Contextual Slate GLM Bandits with Limited Adaptivity

Tanmay Goyal Sukruta Prakash Midigeshi Gaurav Sinha

Microsoft Research India

Slate Bandits

- At each $t \in [T]$, the learner is presented with N item-sets $\{\mathcal{X}_t^1, \dots, \mathcal{X}_t^N\} \subset \mathbb{R}^d$.
- The learner chooses items $\boldsymbol{x}_t^i \in \mathcal{X}_t^i$ and plays the N-tuple $\boldsymbol{x}_t = (\boldsymbol{x}_t^1, \dots, \boldsymbol{x}_t^N)$.
- She receives reward $r_t(\boldsymbol{x}_t)$ sampled from a GLM, i.e, $r_t(\boldsymbol{x}_t) \mid \boldsymbol{x}_t$ is distributed as

$$\mathbb{P}_{\boldsymbol{\theta}^\star}(r_t(\boldsymbol{x}_t) \mid \boldsymbol{x}_t) = \exp\left(r_t(\boldsymbol{x}_t) \cdot (\boldsymbol{x}_t^\top \boldsymbol{\theta}^\star) - b(\boldsymbol{x}_t^\top \boldsymbol{\theta}^\star) + c(r_t(\boldsymbol{x}_t))\right).$$

The link function μ is defined as $\mu(\boldsymbol{x}_t^\top \boldsymbol{\theta}^\star) := \mathbb{E}[r_t(\boldsymbol{x}_t) \mid \boldsymbol{x}_t] = b'(\boldsymbol{x}_t^\top \boldsymbol{\theta}^\star)$.

- Here, $\boldsymbol{x}_t$ is called a *slate* and the item $\boldsymbol{x}_t^i$ is said to be placed in *slot* i.
- The learner aims to choose *slates* in a way to minimize her (expected) regret,

$$R(T) = \mathbb{E}\left[\sum_{t=1}^{T} \max_{\boldsymbol{x} \in \mathcal{X}_t^1 \times \dots \times \mathcal{X}_t^N} r_t(\boldsymbol{x}) - r_t(\boldsymbol{x}_t)\right].$$

- GLM Bandits require dealing with an (exponential) *non-linearity* parameter κ;

$$\kappa = \max_{\boldsymbol{x} \in \{\mathcal{X}_t^1 \times \dots \times \mathcal{X}_t^N\}_{t \in [T]}} \max_{\boldsymbol{\theta} \in \Theta} \frac{1}{\dot{\mu}(\boldsymbol{x}^\top \boldsymbol{\theta})}.$$

Limited Adaptivity

The learner is constrained to make only M policy updates.

Batched Setting

- Learner has to decide the update schedule *apriori*.
- The item-sets are assumed to be sampled from an unknown $\mathcal{D}$.
- $\Theta(\log \log T)$ updates are required and suffice for optimal regret [1,4].

Rarely-Switching Setting

- The learner can choose when to update the policy *on-the-fly*.
- The item-sets are assumed to be adversarially generated.
- $\tilde{\Omega}(d \log T)$ updates are required for optimal regret [2].

Applying SOTA algorithms [4] to our setting results in inefficient algorithms.

Goals

- Achieve *κ-free optimal regret* under limited adaptivity constraints.
- *Achieve efficiency*, i.e, avoid iteration over exponentially large set of slates.
- Handle single (*bandit*) feedback.

Diversity Assumption (A1)

The items $\boldsymbol{x}_t^i$ chosen by our algorithm for all $t \in [T], i \in [N]$ are *diverse* enough, i.e,

$$\exists \rho > 0 \text{ s.t. } \mathbb{E}[\boldsymbol{x}_t^i \boldsymbol{x}_t^{i\top} \mid \mathcal{F}_{t-1}] \succeq \rho \boldsymbol{I} \quad \text{and} \quad \mathbb{E}[\boldsymbol{x}_t^i \mid \mathcal{F}_{t-1}] = \boldsymbol{0}.$$

This is strictly weaker than the one used in [3], who assume $\mathbb{E}[\boldsymbol{x}_t^i \boldsymbol{x}_t^{i\top} \mid \mathcal{F}_{t-1}] \succeq \rho\kappa \boldsymbol{I}$.

