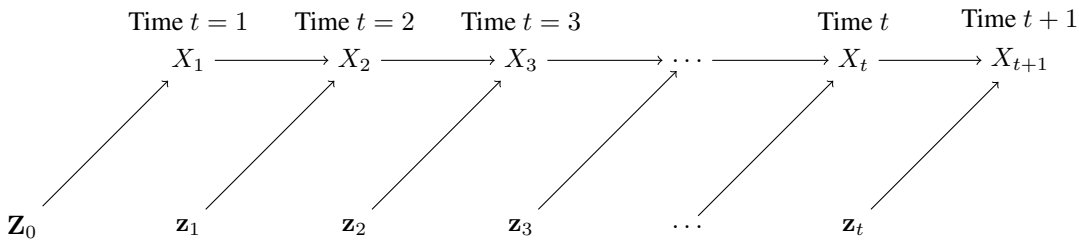


Figure 10: Sequential Dependence Structure.

E.2.1 Prior approximation

First, we state a standard covering argument on sphere.

Lemma 2 (Covering number of a sphere)**.** *There exists a set $\mathcal{C}_\iota \subset \mathbb{S}^{d-1}$ with $|\mathcal{C}_\iota| \leq (1 + 2/\iota)^d$ such that for all $x \in \mathbb{S}^{d-1}$ there exists a $y \in \mathcal{C}_\iota$ with $\|x - y\|_2 \leq \iota$.*

Lemma 3 (Computing spectral norm on a covering set)**.** *Let $\boldsymbol{A}$ be a symmetric $d \times d$ matrix, and let $\mathcal{C}_\iota$ be the an ι-covering of $\mathbb{S}^{d-1}$ for some $\iota \in (0, 1)$. Then,*

$$\|\boldsymbol{A}\| = \sup_{x\in\mathbb{S}^{d-1}} |x^\top \boldsymbol{A} x| \leq (1 - 2\iota)^{-1} \sup_{x\in\mathcal{C}_\iota} |x^\top \boldsymbol{A} x|.$$

For compact set $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d : \|x\| = 1\}$, by standard covering argument in Lemma 3 and the distributional Johnson-Lindenstrauss lemma (Li, 2024b), when

$$M \geq M_1(\varepsilon, \delta) := 256\varepsilon^{-2}(d \log 9 + \log(2/\delta)), \tag{24}$$

the initial good event for prior approximation $G_0(\varepsilon/2)$ holds with probability at least $1 - \delta$.

Next, we are going to show that, under the event $G_0(\varepsilon/2)$, the initial condition on $|\|\mathbf{z}_0\|^2 - 1| \leq (\varepsilon/2)$ in Theorem 2 is satisfied. That is, under the event $G_0(\varepsilon/2)$

$$\begin{aligned}
& (1 - \varepsilon/2)a^\top \boldsymbol{\Sigma}_0 a \leq \|a^\top \boldsymbol{\Sigma}_0^{1/2} \mathbf{Z}_0\|^2 \leq (1 + \varepsilon/2)a^\top \boldsymbol{\Sigma}_0 a, \quad \forall a \in \mathbb{S}^{d-1} \\
\Leftrightarrow \quad & \|\mathbf{Z}_0 \mathbf{Z}_0^\top - \boldsymbol{I}\| \leq \varepsilon/2 \\
\Leftrightarrow \quad & (1 - \varepsilon/2)a^\top \boldsymbol{\Sigma}_0^{-1} a \leq \|a^\top \boldsymbol{\Sigma}_0^{-1/2} \mathbf{Z}_0\|^2 \leq (1 + \varepsilon/2)a^\top \boldsymbol{\Sigma}_0^{-1} a, \quad \forall a \in \mathbb{S}^{d-1}.
\end{aligned} \tag{25}$$

Recall the short notation $\mathbf{s}(a) = a^\top \boldsymbol{\Sigma}_0^{-1/2} \mathbf{Z}_0$ and $s(a)^2 = a^\top \boldsymbol{\Sigma}_0^{-1} a$, we have $\mathbf{z}_0 = \mathbf{s}(a)^\top / s(a)$ satisfying $|\|\mathbf{z}_0\|^2 - 1| \leq (\varepsilon/2)$ according to Equation (25).

E.2.2 Posterior approximation

Notice that $x_0^2 = a^\top \boldsymbol{\Sigma}_0 a \geq \inf_{a\in\mathbb{S}^{d-1}} a^\top \boldsymbol{\Sigma}_0^{-1} a = s_{\min}^2$. By the assumption of the bounded feature in assumption 1, we can examine that $x_t^2 = (a^\top X_t)^2 \leq 1$ for $t \geq 1$. That is, the sequence $(a^\top X_t)_{t\geq 1}$ is 1-square-bounded for any $a \in \mathbb{S}^{d-1}$.

We could also check that $(\mathbf{z}_t)_{t\geq 1}$ is $1/\sqrt{M}$-sub-Gaussian and with unit-norm when the perturbation distribution P_z is Cube $\mathcal{U}(\{1, -1\}^M)$ or Sphere $\mathcal{U}(\mathbb{S}^{M-1})$.

Under the prior approximation event $\mathcal{G}_0(\varepsilon/2)$, we apply Theorem 2 to show that for any fixed $a \in \mathbb{S}^{d-1}$,

$$\forall t \in \mathcal{T}, E_t(a,\varepsilon) := \left\{ |a^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{A}_t \mathbf{A}_t^\top \boldsymbol{\Sigma}_t^{-1} a - a^\top \boldsymbol{\Sigma}_t^{-1} a| \le \varepsilon a^\top \boldsymbol{\Sigma}_t^{-1} a \right\} \tag{26}$$

holds with probability at least $1-\delta$ when

$$M \ge \frac{16(1+\varepsilon)}{\varepsilon^2} \left(\log\left(\frac{1}{\delta}\right) + \log\left(1 + \frac{T}{s_{\min}^2}\right) \right). \tag{27}$$

E.3 Discretization for Posterior Approximation

We need discretization (covering) argument to relate the result in Equation (26) to the desired good event defined in Equation (23)

$$\mathcal{G}_t(\varepsilon) = \left\{ \| \boldsymbol{\Sigma}_t^{-1/2} \mathbf{A}_t \mathbf{A}_t^\top \boldsymbol{\Sigma}_t^{-1/2} - \boldsymbol{I} \| \le \varepsilon \right\}.$$

Standard discretization produces unacceptable results. Utilizing standard discretization for computing spectral norm in Lemma 3, let $\iota = 1/4$, we can show that

$$\bigcap_{a \in \mathcal{C}_{1/4}} E_t(a, \varepsilon/2T) \subseteq \mathcal{G}_t(\varepsilon).$$

This is due to,

$$\begin{aligned}
\| \boldsymbol{\Sigma}_t^{-1/2} \mathbf{A}_t \mathbf{A}_t^\top \boldsymbol{\Sigma}_t^{-1/2} - \boldsymbol{I} \| &= \sup_{x \in \mathbb{S}^{d-1}} \frac{|x^\top (\boldsymbol{\Sigma}_t^{-1} \mathbf{A}_t \mathbf{A}_t^\top \boldsymbol{\Sigma}_t^{-1} - \boldsymbol{\Sigma}_t^{-1}) x|}{x^\top \boldsymbol{\Sigma}_t^{-1} x} \\
&\le \frac{2}{\lambda_{\min}(\boldsymbol{\Sigma}_t^{-1})} \sup_{a \in \mathcal{C}_{1/4}} |a^\top (\boldsymbol{\Sigma}_t^{-1} \mathbf{A}_t \mathbf{A}_t^\top \boldsymbol{\Sigma}_t^{-1} - \boldsymbol{\Sigma}_t^{-1}) a| \\
&\le 2\varepsilon' \frac{\sup_{a \in \mathcal{C}_{1/4}} a^\top \boldsymbol{\Sigma}_t^{-1} a}{\lambda_{\min}(\boldsymbol{\Sigma}_t^{-1})} \le 2\varepsilon' \cdot \kappa(\boldsymbol{\Sigma}_t^{-1}) \le 2T\varepsilon'.
\end{aligned}$$

Then by union bound over $\mathcal{C}_{1/4}$, plugging in $\varepsilon/2T$ to Equation (27), we require $M \ge \tilde{O}(dT^2 \log T)$ to let $\bigcap_{a \in \mathcal{C}_{1/4}} E_t(a, \varepsilon/2T)$ hold with probability at least $1-\delta$. This result is not acceptable as the per-step computation complexity is growing unbounded polynimally with the interaction steps T. In the next section, we provide a non-trivial discretization to resolve this analytical problem.

Variance-aware discretization. The key contribution here is that we choose a variance weighted norm to measure discretization error. This variance-awareness, together with specific choice on a $O(1/\sqrt{T})$-discretization error and a constant approximation error ε, eventually arrives at $O(d \log T)$ log covering number and $M = \tilde{O}(d \log T)$ in Lemma 1.

Let $\mathbf{S}_t = \boldsymbol{\Sigma}_t^{-1} \mathbf{A}_t = \mathbf{X}_t^\top \mathbf{Z}_t$ and $\boldsymbol{\Gamma}_t = \boldsymbol{\Sigma}_t^{1/2} \mathbf{S}_t = \boldsymbol{\Sigma}_t^{-1/2} \mathbf{A}_t$. Notice that, from Equation (26), the event holds with probability at least $1-\delta'$

$$\forall t \in \mathcal{T}, E_t(a, \varepsilon') = \left\{ \frac{|a^\top \mathbf{S}_t \mathbf{S}_t^\top a - a^\top \boldsymbol{\Sigma}_t^{-1} a|}{a^\top \boldsymbol{\Sigma}_t^{-1} a} \le \varepsilon' \right\}$$

when

$$M \ge \frac{16(1+\varepsilon')}{(\varepsilon')^2} \left(\log\left(\frac{1}{\delta'}\right) + \log\left(1 + \frac{T}{s_{\min}^2}\right) \right).$$

Let $\mathcal{C}_\iota \subset \mathbb{S}^{d-1}$ be the ι-covering set in Lemma 2 and the event $\bigcap_{a \in \mathcal{C}_\iota} E_t(a, \varepsilon')$ holds. Let $x \in \mathbb{S}^{d-1}$ and $y \in \mathcal{C}_\iota$ such that $\|x - y\| \le \iota$. Define short notation $u = \boldsymbol{\Sigma}_t^{-1/2} x$, $v = \boldsymbol{\Sigma}_t^{-1/2} y$.

$$\begin{aligned}
&\frac{|x^\top \mathbf{S}_t \mathbf{S}_t^\top x - x^\top \boldsymbol{\Sigma}_t^{-1} x|}{x^\top \boldsymbol{\Sigma}_t^{-1} x} - \frac{|y^\top \mathbf{S}_t \mathbf{S}_t^\top y - y^\top \boldsymbol{\Sigma}_t^{-1} y|}{y^\top \boldsymbol{\Sigma}_t^{-1} y} \\
&= \frac{|u^\top \boldsymbol{\Gamma}_t \boldsymbol{\Gamma}_t^\top u - u^\top u|}{u^\top u} - \frac{|v^\top \boldsymbol{\Gamma}_t \boldsymbol{\Gamma}_t^\top v - v^\top v|}{v^\top v} = \frac{|\|\boldsymbol{\Gamma}_t u\|^2 - \|u\|^2|}{\|u\|^2} - \frac{|\|\boldsymbol{\Gamma}_t v\|^2 - \|v\|^2|}{\|v\|^2} \\
&\le \left| \frac{\|\boldsymbol{\Gamma}_t u\|^2}{\|u\|^2} - \frac{\|\boldsymbol{\Gamma}_t v\|^2}{\|v\|^2} \right| = \underbrace{\left| \frac{\|\boldsymbol{\Gamma}_t u\|^2 - \|\boldsymbol{\Gamma}_t v\|^2}{\|u\|^2} \right|}_{(I)} + \underbrace{\|\boldsymbol{\Gamma}_t v\|^2 \left| \frac{1}{\|u\|^2} - \frac{1}{\|v\|^2} \right|}_{(II)}.
\end{aligned}$$

We bound (I) and (II) separately. W.L.O.G, assume $\|u\| \geq \|v\|$. Recall $s_{\max}^2 \geq a^\top \boldsymbol{\Sigma}_0^{-1} a \geq s_{\min}^2$ for all $a \in \mathbb{S}^{d-1}$. Since $\|u\| = x^\top \boldsymbol{\Sigma}_t^{-1} x = x^\top (\boldsymbol{\Sigma}_0^{-1} + \sum_{s=1}^t X_s X_s^\top) x$, we have $s_{\min}^2 \leq \|u\| \leq s_{\max}^2 + t$. For (I), we have

$$
\begin{aligned}
(I) &\leq \frac{(\|\boldsymbol{\Gamma}_t u\| - \|\boldsymbol{\Gamma}_t v\|)(\|\boldsymbol{\Gamma}_t u\| + \|\boldsymbol{\Gamma}_t v\|)}{\|u\|^2} \leq \frac{\|\boldsymbol{\Gamma}_t (u-v)\|}{s_{\min}} \left(\frac{\|\boldsymbol{\Gamma}_t u\|}{\|u\|} + \frac{\|\boldsymbol{\Gamma}_t v\|}{\|v\|} \right) \\
&\leq \frac{\|\boldsymbol{\Gamma}_t\| \|u-v\|}{s_{\min}} (2\|\boldsymbol{\Gamma}_t\|) \leq \frac{2\|\boldsymbol{\Gamma}_t\|^2 \|\boldsymbol{\Sigma}_t^{-1/2}\| \iota}{s_{\min}} \leq \frac{2\|\boldsymbol{\Gamma}_t\|^2 \iota \sqrt{s_{\max}^2 + t}}{s_{\min}}.
\end{aligned}
$$

For (II), we have

$$
\begin{aligned}
(II) &\leq \frac{\|\boldsymbol{\Gamma}_t v\|^2}{\|v\|^2} \frac{\|u\|^2 - \|v\|^2}{\|u\|^2} \leq \|\boldsymbol{\Gamma}_t\|^2 \frac{\|u\|^2 - \|v\|^2}{\|u\|^2} \leq \|\boldsymbol{\Gamma}_t\|^2 \frac{(\|u\| - \|v\|)(\|u\| + \|v\|)}{\|u\|^2} \\
&\leq \frac{2\|\boldsymbol{\Gamma}_t\|^2 \|u-v\|}{s_{\min}} \leq \frac{2\|\boldsymbol{\Gamma}_t\|^2 \|\boldsymbol{\Sigma}_t^{-1/2}\| \iota}{s_{\min}} \leq \frac{2\|\boldsymbol{\Gamma}_t\|^2 \iota \sqrt{s_{\max}^2 + t}}{s_{\min}}.
\end{aligned}
$$

Then, putting (I) and (II) together, by the variance-aware discretization argument, we have the spectral norm

$$
\begin{aligned}
\|\boldsymbol{\Sigma}_t^{-1/2} \mathbf{A}_t \mathbf{A}_t^\top \boldsymbol{\Sigma}_t^{-1/2} - \boldsymbol{I}\| &= \sup_{x \in \mathbb{S}^{d-1}} \frac{|x^\top (\boldsymbol{\Sigma}_t^{-1} \mathbf{A}_t \mathbf{A}_t^\top \boldsymbol{\Sigma}_t^{-1} - \boldsymbol{\Sigma}_t^{-1}) x|}{x^\top \boldsymbol{\Sigma}_t^{-1} x} \\
&\leq \frac{4\|\boldsymbol{\Gamma}_t\|^2 \iota \sqrt{s_{\max}^2 + t}}{s_{\min}} + \sup_{y \in \mathcal{C}_\iota} \frac{|y^\top (\boldsymbol{\Sigma}_t^{-1} \mathbf{A}_t \mathbf{A}_t^\top \boldsymbol{\Sigma}_t^{-1} - \boldsymbol{\Sigma}_t^{-1}) y|}{y^\top \boldsymbol{\Sigma}_t^{-1} y} \\
&\leq \frac{4\|\boldsymbol{\Gamma}_t\|^2 \iota \sqrt{s_{\max}^2 + t}}{s_{\min}} + \varepsilon'. \qquad (28)
\end{aligned}
$$

Let

$$
\iota = \frac{\alpha s_{\min}}{4\sqrt{s_{\max}^2 + T}},
$$

where α to be determined. Equivalent formulation of the norm is $\|\boldsymbol{\Gamma}_t\|^2 = \lambda_{\max}(\boldsymbol{\Gamma}_t \boldsymbol{\Gamma}_t^\top)$ and

$$
\|\boldsymbol{\Sigma}_t^{-1/2} \mathbf{A}_t \mathbf{A}_t^\top \boldsymbol{\Sigma}_t^{-1/2} - \boldsymbol{I}\| = \max\{\lambda_{\max}(\boldsymbol{\Gamma}_t \boldsymbol{\Gamma}_t^\top) - 1, 1 - \lambda_{\min}(\boldsymbol{\Gamma}_t \boldsymbol{\Gamma}_t^\top)\}.
$$

Thus, we derive from Equation (28),

$$
\lambda_{\max}(\boldsymbol{\Gamma}_t \boldsymbol{\Gamma}_t^\top) \leq \frac{1+\varepsilon'}{1-\alpha}, \quad \lambda_{\min}(\boldsymbol{\Gamma}_t \boldsymbol{\Gamma}_t^\top) \geq 1 - \varepsilon' - \alpha \lambda_{\max}(\boldsymbol{\Gamma}_t \boldsymbol{\Gamma}_t^\top) \geq 1 - \varepsilon' - \frac{\alpha(1+\varepsilon')}{1-\alpha}.
$$

Claim 1. *If* $\frac{1+\varepsilon'}{1-\alpha} = 1 + \varepsilon$ *and* $\varepsilon' + \frac{\alpha(1+\varepsilon')}{1-\alpha} = \varepsilon$*, then*

$$
1 - \varepsilon \leq \lambda_{\min}(\boldsymbol{\Gamma}_t \boldsymbol{\Gamma}_t^\top) \leq \lambda_{\max}(\boldsymbol{\Gamma}_t \boldsymbol{\Gamma}_t^\top) \leq 1 + \varepsilon.
$$

Let $\varepsilon = 1/2$, then $(\varepsilon', \alpha) = (1/4, 1/6)$ suffices for the Claim 1. That is to say the following configuration for discretization error ι suffices,

$$
\iota = \frac{s_{\min}}{24\sqrt{s_{\max}^2 + T}}.
$$

The covering number is $|\mathcal{C}_\iota| \leq (1 + 2/\iota)^d \leq (1 + (48/s_{\min})\sqrt{s_{\max}^2 + T})^d$. By union bound and define $\delta' = \delta / (1 + (48/s_{\min})\sqrt{s_{\max}^2 + T})^d$, we have

$$
\mathbb{P}\left(\bigcap_{t \in \mathcal{T}} \mathcal{G}_t(1/2) \mid \mathcal{G}_0(1/4) \right) \geq 1 - \delta,
$$

when

$$
M \geq M_2(\delta) := \frac{16(5/4)}{(1/4)^2} \left(d \log \left(\frac{1 + (48/s_{\min})\sqrt{s_{\max}^2 + T}}{\delta} \right) + \log \left(1 + \frac{T}{s_{\min}^2} \right) \right).
$$

Here the constant is 320.

Put things together. When $M \geq M_3 := \max\{M_1(1/2, \delta/2), M_2(\delta/2)\}$, we have

$$\mathbb{P}\left(\bigcap_{t\in\mathcal{T}} \mathcal{G}_t(1/2)\right) = \mathbb{P}\left(\bigcap_{t\in\mathcal{T}} \mathcal{G}_t(1/2) \mid \mathcal{G}_0(1/4)\right)\mathbb{P}\left(\mathcal{G}_0(1/4)\right) \geq (1-\delta/2)^2 \geq 1-\delta.$$

With some calculations, we derive

$$M_1(1/2, \delta/2) = 1024(d\log 9 + \log(4/\delta)),$$

and

$$M_2(\delta/2) = 320\left(d\log\left(\frac{2+(96/s_{\min})\sqrt{s_{\max}^2+T}}{\delta}\right) + \log\left(1+\frac{T}{s_{\min}^2}\right)\right).$$

Since the total time periods T is the dominant growing term, there exist a constant T_0 such that $M_3 = M_2(\delta/2)$ when $T > T_0$.

F Technical Details in Regret Analysis

F.1 General Regret Bound

We start by providing a general analytical framework for agent, potentially randomized, operating in the generic bandit environments. Let us introduce a few necessary definitions to facilitate the understanding and analysis. The confidence bound is used for uncertainty estimation over the ture function f^* given the history $\mathcal{H}_t$.

Definition 4 (Confidence bounds)**.** *Confidence bounds are a sequence of real-valued $\mathcal{H}_t$-measurable functions $\mathrm{L}_t(\cdot)$ and $\mathrm{U}_t(\cdot)$ for $t \in [T]$ such that, w.p. at least $1-\delta$, the joint event $\mathcal{E} = \cap_{t\in[T]}\mathcal{E}_t$ holds, where $\mathcal{E}_t := \{f^*(a) \in [\mathrm{L}_t(a), \mathrm{U}_t(a)], \forall x \in \mathcal{X}_t\}$.*

The agent may not perform well unless it is well-behaved, defined by *reasonableness* and *optimism*. Intuitively, an agent that explores too much or too little will incur a high regret. Reasonableness and optimism are the mechanisms for controlling these potential flaws respectively.