Implication of (A1)

- We show that the eigenvalues of the design matrices grow linearly with time:

$$\forall i \in [N], \text{ let } \boldsymbol{H}_t^i = \sum \dot{\mu}(\boldsymbol{x}_t^\top \boldsymbol{\theta}^\star) \boldsymbol{x}_t^i \boldsymbol{x}_t^{i\top}, \text{ then, } \mathbb{P}[\lambda_{\min}(\boldsymbol{H}_t^i) \geqslant c\rho t] \geqslant 1 - \delta.$$

- This, in turn, allows us to show to show a *multiplicative equivalence*, i.e

$$\boldsymbol{H} \succeq c \cdot \mathrm{diag}(\boldsymbol{H}^1, \dots, \boldsymbol{H}^N) \implies \|\boldsymbol{x}\|_{\boldsymbol{H}^{-1}} \leqslant \sqrt{c} \|\boldsymbol{x}^i\|_{\boldsymbol{H}^{i^{-1}}}, \text{ where } \boldsymbol{H} = \sum \dot{\mu}(\boldsymbol{x}^\top \boldsymbol{\theta}) \boldsymbol{x}\boldsymbol{x}^\top.$$

Optimal Designs

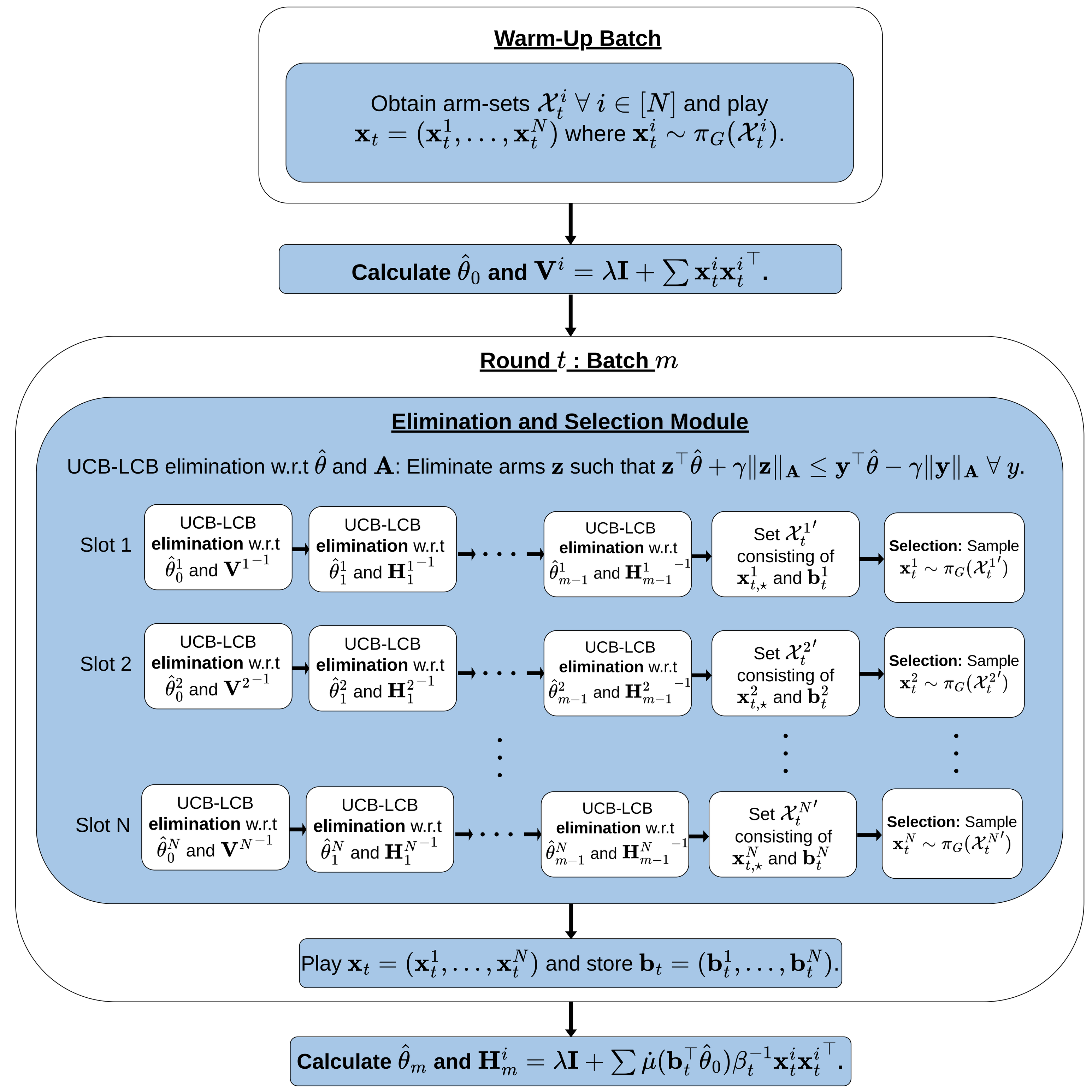
- **G-Optimal Designs**: These are the solution to the optimization problem:

$$\pi_G(\mathcal{X}) = \arg\min_{\pi \in \Delta(\mathcal{X})} \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|^2_{\boldsymbol{V}_\pi^{-1}}, \quad \text{where } \boldsymbol{V}_\pi = \sum_{\boldsymbol{x} \in \mathcal{X}} \pi(\boldsymbol{x}) \boldsymbol{x}\boldsymbol{x}^\top.$$

While finding π_G is NP-hard, approximate solutions can be found efficiently.

- **Distributional Optimal Designs**: Introduced by [2], these give optimal design guarantees for stochastically sampled arm-sets.

B-SlateGLinCB: A Batched Algorithm



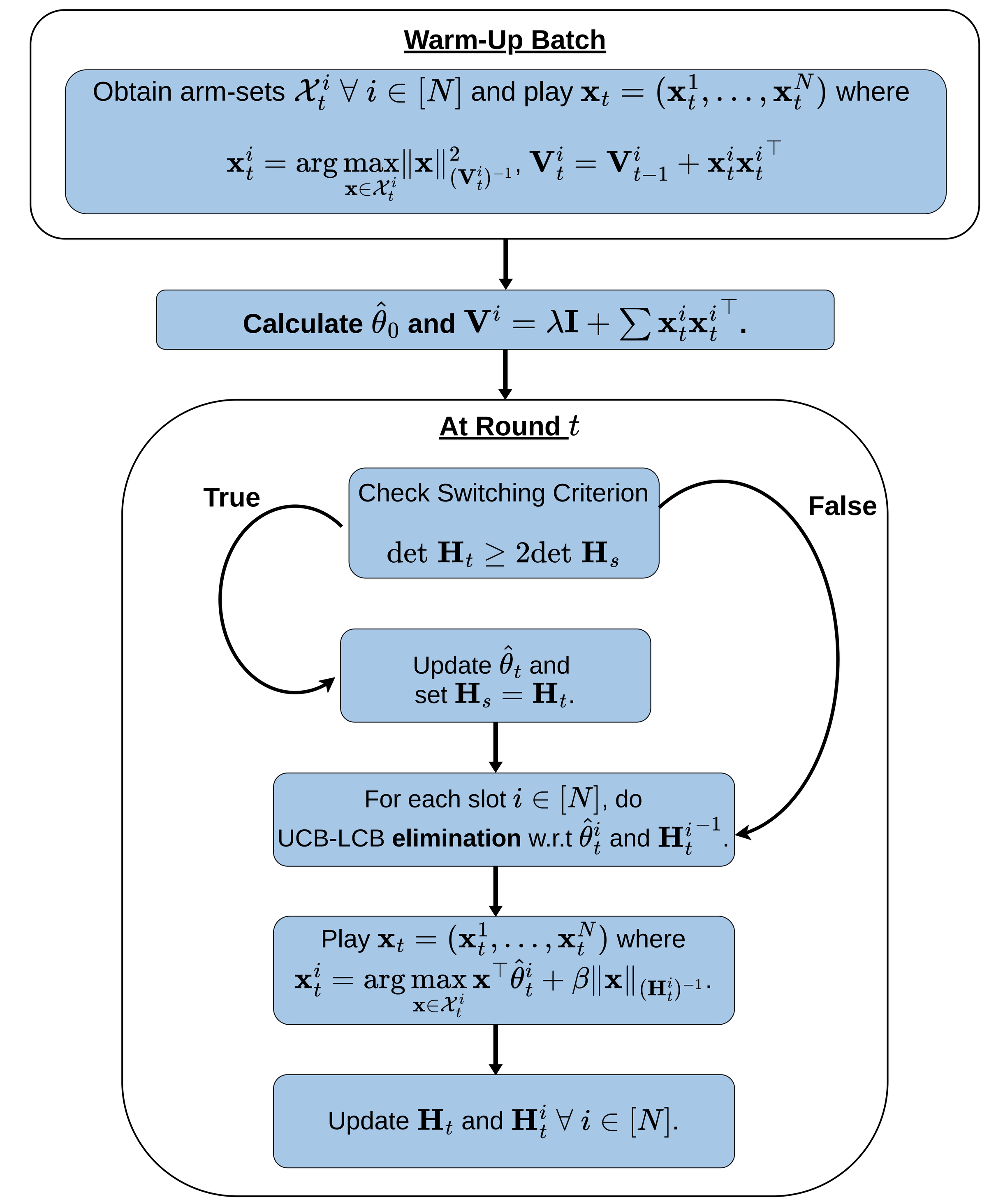
Key Ideas and Highlights

- Idea: (A1) allows to *reduce* the slate-level problem to a slot-level problem.
- Extending [4]: When $N > 1$, we get the terms $\dot{\mu}(\boldsymbol{x}^{i\top}\theta^{\star i})/\dot{\mu}(\boldsymbol{x}^\top\theta^\star)$ (can be $\propto \kappa$).
- Scaling Slate $\boldsymbol{b}_t$: Define $\boldsymbol{b}_t = (\boldsymbol{b}_t^1, \dots, \boldsymbol{b}_t^N)$ as

$$\boldsymbol{b}_t^i = \arg\max_{\boldsymbol{x} \in \mathcal{X}_t^i} \|\boldsymbol{x}\|_{(\boldsymbol{V}^i)^{-1}}.$$

- The Fix: Our scaling slate leads to the terms $\dot{\mu}(\boldsymbol{b}^\top\theta^\star)/\dot{\mu}(\boldsymbol{x}^\top\theta^\star)$. We show that these terms grow as $\mathcal{O}(1)$. Thus, we avoid a potential κ-dependency.

RS-SlateGLinCB: A Rarely-Switching Algorithm



Key Ideas and Highlights

- Idea: Using (A1), if a "switch" happens at round t,

$$\det \boldsymbol{H}_t \geqslant 2 \det \boldsymbol{H}_s \implies \|\boldsymbol{x}\|_{\boldsymbol{H}_s^{-1}} \leqslant \sqrt{2}\|\boldsymbol{x}\|_{\boldsymbol{H}_t^{-1}} \leqslant \sqrt{2c}\|\boldsymbol{x}^i\|_{\boldsymbol{H}_t^{i^{-1}}}.$$

- Removing *Switching Criterion I* from [4]: A consequence of using (A1) to show that eigenvalues of the design matrix $\boldsymbol{V}^i$ grow near-linearly.
- Improvement in #Switches over [4]: Our algorithms makes $\tilde{\mathcal{O}}(Nd\log T)$ updates, improving over [4], which makes $\tilde{\mathcal{O}}(\kappa N^2 d^2 \log^2 T)$ updates.
- Matching the lower bound: We match the bound in [2] upto polylog factors.

Regret Bounds

B-SlateGLinCB

$$R(T) = \tilde{\mathcal{O}}\left(N d^{3/2} \sqrt{T \cdot \mathbb{E}_{\mathcal{X}}\, \dot{\mu}(\boldsymbol{x}_\star^\top \boldsymbol{\theta}^\star)}\right)$$

B-SlateGLinCB with Distributional Optimal Designs

$$R(T) = \tilde{\mathcal{O}}\left(Nd\sqrt{T} \cdot \sqrt{\min\{d\, \mathbb{E}_{\mathcal{X}}\, \dot{\mu}(\boldsymbol{x}_\star^\top \boldsymbol{\theta}^\star), N \max_{\mathcal{X}} \dot{\mu}(\boldsymbol{x}_\star^\top \boldsymbol{\theta}^\star)\}}\right)$$

RS-SlateGLinCB

$$R(T) = \tilde{\mathcal{O}}\left(Nd\sqrt{\sum_{t=1}^{T} \dot{\mu}(\boldsymbol{x}_{t,\star}^\top \boldsymbol{\theta}^\star)}\right)$$