Definition 5 (Reasonableness)**.** *Given confidence bounds $\mathrm{L}_t(\cdot)$ and $\mathrm{U}_t(\cdot)$ for $t \in [T]$, an (randomized) agent is called reasonable if it produces a sequence of functions $(\tilde{f}_t(\cdot), t \in [T])$ such that w.p. at least $1-\delta$, the joint event $\tilde{\mathcal{E}} = \cap_{t\in[T]}\tilde{\mathcal{E}}_t$ holds, where $\tilde{\mathcal{E}}_t := \{\tilde{f}_t(a) \in [\mathrm{L}_t(a), \mathrm{U}_t(a)], \forall x \in \mathcal{X}_t\}$.*

In short, *reasonableness* ensures that the chosen action according to $\tilde{f}_t$ is close to the best action which ensures agent does not explore actions unnecessarily. The following *optimism* guarantees the agent sufficient explores.

Definition 6 (p-optimism)**.** *Let p be a sequence of positive real number $(p_t, t \in [T])$. We say an (randomized) agent is p-optimistic when it produces a sequence of functions $(\tilde{f}_t(\cdot), t \in [T])$ such that for all $t \in [T]$, $\tilde{f}_t(\cdot)$ is p_t-optimistic, i.e., $\mathbb{P}(\max_{x\in\mathcal{X}_t} \tilde{f}_t(a) \geq \max_{x\in\mathcal{X}_t} f^*(a) \mid \mathcal{H}_t) \geq p_t$.*

The generic agent satisfying the conditions on *reasonableness* and *optimism* has desired behavior.

Building upon the definitions of Reasonableness and Optimism, we establish a general regret bound applicable to any agent satisfying these conditions.

Theorem 3 (General Regret Bound)**.** *Given confidence bounds as defined in Definition 4, and assuming the agent is both reasonable and p-optimistic, the cumulative regret over T time steps satisfies*

$$R(T) \leq \sum_{t=1}^{T} \frac{1}{p_t}\mathbb{E}\left[U_t(X_t) - L_t(X_t) \mid \mathcal{H}_t\right] + \sum_{t=1}^{T}\left(U_t(X_t) - L_t(X_t)\right), \tag{29}$$

with probability at least $1-\delta$.

Interpretation The regret bound in equation 29 decomposes into two main components:

1. **Exploration-Exploitation Trade-off:** The first term scales with $\frac{1}{p_t}$ and the expected width of the confidence bounds. A higher p_t (i.e., greater optimism) reduces this component, promoting exploration.
2. **Confidence Bound Widths:** The second term aggregates the widths of the confidence intervals across all time steps, reflecting the uncertainty inherent in the agent's estimates.

For the regret to be sublinear in T, it is essential that the confidence bounds $U_t(a) - L_t(a)$ shrink appropriately as t increases, ensuring that both terms grow slower than linearly with T.

Proof. Let $X_t = \max_{x \in \mathcal{X}_t} \tilde{f}_t(a)$ and $A_t^* = \max_{x \in \mathcal{X}_t} f^*(a)$. Let $B_t = \max_{x \in \mathcal{A}_t} L_t(a)$, which is $\mathcal{H}_t$-measurable. Conditioned on the event $\mathcal{E} \cap \tilde{\mathcal{E}}$, both $f^*(X_t^*) \geq B_t$ and $\tilde{f}_t(X_t) \geq B_t$ hold. By p-optimism and the fact $(f^*(X_t^*) - B_t)$ is $\mathcal{H}_t$-measurable and positive,

$$p_t \leq \mathbb{P}(f_t(X_t) - B_t \geq f^*(X_t^*) - B_t \mid \mathcal{H}_t) \overset{(*)}{\leq} \mathbb{E}[f_t(X_t) - B_t \mid \mathcal{H}_t]/(f^*(X_t^*) - B_t),$$

where $(*)$ is due to Markov inequality. Rearranging and using the additional fact $B_t \geq L_t(X_t)$ yield

$$f^*(X_t^*) - \tilde{f}_t(X_t) \leq f^*(X_t^*) - B_t \leq \frac{1}{p_t}\mathbb{E}[f_t(X_t) - B_t \mid \mathcal{H}_t] \leq \frac{1}{p_t}\mathbb{E}[U_t(X_t) - L_t(X_t) \mid \mathcal{H}_t]. \tag{30}$$

By the reasonableness, $\tilde{f}_t(X_t) \leq U_t(X_t)$. Then, from the definition of confidence bounds

$$\tilde{f}_t(X_t) - f^*(X_t) \leq U_t(X_t) - L_t(X_t) \tag{31}$$

Putting Equations (30) and (31) together and then summing over the time index t yields the general regret upper bound. □

F.2 Proof of Theorem 1 for linear contextual bandits

To make the proof easy to access, we restate the core results and a few notations that is needed for the proof of the propositions.

P_ζ	Gaussian $N(0, I_M)$	Sphere $\sqrt{M}\mathcal{U}(\mathbb{S}^{M-1})$	Cube $\mathcal{U}(\{1,-1\}^M)$	Coord $\mathcal{U}(\{\pm e_i\}_{i\in[M]})$	Sparse
$\rho(P_\zeta)$	$\rho_1 \wedge \rho_3$	$\rho_2 \wedge \rho_3$	$\rho_2 \wedge \rho_3$	ρ_2	ρ_2
$p(P_\zeta)$	$\frac{1}{4\sqrt{e\pi}}$	$\frac{1}{2} - \frac{e^{1/12}}{\sqrt{2\pi}}$	$7/32$	$\frac{1}{2M}$	N/A

Table 4: (Restate of Table Table 2) The coefficient $\rho(P_\zeta)$ and $p(P_\zeta)$ related to reasonableness and optimism condition.

Adapting the results from (Abbasi-Yadkori et al., 2011b; Abeille & Lazaric, 2017), let $\beta_t = \sqrt{\lambda} + \sqrt{2\log(1/\delta) + \log\det(\boldsymbol{\Sigma}_{t-1}^{-1}/\lambda^d)}$. Under assumption 1, we define the confidence bound as

$$L_t(\cdot) = (-1) \vee (\langle \mu_{t-1}, \phi(\cdot)\rangle - \beta_t \|\phi(\cdot)\|_{\boldsymbol{\Sigma}_{t-1}}), U_t(\cdot) = 1 \wedge (\langle \mu_{t-1}, \phi(\cdot)\rangle + \beta_t \|\phi(\cdot)\|_{\boldsymbol{\Sigma}_{t-1}})$$

For the purpose of analysis within various reference distribution, we define a slightly inflated confidence bounds as

$$L_t(\cdot; P_\zeta) = (\langle \mu_{t-1}, \phi(\cdot)\rangle - \beta_t \rho(P_\zeta) \|\phi(\cdot)\|_{\boldsymbol{\Sigma}_{t-1}}) \vee (-1),$$
$$U_t(\cdot; P_\zeta) = (\langle \mu_{t-1}, \phi(\cdot)\rangle + \beta_t \rho(P_\zeta) \|\phi(\cdot)\|_{\boldsymbol{\Sigma}_{t-1}}) \wedge 1.$$

$\rho(P_\zeta)$ is defined via $\rho_1 = O(\sqrt{M\log(M/\delta)})$, $\rho_2 = O(\sqrt{M})$, and $\rho_3 = O(\sqrt{\log(|\mathcal{X}|/\delta)})$ and Table 2. An immediate observation is that $[L_t(\cdot), U_t(\cdot)] \subset [L_t(\cdot; P_\zeta), U_t(\cdot; P_\zeta)]$. Thus, $L_t(\cdot; P_\zeta)$ and $U_t(\cdot; P_\zeta)]$ are also confidence bounds. We consider the the following functional form for Ensemble++ under linear setup: for time t,

$$\tilde{f}_t(a) := f_{\theta_t}(a, \zeta_t) = \langle \phi(a), \beta_t \mathbf{A}_{t-1}\zeta_t + \mu_{t-1} \rangle, \quad \forall x \in \mathcal{X}, \tag{32}$$

where the parameters include $\theta_t = (\mathbf{A}_t, \mu_t)$.

The condition on the propositions and theorem for regret analysis is when Equation (18) is satisfied, that is when $M = \Theta(d \log T)$, the Lemma 1 implies that with high probability, the good events $\mathcal{G} = \bigcap_{t=0}^{T} \mathcal{G}_t$ hold jointly, where

$$\mathcal{G}_t := \left\{ \frac{1}{2} x^\top \boldsymbol{\Sigma}_t x \le x^\top \mathbf{A}_t \mathbf{A}_t^\top x \le \frac{3}{2} x^\top \boldsymbol{\Sigma}_t x, \quad \forall x \in \mathbb{R}^d \right\}.$$

In the following section, we discuss the proof conditioned on the joint event $\mathcal{G}$ and also the confidence event that $f^*(a) \in [L_t(a), U_t(a)]$ for all $t \in [T]$ and $x \in \mathcal{X}$.

F.2.1 Proof of Proposition 1

Notice that from Equation (32), we derive

$$\begin{aligned}
|\tilde{f}_t(a) - \langle \mu_{t-1}, \phi(a) \rangle| &= |\langle \phi(a), \beta_t \mathbf{A}_{t-1}\zeta_t \rangle| \\
&= \beta_t \sqrt{\phi(a)^\top \mathbf{A}_{t-1}\mathbf{A}_{t-1}^\top \phi(a)} \left| \left\langle \frac{\phi(a)^\top \mathbf{A}_{t-1}}{\|\phi(a)^\top \mathbf{A}_{t-1}\|}, \zeta_t \right\rangle \right| \\
&\le (3/2)\beta_t \sqrt{\phi(a)^\top \boldsymbol{\Sigma}_{t-1}\phi(a)} \left| \left\langle \frac{\phi(a)^\top \mathbf{A}_{t-1}}{\|\phi(a)^\top \mathbf{A}_{t-1}\|}, \zeta_t \right\rangle \right|,
\end{aligned}$$

where the last inequality is due to the good event $\mathcal{G}$. For compact action set, we use Cauchy–Schwarz inequality,

$$\left| \left\langle \frac{\phi(a)^\top \mathbf{A}_{t-1}}{\|\phi(a)^\top \mathbf{A}_{t-1}\|}, \zeta_t \right\rangle \right| \le \|\zeta_t\|.$$

Using the concentration properties of P_ζ in Appendix G to upper bound $\|\zeta_t\|$ yields part of the results. For finite action set $\mathcal{X}$, also taking the advantages of the concentration properties of several reference distributions P_ζ in Appendix G to bound the conditionally probability

$$\mathbb{P}\left(\left| \left\langle \frac{\phi(a)^\top \mathbf{A}_{t-1}}{\|\phi(a)^\top \mathbf{A}_{t-1}\|}, \zeta_t \right\rangle \right| \le \sqrt{\log \frac{2|\mathcal{X}|}{\delta}} \mid \mathcal{H}_t, \mathcal{Z}_t \right) \ge 1 - \delta,$$

as ξ_t is independent of the history $\mathcal{H}_t, \mathcal{Z}_t$. Finally, the inflated coefficient $\rho(P_\zeta)$ defined in Table 2 suffices to make $\tilde{f}_t(\cdot) \in [L_t(\cdot; P_\zeta), U_t(\cdot; P_\zeta)]$ reasonable.

Proposition 1. *Under linear setups in Equations (3) and (32), if Equation (18) is satisfied, linear Ensemble++ is reasonable, i.e.,* $\forall t \in [T], \tilde{f}_t(\cdot) = f_{\theta_t}(\cdot, \zeta_t) \in [L_t(\cdot; P_\zeta), U_t(\cdot; P_\zeta)]$ *w.p.* $1 - \delta$.

Proposition 2. *Under linear setups in Equations (3) and (32), if Equation (18) is satisfied, linear Ensemble++ using reference distribution* P_ζ *is* $p(P_\zeta)$*-optimistic.*

F.2.2 Proof of Proposition 2

Let $X_t = \max_{x \in \mathcal{X}_t} \tilde{f}_t(a)$ and $A_t^* = \max_{x \in \mathcal{X}_t} f^*(a)$. Conditioned on $\mathcal{G}$ and confidence event,

$$\begin{aligned}
\tilde{f}_t(X_t) - f^*(A_t^*) &\ge \tilde{f}_t(A_t^*) - f^*(A_t^*) \ge \tilde{f}_t(A_t^*) - U^*(A_t^*) \\
&= \langle \phi(A_t^*), 2\beta_t \mathbf{A}_{t-1}\zeta_t \rangle - \beta_t \|A_t^*\|_{\boldsymbol{\Sigma}_{t-1}} \\
&= 2\beta_t \sqrt{\phi(A_t^*)^\top \mathbf{A}_{t-1}\mathbf{A}_{t-1}^\top \phi(A_t^*)} \left\langle \frac{\phi(A_t^*)^\top \mathbf{A}_{t-1}}{\|\phi(A_t^*)^\top \mathbf{A}_{t-1}\|}, \zeta_t \right\rangle - \beta_t \|\phi(A_t^*)\|_{\boldsymbol{\Sigma}_{t-1}} \\
&\ge \beta_t \|\phi(A_t^*)\|_{\boldsymbol{\Sigma}_{t-1}} \left(\left\langle \frac{\phi(A_t^*)^\top \mathbf{A}_{t-1}}{\|\phi(A_t^*)^\top \mathbf{A}_{t-1}\|}, \zeta_t \right\rangle - 1 \right).
\end{aligned}$$

We consider the conditional probability,

$$\begin{aligned}
\mathbb{P}(\tilde{f}_t(X_t) \ge f^*(A_t^*) \mid \mathcal{H}_t, \boldsymbol{Z}_t) &\ge \mathbb{P}\left(\beta_t \|\phi(A_t^*)\|_{\boldsymbol{\Sigma}_{t-1}} \left(\left\langle \frac{\phi(A_t^*)^\top \mathbf{A}_{t-1}}{\|\phi(A_t^*)^\top \mathbf{A}_{t-1}\|}, \zeta_t \right\rangle - 1 \right) \mid \mathcal{H}_t, \mathcal{Z}_t \right) \\
&= \mathbb{P}(\langle v, \zeta_t \rangle \ge 1),
\end{aligned} \tag{33}$$

where v is a fixed unit vector in $\mathbb{R}^M$. The final probability bound in *Equation* (33) for each reference distribution P_ζ is essentially the anti-concentration bounds. Please find the anti-concentration results for each distribution in Appendix G, resulting in the Table 2.

Proof. The Theorem 1 follows directly from Propositions 1 and 2 and Theorem 3. Additionally, it requires the Azuma's inequality for the sum of bounded martingale difference: as $U_t(\cdot) - L_t(\cdot) \leq 2$ is bounded, we have

$$\sum_{t\in[T]} \mathbb{E}[(U_t(X_t) - L_t(X_t)) \mid \mathcal{H}_t] - (U_t(X_t) - L_t(X_t)) \leq O(\sqrt{T\log(1/\delta)}),$$

with probability at least $1-\delta$.

Then, it suffices to bound the summation of width between upper and lower confidence bounds

$$\sum_{t\in[T]} (U_t(X_t) - L_t(X_t)) \leq \rho(P_\zeta) \sum_{t\in[T]} 2\beta_t \|\phi(X_t)\|_{\boldsymbol{\Sigma}_{t-1}},$$

which depends on a distribution-dependent coefficient $\rho(P_\zeta)$. Under linear bandit setups in assumption 1, we use the elliptical potential lemma (e.g. Lemma 19.4 in (Lattimore & Szepesvári, 2020) and (Abbasi-Yadkori et al., 2011a)) to bound this summation. □

G Sampling, Isotropy, Concentration and Anti-concentration

Definition 7 (Isotropic)**.** *A distribution P over $\mathbb{R}^M$ is called isotropic if $\mathbb{E}_{X\sim P}[X_iX_j] = \delta_{ij}$, i.e., $\mathbb{E}_{X\sim P}[XX^\top] = I$. Equivalently, P is isotropic if $\mathbb{E}_{X\sim P}[\langle X, x\rangle^2] = \|x\|^2$, for all $x \in \mathbb{R}^M$.*

Isotropy property (Definition 7) is used for update distribution and proving the Equation (3). The sub-Gaussianness (Definition 2) in concentration property is used for perturbation distributions and proving Lemma 1. The concentration and anti-concentration properties are used for reference distributions and discussion on the reasonableness condition (Proposition 1) and optimism condition (Proposition 2).

Let us discuss each distribution case by case.

G.1 Sphere $P_\zeta = \mathcal{U}(\sqrt{M}\mathbb{S}^{M-1})$

Algorithm 3 Symmetric Index Sampling for $\mathcal{U}(\sqrt{M}\mathbb{S}^{M-1})$

Input: Number of ensemble members M
1: Sample vector v: $v_i \sim N(0,1)$ for $i = 1,\dots,M$
2: Construct index vector: $\xi = \sqrt{M}v/\|v\|$
3: **Return** ξ

Isotropy. By the rotational invariance of sphere distribution, we know for any fixed orthogonal matrix Q,

$$\langle \zeta, x\rangle \sim \langle Q\zeta, x\rangle = \langle \zeta, Q^\top x\rangle, \quad \forall x \in \mathbb{R}^d.$$

Then, for any fixed x, we select M orthogonal matrix $Q_1,\dots,Q_M$ to rotate x such that $Q_i^\top x = \|x\|e_i$ where e_i is the i-th coordinate vector. With this construction, for any fixed x,

$$M\mathbb{E}[\langle \zeta, x\rangle^2] = \mathbb{E}[\sum_{i=1}^M \langle \zeta, x_i\rangle^2] = \mathbb{E}[\|x\|^2 \sum_{i=1}^M \zeta_i^2] = M\|x\|^2$$

and hence $\mathbb{E}[\langle \zeta, x\rangle^2] = \|x\|^2$, which is the definition of isotropic random vector.

Concentration. By definition, $\|\zeta\| = \sqrt{M}$. For a random variable $\zeta \sim \mathcal{U}(\mathbb{S}^{M-1})$ and any fixed $v \in \mathbb{S}^{M-1}$, the inner product follows the transformed Beta distribution

$$\langle \zeta, v\rangle \sim 2\,\mathrm{Beta}(\frac{M-1}{2}, \frac{M-1}{2}) - 1.$$

Evidenced by (Skorski, 2023; Li, 2024a), $P_\zeta = \mathcal{U}(\sqrt{M}\mathbb{S}^{M-1})$ is 1-sub-Gaussian. For finite action set $\mathcal{A}$, using the concentration of Beta random variables with union bound, we have

$$\mathbb{P}\left(\forall a \in \mathcal{A}, \langle \zeta, \phi(a)\rangle \leq \|\phi(a)\|\sqrt{\log\frac{2|\mathcal{A}|}{\delta}}\right) \geq 1-\delta,$$

Anti-concentration. Let's start by rewriting the problem in terms of the incomplete Beta function:

Given:

$$X \sim \text{Beta}\left(\frac{M-1}{2}, \frac{M-1}{2}\right)$$

We want to find:

$$\text{P}\left(\langle \zeta, v\rangle \geq 1\right) = \text{P}\left(2X - 1 > \frac{1}{\sqrt{M}}\right) = \text{P}\left(X > \frac{1}{2} + \frac{1}{2\sqrt{M}}\right).$$

Theorem 4. *For all $d \geq 2$, the random variable $X \sim$ Beta $\left(\frac{d-1}{2}, \frac{d-1}{2}\right)$ has the following anti-concentration behavior*

$$P\left(X > \frac{1}{2} + \frac{1}{2\sqrt{d}}\right) \geq \frac{1}{2} - \frac{e^{1/12}}{\sqrt{2\pi}}.$$

Remark 5. *We did not find any literature that can help derive such anti-concentration results for Beta distribution.*

Proof. Using the incomplete Beta function $I_x(a,b)$, this probability can be expressed as:

$$\text{P}\left(X > \frac{1}{2} + \frac{1}{2\sqrt{d}}\right) = 1 - I_{\left(\frac{1}{2}+\frac{1}{2\sqrt{d}}\right)}\left(\frac{d-1}{2}, \frac{d-1}{2}\right)$$

To compute $I_{\left(\frac{1}{2}+\frac{1}{2\sqrt{d}}\right)}\left(\frac{d-1}{2}, \frac{d-1}{2}\right)$, we will use the following relationship for the regularized incomplete Beta function $I_x(a,b)$:

$$I_x(a,b) = \frac{B(x;a,b)}{B(a,b)}$$

where $B(x;a,b)$ is the incomplete Beta function and $B(a,b) := B(1;a,b)$ is the complete Beta function.