Experiments and References

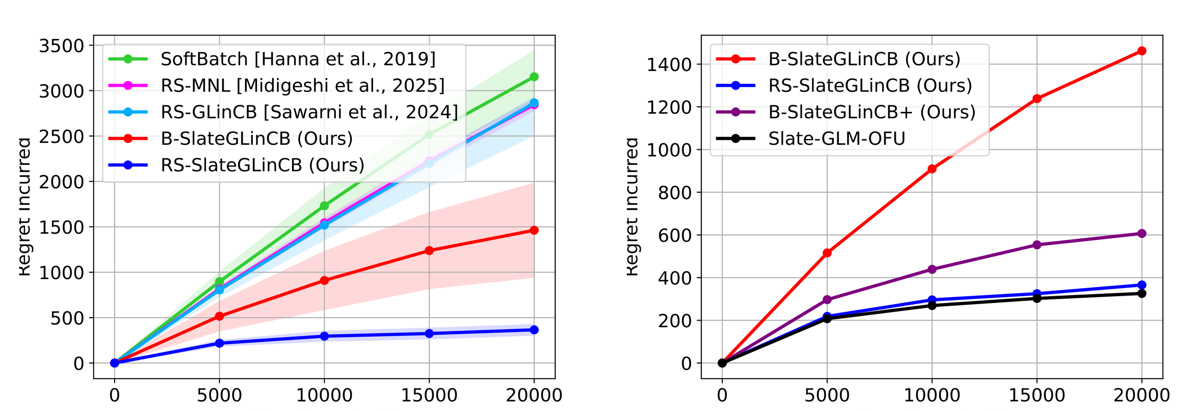


Figure 1: $\|\boldsymbol{\theta}^\star\| \leqslant 2,\ N = 5,\ d = 5$

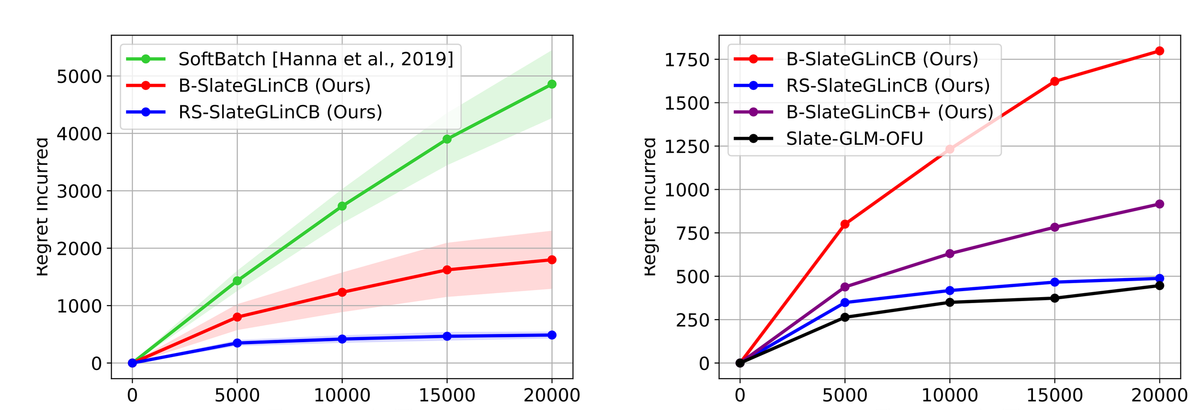


Figure 2: $\|\boldsymbol{\theta}^\star\| \leqslant 5,\ N = 10,\ d = 5$

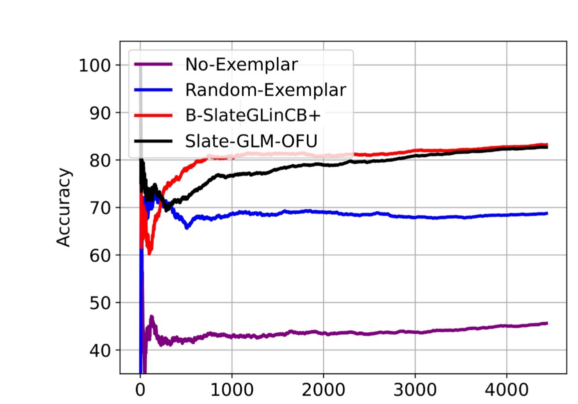


Figure 3: Prompt Tuning

[1] Gao et al. Batched Multi-armed Bandits Problem. NeurIPS 2019.
[2] Ruan et al. Linear Bandits with Limited Adaptivity and Learning Distributional Optimal Designs. STOC 2021.
[3] Goyal and Sinha. Efficient Algorithms for Logistic Contextual Slate Bandits with Bandit Feedback. UAI 2025
[4] Sawarni et al. Generalized Linear Bandits with Limited Adaptivity. NeurIPS 2024.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea.

arXiv:2606.31449