For $a = b = \frac{d-1}{2}$, the complete Beta function is:

$$B\left(\frac{d-1}{2}, \frac{d-1}{2}\right) = \frac{\Gamma\left(\frac{d-1}{2}\right)\Gamma\left(\frac{d-1}{2}\right)}{\Gamma(d-1)}$$

Using the property of the Gamma function:

$$\Gamma(n+1) = n\Gamma(n).$$

Let's compute the incomplete Beta function for $x = \frac{1}{2} + \frac{1}{2\sqrt{d}}$ and $a = b = \frac{d-1}{2}$:

1. Calculate the incomplete Beta function $B\left(x; \frac{d-1}{2}, \frac{d-1}{2}\right)$:

$$B\left(\frac{1}{2} + \frac{1}{2\sqrt{d}}; \frac{d-1}{2}, \frac{d-1}{2}\right) = \int_0^{\frac{1}{2}+\frac{1}{2\sqrt{d}}} t^{\frac{d-3}{2}}(1-t)^{\frac{d-3}{2}}\, dt$$

As $f(t) = t^{\frac{d-3}{2}}(1-t)^{\frac{d-3}{2}}$ is symmetric at $t = 1/2$ in the interval $[0,1]$,

$$B\left(\frac{1}{2} + \frac{1}{2\sqrt{d}}; \frac{d-1}{2}, \frac{d-1}{2}\right) = \frac{1}{2}B\left(\frac{d-1}{2}, \frac{d-1}{2}\right) + \int_{\frac{1}{2}}^{\frac{1}{2}+\frac{1}{2\sqrt{d}}} t^{\frac{d-3}{2}}(1-t)^{\frac{d-3}{2}}\, dt.$$

2. Calculate the regularized incomplete Beta function $I_x(a,b)$:

$$I_{\left(\frac{1}{2}+\frac{1}{2\sqrt{d}}\right)}\left(\frac{d-1}{2},\frac{d-1}{2}\right)=\frac{B\left(\frac{1}{2}+\frac{1}{2\sqrt{d}};\frac{d-1}{2},\frac{d-1}{2}\right)}{B\left(\frac{d-1}{2},\frac{d-1}{2}\right)}$$

As the function $f(t)=t^{\frac{d-3}{2}}(1-t)^{\frac{d-3}{2}}$ achieves the maximum at $t=1/2$, we could upper bound the incomplete Beta function by

$$\int_{\frac{1}{2}}^{\frac{1}{2}+\frac{1}{2\sqrt{d}}} t^{\frac{d-3}{2}}(1-t)^{\frac{d-3}{2}}\,dt \le \left(\frac{1}{4}\right)^{\frac{d-3}{2}}\left(\frac{1}{2\sqrt{d}}\right)=\left(\frac{1}{2}\right)^{d-3}\left(\frac{1}{2\sqrt{d}}\right). \tag{34}$$

The complete Beta function can be expressed as

$$B\left(\frac{d-1}{2},\frac{d-1}{2}\right)=\frac{\Gamma\left(\frac{d-1}{2}\right)\Gamma\left(\frac{d-1}{2}\right)}{\Gamma(d-1)},$$

where $\Gamma(\cdot)$ is the Gamma function. We use the Stirling's Approximation on Gamma function which could provide strict lower bound(Nemes, 2015)

$$\Gamma(z)\ge\sqrt{2\pi}z^{z-\frac{1}{2}}e^{-z},$$

and upper bound (Gronwall, 1918)

$$\Gamma(z)\le\sqrt{2\pi}z^{z-\frac{1}{2}}e^{-z+\frac{1}{12z}}$$

for all $z>0$. Immediately, the lower bound of the complete Beta function is

$$B\left(\frac{d-1}{2},\frac{d-1}{2}\right)\ge\frac{\sqrt{2\pi}((d-1)/2)^{d-2}e^{-(d-1)}}{(d-1)^{d-\frac{3}{2}}e^{-d+1+\frac{1}{12(d-1)}}}=\sqrt{2\pi}\left(\frac{1}{2}\right)^{d-2}(d-1)^{-1/2}e^{-\frac{1}{12(d-1)}}.$$

As $e^{-\frac{1}{12(d-1)}}\ge e^{-1/12}$ whenever $d\ge 2$, we further lower bound

$$B\left(\frac{d-1}{2},\frac{d-1}{2}\right)\ge\sqrt{2\pi}e^{-1/12}\left(\frac{1}{2}\right)^{d-2}\frac{1}{\sqrt{d}}. \tag{35}$$

Finally, combining Equations (34) and (35) yields

$$\begin{aligned}I_{\left(\frac{1}{2}+\frac{1}{2\sqrt{d}}\right)}\left(\frac{d-1}{2},\frac{d-1}{2}\right)&\le\frac{1}{2}+\frac{2e^{1/12}(\frac{1}{2\sqrt{d}})}{\sqrt{2\pi}\frac{1}{\sqrt{d}}}\\&\le\frac{1}{2}+\frac{e^{1/12}}{\sqrt{2\pi}},\end{aligned}$$

and

$$P(X>\frac{1}{2}+\frac{1}{2\sqrt{d}})\ge\frac{1}{2}-\frac{e^{1/12}}{\sqrt{2\pi}}\approx 0.0668.$$

□

G.2 CUBE $P_\zeta=\mathcal{U}(\{1,-1\}^M)$

Algorithm 4 Symmetric Index Sampling for $\mathcal{U}(\{-1,1\}^M)$

Input: Number of ensemble members M
1: Sample vector ξ: $\xi_i\sim\mathcal{U}(\{-1,1\})$ for $i=1,\dots,M$
2: **Return** ξ

Isotropy. Easy to verify by definition.

Concentration. By definition, $\|\xi\| = \sqrt{M}$. Also notice that we could sample the random vector ζ by sample each entry independently from $\zeta_i \sim \mathcal{U}(\{1, -1\})$ for $i \in [M]$. Then, for any $v \in \mathbb{S}^{M-1}$, by independence,

$$\mathbb{E}[\exp(\lambda\langle v, \zeta\rangle)] = \prod_{i=1}^{m} \mathbb{E}[\exp(\lambda v_i \mathrm{z}_i)] \leq \prod_{i=1}^{m} \exp(\lambda^2 v_i^2) = \exp(\lambda^2 \sum_i v_i^2).$$

The inequality is due to MGF of rademacher distribution (e.g. Example 2.3 in (Wainwright, 2019)). Then we confirm that $P_\zeta = \mathcal{U}(\{1, -1\}^M)$ is 1-sub-Gaussian. For finite action set $\mathcal{A}$, we have from sub-Gaussian property

$$\mathbb{P}\left(\forall a \in \mathcal{A}, \langle \zeta, \phi(a)\rangle \leq \|\phi(a)\|\sqrt{\log \frac{2|\mathcal{A}|}{\delta}}\right) \geq 1 - \delta.$$

Anti-concentration. Using the anti-concentration result from (Hollom & Portier, 2023), we have for any fixed unit vector v in $\mathbb{R}^M$

$$P(\langle \zeta, v\rangle) \geq 7/32 \approx 0.21875.$$

G.3 GAUSSIAN $P_\zeta = N(0, I_M)$

Algorithm 5 Symmetric Index Sampling for $N(0, \boldsymbol{I}_M)$

Input: Number of ensemble members M
1: Sample vector ξ: $\xi_i \sim N(0, 1)$ for $i = 1, \ldots, M$
2: **Return** ξ

Isotropy. Easy to verify by definition.

Concentration. The concentration property comes directly from the Chernoff bound for standard Gaussian random variable together with union bound argument. For any $\alpha > 0$, we have

$$\mathbb{P}(\|\zeta\| \leq \alpha\sqrt{M}) \geq \mathbb{P}\left(\forall 1 \leq i \leq M, |\zeta_i| \leq \alpha\right) \geq 1 - M\mathbb{P}\left(|\zeta_i| \geq \alpha\right).$$

Standard concentration inequality for Gaussian random variable gives, $\forall \alpha > 0$,

$$\mathbb{P}\left(|\zeta_i| \geq \alpha\right) \leq 2e^{-\alpha^2/2}.$$

Plugging everything together with $\alpha = \sqrt{2\log\frac{2M}{\delta}}$ gives the desired result, which is

$$\|\zeta\| \leq \sqrt{2M\log\frac{2M}{\delta}}, \quad w.p.\ 1 - \delta.$$

For the case of finite action set $\mathcal{A}$,

$$\mathbb{P}\left(\forall a \in \mathcal{A}, \langle \zeta, \phi(a)\rangle \leq \|\phi(a)\|\sqrt{\log \frac{2|\mathcal{A}|}{\delta}}\right) \geq 1 - \delta.$$

Anti-concentration. Here $\langle \zeta, v\rangle \sim N(0, 1)$ for for any fixed unit vector v in $\mathbb{R}^M$.

$$P(N(0, 1) \geq 1) = \frac{1}{2}\operatorname{erfc}\left(\frac{1}{\sqrt{2}}\right) \geq \frac{1}{4\sqrt{e\pi}} \approx 0.0856$$

G.4 COORD $P_\zeta = \mathcal{U}(\sqrt{M}\{\pm e_1, \ldots, \pm e_M\})$

Isotropy. Easy to verify by definition,

$$\mathbb{E}[\zeta\zeta^\top] = \frac{1}{2M}\sum_{i=1}^{M} 2Me_ie_i^\top = \boldsymbol{I}. \tag{36}$$

Algorithm 6 Symmetric Index Sampling for $\mathcal{U}(\sqrt{M}\{\pm e_1, \ldots, \pm e_M\})$

Input: Number of ensemble members M
1: Sample index: $i \sim \mathcal{U}(\{1, \ldots, M\})$
2: Sample sign: $s \sim \mathcal{U}(\{-1, 1\})$
3: Construct index vector: $\xi = s\sqrt{M}e_i$
4: **Return** ξ

Concentration. By definition, $\|\zeta\| = \sqrt{M}$.

Anti-concentration.

$$P(\langle \zeta, v \rangle \geq 1) = \frac{1}{2M} \sum_{j \in [M]} (\mathbb{1}_{v_j \geq \frac{1}{\sqrt{M}}} + \mathbb{1}_{-v_j \geq \frac{1}{\sqrt{M}}}) = \frac{1}{2M} \sum_{j \in [M]} (\mathbb{1}_{|v_j| \geq \frac{1}{\sqrt{M}}}) \geq \frac{1}{2M},$$

where the last inequality is due to a simple fact that for any fixed $v \in \mathbb{R}^M$ with unit norm $\|v\| = 1$, there always exists an entry $j \in [M]$ with $|v_j| \geq \frac{1}{\sqrt{M}}$.

G.5 Sparse distribution P_ζ

Algorithm 7 Symmetric Index Sampling for s-sparse random vector

Input: Number of ensemble members M, sparsity s
1: Sample sign: $\omega_i \sim \mathcal{U}(\{-1, 1\})$ for $i = 1, \ldots, M$
2: Construct a set $\mathcal{S}$ by randomly pick s elements from $\{1, \ldots, M\}$ without replacement
3: Let $\eta_i = 1$ for $i \in \mathcal{S}$ and $\eta_{i'} = 0$ for $i' \in \{1, \ldots, M\} \setminus \mathcal{S}$
4: Construct index vector ξ: $\xi_i = \omega_i \cdot \eta_i$
5: **Return** ξ

Definition 8 (s-sparse distribution)**.** *The sparse vector is in the form $\zeta = \sqrt{\frac{M}{s}}\eta \odot \omega$ where $P_\omega := \mathcal{U}(\{1, -1\}^M)$, and η is independently and uniformly sampled from all possible s-hot vectors, where s-hot vectors is with exactly s non-zero entries with number 1. This construction is introduced by (Kane & Nelson, 2014).*

Isotropy. By definition,

$$\mathbb{E}[\zeta_j \zeta_k] = \frac{M}{s}\mathbb{E}[\eta_j \eta_k]\mathbb{E}[\omega_j \omega_k] = \frac{M}{s}\delta_{jk}\mathbb{E}[\omega_j] = \delta_{hk}. \tag{37}$$

Therefore, the sparse distribution in Definition 8 is indeed isotropic distribution.

Concentration. $\|\zeta\| = \sqrt{M}$.

Anti-concentration. Not clear.

H IN-DEPTH EMPIRICAL AND ABLATION STUDIES

In this section, we dive into the intricacies of each evaluation testbed. Through a comprehensive set of empirical results, we'll further illuminate the benefits afforded by Ensemble++. All experiments are conducted on P40 GPUs to maintain processing standardization.

H.1 ADDITIONAL EXPERIMENTS ON LINEAR BANDIT

We begin by examining Linear Ensemble++ Sampling in linear bandits. In this section, we focus on studying the impact of perturbation and reference distributions, and we provide detailed results under varying numbers of ensembles M.

Environment Settings: We use the action feature set $\mathcal{X}$ to denote the set of features $\phi(a) : a \in \mathcal{A}$ induced by action set $\mathcal{A}$ and feature mapping $\phi(\cdot)$. We build two linear bandit environments with different action distribution as follow:

- **Finite-action Linear Bandit**: We construct the finite set $\mathcal{X}$ by uniformly sampling a set of action features from the range $[-1/\sqrt{5}, 1/\sqrt{5}]^d$ where d is the ambient dimension of the linear reward function. This environment builds upon prior research Russo & Van Roy (2018). We vary the action size $|\mathcal{X}|$ over a set of $\{100, 1000, 10000\}$, and the ambient dimension across $\{10, 50\}$.
- **Compact-action Linear Bandit**: Let the action feature set $\mathcal{X} = \mathbb{S}^{d-1}$ be the unit sphere. In this environment, we vary the ambient dimension d over a set of $\{10, 50, 100\}$.

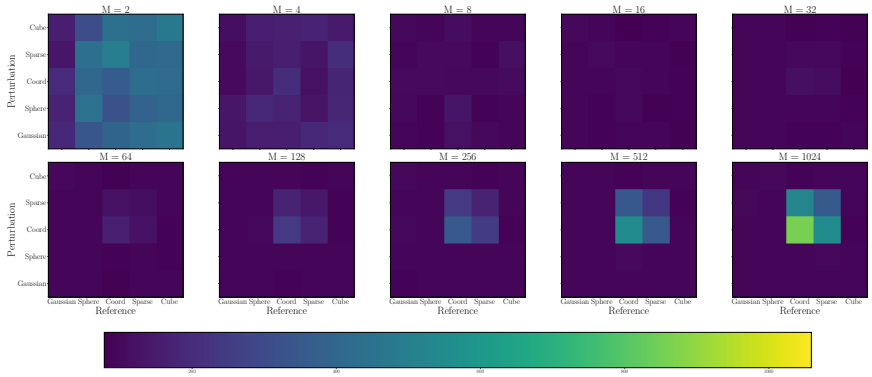
In both environments, the reward of each feature $X_t \in \mathbb{R}^d$ is computed as $r_t = X_t^\top \theta + \epsilon$, where $\theta \sim \mathcal{N}(0, 10\mathrm{I})$ is drawn from the multivariate Gaussian prior distribution, and $\epsilon \sim \mathcal{N}(0, 1)$ is an independent additive Gaussian noise term. At every step t, only the reward from the chosen feature X_t is discernible. To ensure robust results, each experiment is executed a total of 1000 time steps and repeated 200 times.

Impact of Reference and Perturbation Distributions: We investigated all 25 combinations of perturbation and reference distribution under different scales of the linear bandit environments and numerous #ensembles M. As depicted in Figures 11 to 13, the outcomes across diverse problem scales corroborate each other. **The use of a Gaussian reference distribution significantly enhances performance when the M is relatively small, such as when M is 2 or 4.** As the #ensembles M grows, all combinations show an analogous performance under varying problem scales. However, it is worth noting that for extremely large M, such as 512 or 1024, combinations involving the Coordinate perturbation and Coordinate reference distribution significantly underperform compared to other combinations. Given that Coordinate distributions are used in the Ensemble+, the results prompt a compelling argument. Linear Ensemble++ Sampling equipped with a continuous reference distribution presents a superior performance, suggesting its potential for surpassing traditional Linear Ensemble Sampling. These findings strongly support the superior advantage of our index sampling method, validating our theoretical analysis.

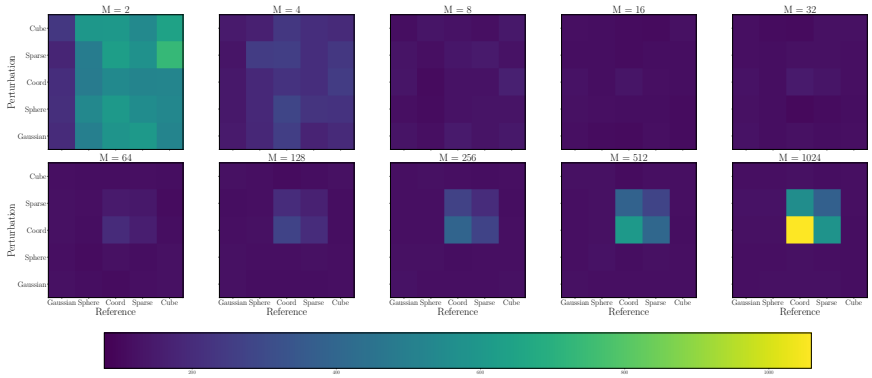
Analysis of Computational Efficiency: We delve deeper into the effects of varying #ensembles M within Linear Ensemble++ Sampling. We assess its performance across different combinations of perturbation and reference distributions using an assortment of $M \in \{4, 8, 16, 32, 64, 128, 256, 512, 1024\}$. The outcomes, visualized in Figures 14 and 15, are consistent with the findings illustrated in Figures 11 to 13. We observe that for large M, the Coordinate perturbation and Coordinate reference distributions degrade performance, indicating that the index sampling method employed by Ensemble+ lacks efficiency. However, when Linear Ensemble++ Sampling utilizes Gaussian or Sphere reference distributions, it achieves satisfactory performance, comparable to Thompson Sampling with small M.

Remark 6 (Limitation of Theorem 1.)**.** *Notice that Theorem 1 suggest that when $M \geq O(d \log T)$, the regret bound of Linear Ensemble sampling would increase with factor $M^{3/2}$, which contradicts with our empirical evidence in Figures 11 to 15.*

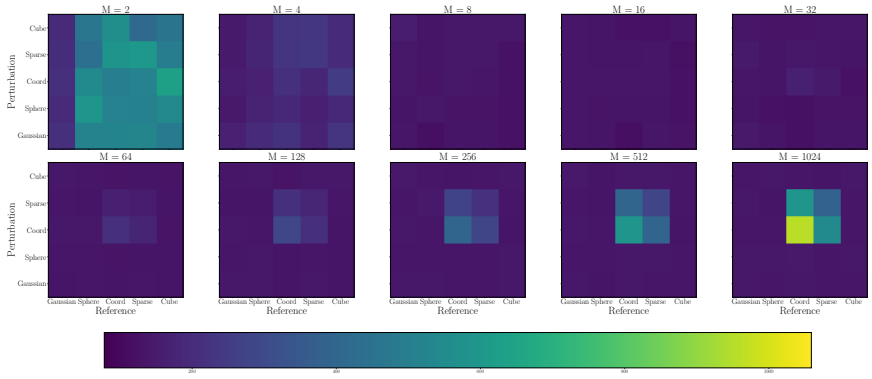
Remark 7 (Good prediction of Theorem 1.)**.** *Our empirical evidence in Figures 11 to 15 confirms the Theorem 1 in finite decision set setting for continuous-support reference distributions: when M is larger then a threshold $O(d \log T)$, the regret has no dependence on M.*



(a) $d = 10 \quad |\mathcal{X}| = 100$

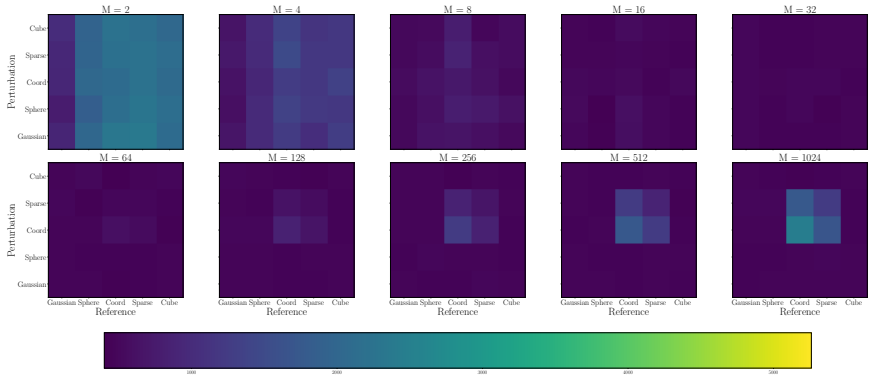


(b) $d = 10 \quad |\mathcal{X}| = 1000$

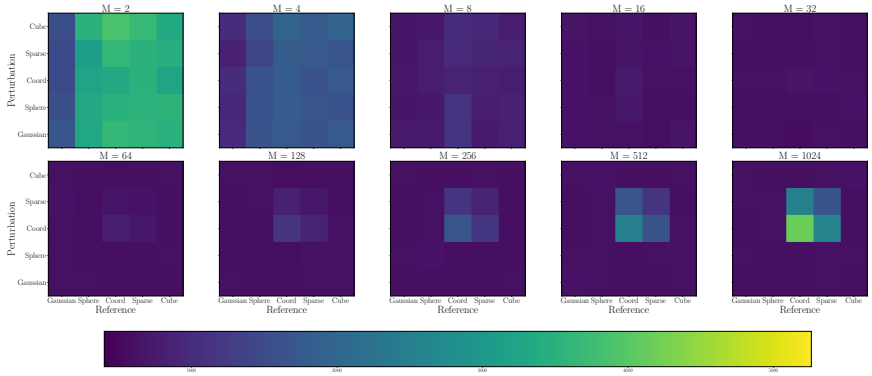


(c) $d = 10 \quad |\mathcal{X}| = 10000$

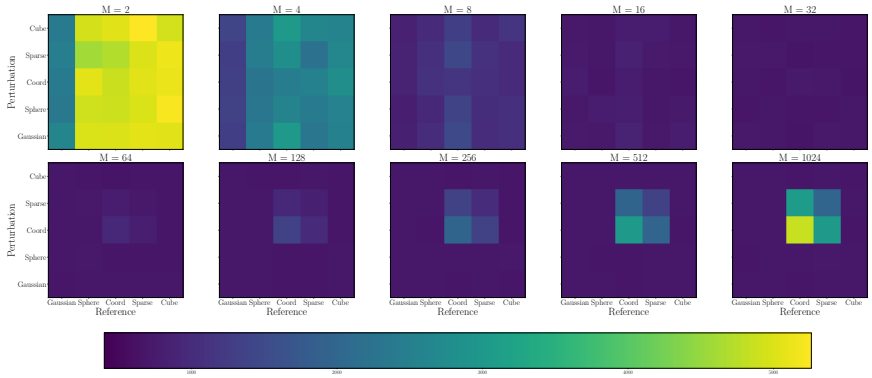
Figure 11: Results on the combinations of perturbation and reference distribution in Finite-action Linear Bandit under action dimension $d = 10$. A deeper color signifies lower accumulated regret and hence superior performance. Gaussian reference distribution significantly enhances performance.



(a) $d = 50 \quad |\mathcal{X}| = 100$

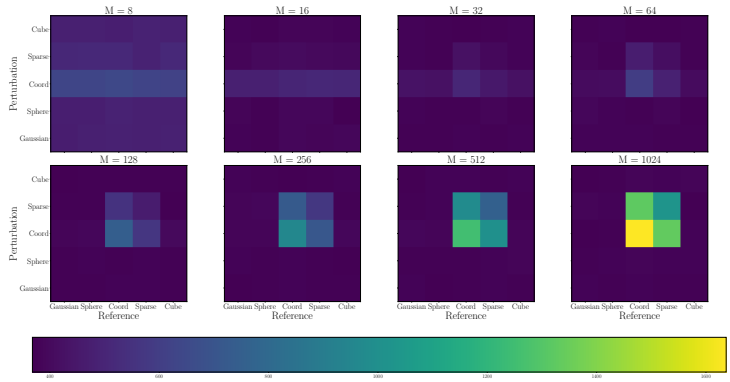


(b) $d = 50 \quad |\mathcal{X}| = 1000$

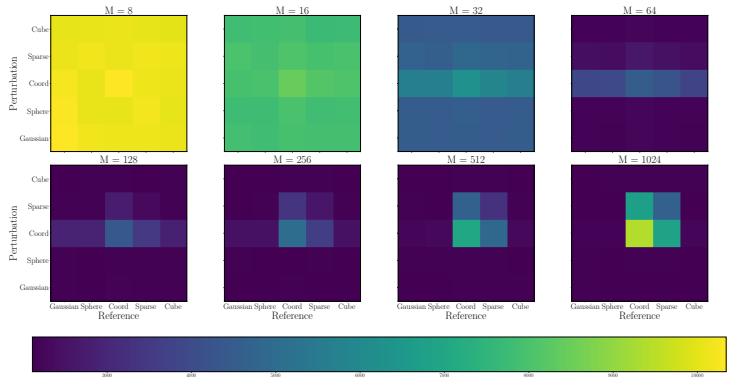


(c) $d = 50 \quad |\mathcal{X}| = 10000$

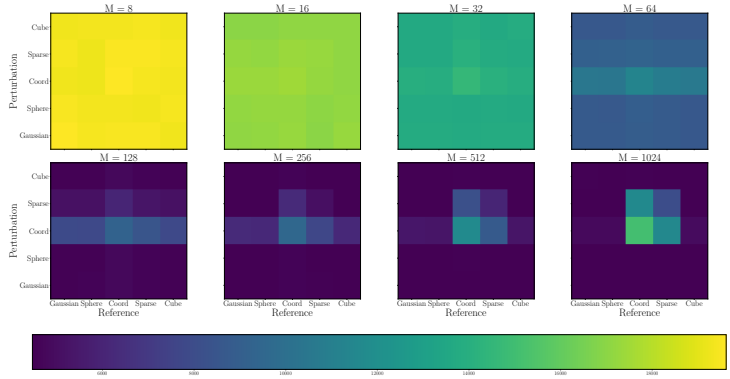
Figure 12: Results on the combinations of perturbation and reference distribution in Finite-action Linear Bandit under action dimension $d = 50$. A deeper color signifies lower accumulated regret and hence superior performance. Gaussian reference distribution significantly enhances performance.



(a) $d = 10$

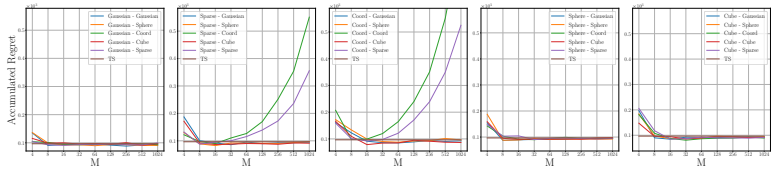


(b) $d = 50$

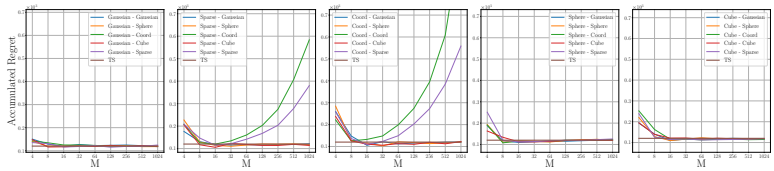


(c) $d = 100$

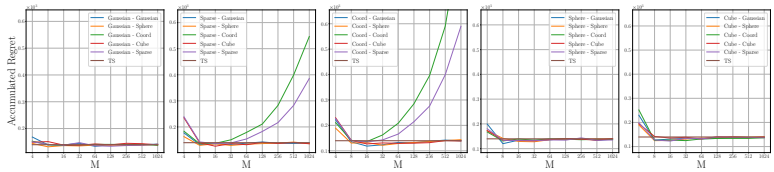
Figure 13: Results on the combinations of perturbation and reference distribution in Compact-action Linear Bandit. A deeper color signifies lower accumulated regret and hence superior performance. Gaussian reference distribution significantly enhances performance.



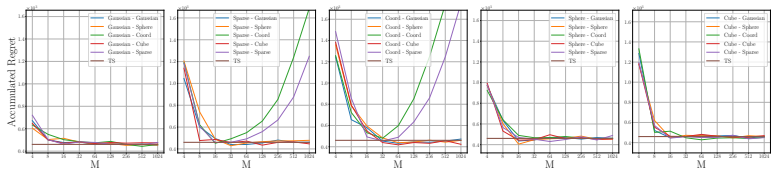
(a) $d = 10 \quad |\mathcal{X}| = 100$



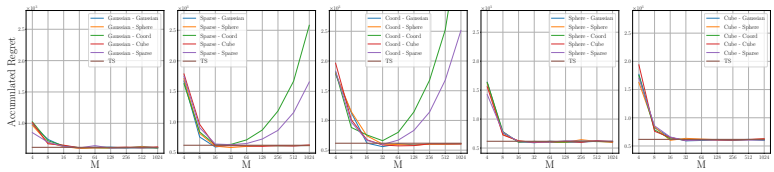
(b) $d = 10 \quad |\mathcal{X}| = 1000$



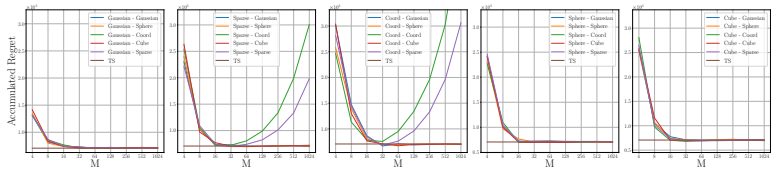
(c) $d = 10 \quad |\mathcal{X}| = 10000$



(d) $d = 50 \quad |\mathcal{X}| = 100$

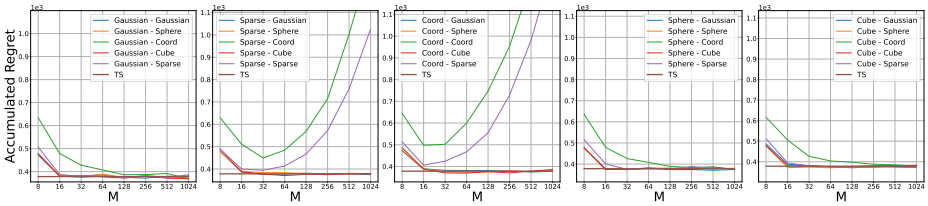


(e) $d = 50 \quad |\mathcal{X}| = 1000$

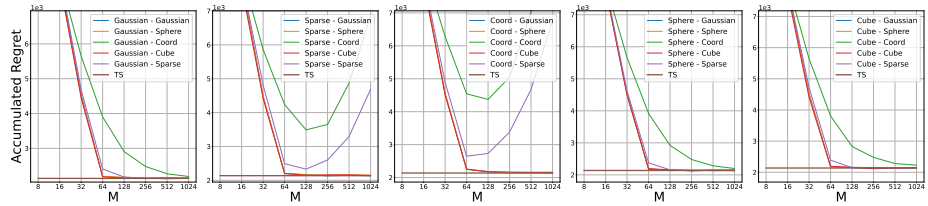


(f) $d = 50 \quad |\mathcal{X}| = 10000$

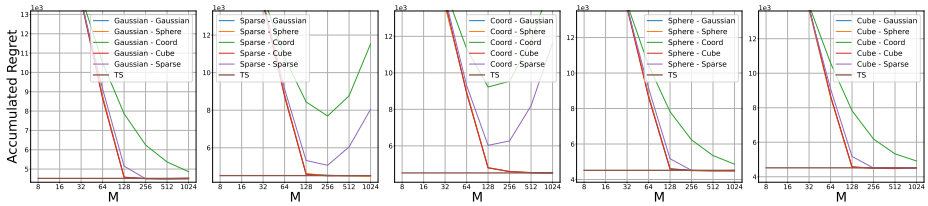
Figure 14: Results on regret under various #ensembles M in Finite-action Linear Bandit. The label $A - B$ indicates that Ensemble++ uses A as the reference distribution and B as the perturbation distribution. Ensemble++ with Gaussian or Sphere reference distribution could achieve comparable performance with that of Thompson sampling under same M for different action spaces $|\mathcal{X}|$.



(a) $d = 10$



(b) $d = 50$



(c) $d = 100$

Figure 15: Results on regret under various #ensembles M in Compact-action Linear Bandit. The label $A - B$ indicates that Ensemble++ uses A as the reference distribution and B as the perturbation distribution. Ensemble++ with Gaussian or Sphere reference distribution could achieve comparable performance with that of Thompson sampling under small M.

H.2 Additional Experiments on Nonlinear Bandit

We conduct more comprehensive comparison of Ensemble++ with several baselines that utilize approximate posterior sampling across a wide range of nonlinear bandits.

Environments Settings: We formulate several nonlinear contextual bandit environments, with rewards generated by nonlinear functions in each.

- **Quadratic Bandit**: Its reward generation mechanism is built on a quadratic function, expressed as $f_1(x) = 10^{-2}(x^\top \Theta \Theta^\top x)$. Here, $x \in \mathbb{R}^d$ stands for the action, while $\Theta \in \mathbb{R}^{d \times d}$ is a matrix filled with random variables originating from $\mathcal{N}(0, 1)$. This task is used as the testbed in Zhou et al. (2020).
- **Vector Quadratic Bandit**: Its reward generation mechanism is built on a different quadratic function, expressed as $f_2(x) = 10(x^\top \theta)^2$. Here, $a \in \mathbb{R}^d$ stands for the action, while $\theta \in \mathbb{R}^d$ is a vector filled with random variables generated from a uniform distribution over the unit ball. This task is utilized as the testbed in Zhou et al. (2020); Xu et al. (2022).
- **Neural Bandit**: This bandit employs a nonlinear neural network built on 2-layer MLPs with 50 units and ReLU activations, producing two output logits. We apply the softmax function with a temperature parameter $p = 0.1$ to the two output logits to obtain probabilities. Subsequently, we use binomial sampling based on the second probability to generate the reward. The temperature parameter p is used to control the signal-to-noise ratio. This task is used as the testbed in Osband et al. (2022; 2023a).
- **UCI Dataset**: Following prior works (Riquelme et al., 2018; Kveton et al., 2020b), we conduct contextual bandits with N-class classification using the UCI datasets (Asuncion et al., 2007) Mushroom and Shuttle. Specifically, given a data feature $x \in \mathbb{R}^d$ in the dataset, we construct context vectors for N arms, such as $x^{(1)} = (x, 0, \cdots, 0), \cdots, x^{(N)} = (0, 0, \cdots, x) \in \mathbb{R}^{Nd}$. Only the arm $x^{(j)}$ where j matches the correct class of this data x has a reward of 1, while all other arms have a reward of 0.
- **Online Hate Speech Detection**: The motivation, problem formulation and environment setups of the automated content moderation task are detailed in Section 4.

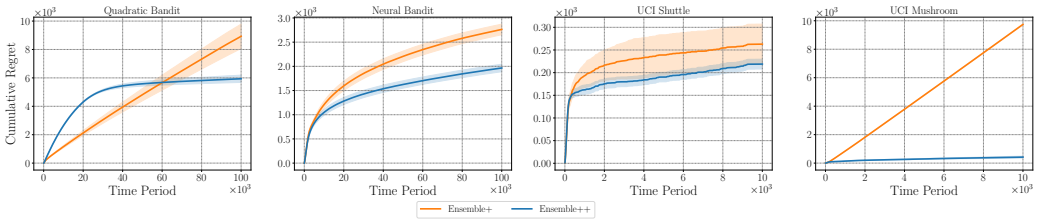
In all tasks except the **Neural Bandit**, the original reward r is disrupted by additive Gaussian noise ϵ drawn from $\mathcal{N}(0, 0.1)$. In the **Neural Bandit**, we use the temperature parameter p to introduce noise into the reward. For the first three tasks, we set the action dimension d to 100 and generate a total of 1000 candidated actions, randomly sampling 50 actions in each round. Each experiment is repeated with 10 distinct random seeds to ensure robust results.

Comparison Results with Baselines: We set the Sphere reference distribution, Coordinate update distribution, and Sphere perturbation distribution for Ensemble++ to compare with baselines. When comparing with Ensemble+ (Osband et al., 2018) and EpiNet (Osband et al., 2023a), we use the same hyperparameters, such as prior scale, learning rate, and batch size. Additionally, we employ the same network backbone for feature extraction to ensure fairness. As shown in Figure 16(a) and (b), **Ensemble++ achieves sublinear regret and consistently outperforms these baselines across all tasks, demonstrating superior data efficiency.**

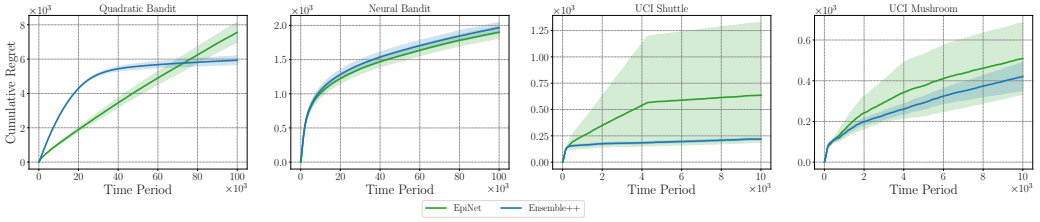
For comparison with LMCTS (Xu et al., 2022), we use its official implementation[4] to ensure credible results. As illustrated in Figure 16(c), Ensemble++ consistently outperforms LMCTS. Notably, LMCTS uses the entire buffer data to update the network per step, which incurs significant computational costs. In contrast, **Ensemble++ achieves better performance with bounded computational steps, requiring only a minibatch to update the network.** These findings highlight the effective exploration and computational efficiency of Ensemble++.

Additional Comparison on Trade-off between Regret and Computation: We have demonstrated that Ensemble++ can achieve sublinear regret with moderate computational cost in the Quadratic Bandit, as shown in **??**. Here, we further investigate the frontier relationship between regret and computation in the Neural Bandit. As shown in Figure 17, we observe similar findings: Ensemble++ achieves minimal cumulative regret with the lowest computational cost. These results substantiate the scalability and efficiency of Ensemble++ when combined with neural networks.

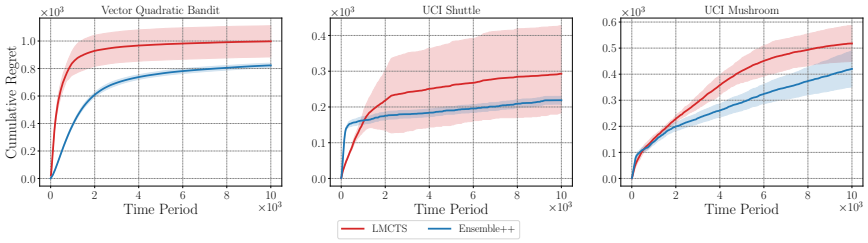
[4] https://github.com/devzhk/LMCTS



(a) Comparison results with Ensemble+ (Osband et al., 2018)

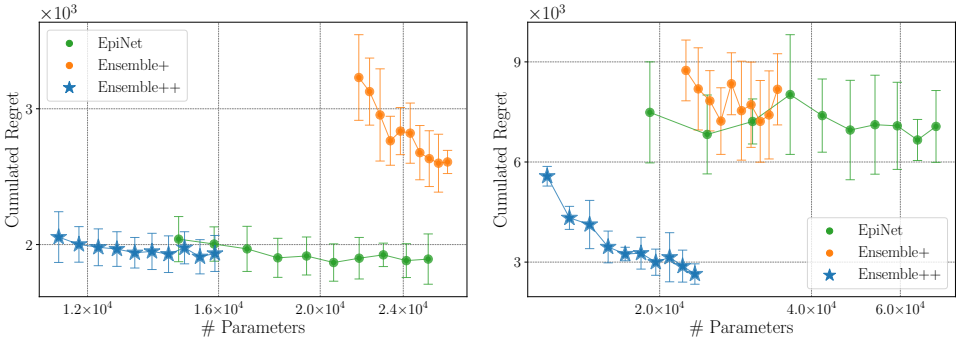


(b) Comparison results with EpiNet (Osband et al., 2023a)



(c) Comparison results with LMCTS (Xu et al., 2022)

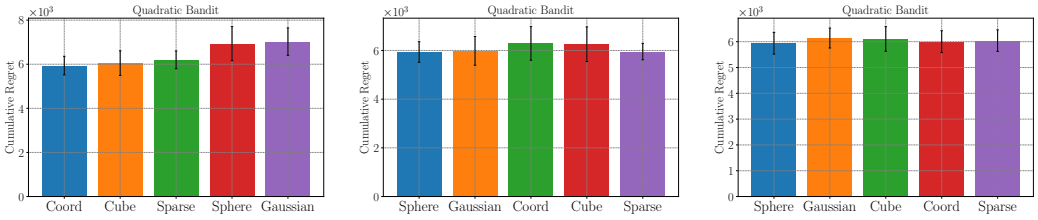
Figure 16: Results on different bandits with various baselines. Ensemble++ could achieve better performance compared to other methods.



(a) (b)

Figure 17: The regret-computation trade-off in (a) Neural Bandit and (b) Quadratic Bandit. Ensemble++ beats the SOTA baselines, e.g. Ensemble+ and EpiNet.

Ablation Study on Quadratic Bandit: To further evaluate the impact of different design of distributions, we perform an ablation study on the Quadratic Bandit. When fixing the Sphere reference distribution, we find that discrete update distributions such as Coordinate, Cube, and Sparse achieve similar better performance, as shown in Figure 18(a). Conversely, when fixing the Coordinate update distribution, continuous reference distributions like Sphere and Gaussian also yield comparable better performance, as depicted in Figure 18(b). Regarding the perturbation distribution, our findings indicate that it does not significantly influence performance when the neural network is involved in Ensemble++. This is evidenced in Figure 18(c), where all different perturbation distributions achieve similar performance.



(a) Results on update distribution. (b) Results on reference distribution. (c) Results on perturbation distribution.

Figure 18: Ablation studies about different distributions on the Quadratic Bandit.