Scalable Policy Maximization Under Network Interference

Aidan Gleich
Duke University

Eric Laber
Duke University

Alexander Volfovsky
Duke University

Abstract

Many interventions, such as vaccines in clinical trials or coupons in online marketplaces, must be assigned sequentially without full knowledge of their effects. Multi-armed bandit algorithms have proven successful in such settings. However, standard independence assumptions fail when the treatment status of one individual impacts the outcomes of others, a phenomenon known as interference. We study optimal-policy learning under interference on large networks. Existing approaches to this problem require repeated observations of the same fixed network and struggle to scale in sample size beyond as few as fifteen connected units — both limit applications. We show that common assumptions on the structure of interference enable a parsimonious linear parameterization of the reward function. We develop a scalable Thompson sampling algorithm that maximizes cumulative rewards on an n-node network while allowing for both nodes and edges to be sampled at each time period. We prove upper and lower bounds on Bayesian regret that imply near-optimality. Simulation experiments show that our algorithm learns quickly and outperforms existing methods. The results close a key scalability gap between causal inference methods for interference and practical bandit algorithms, enabling policy optimization in large-scale networked systems.

1 INTRODUCTION

Sequential learning and decision making arises in contexts as varied as dynamic pricing in microeconomics (Rothschild, 1974), experimentation in online platforms (Wager and Xu, 2021), and adaptive treatment allocations in clinical research (Murphy, 2003). Many such problems revolve around the decision of which individuals to treat. For example, an online marketplace may want to assign coupons to customers who otherwise would not purchase an item, or a ride-sharing company may want to assign bonuses to drivers who would otherwise leave the platform. Both examples involve maximizing the impact of treatment, but doing so requires first learning its effect. Multi-armed bandit (MAB) algorithms have emerged as an effective method to solve both the learning and maximization problems simultaneously.

Classic bandit formulations assume that the outcome of an individual is not impacted by the treatment status of others. This assumption is violated in many important contexts, such as epidemiology (Benjamin-Chung et al., 2017), marketplaces (Munro et al., 2025), and social networks (Ogburn et al., 2024). We study **interference** or **spillover effects**, where treatments impact not only the treated units but also their peers. We focus on interference that spreads along a network, where the nodes of the network define individuals or arms and the edges specify connections. For example, online experimentation often occurs on social networks or internet marketplaces where interference impacts outcomes and the underlying network changes over time.

Despite their relevance to such problems, little work on MABs exists in this setting; existing algorithms struggle to scale beyond networks of size $n \approx 15$ (Agarwal et al., 2024; Jamshidi et al., 2025). Moreover, these methods require that the network remains fixed across time *and* that observations of rewards across time are independent. The simultaneous assumptions of a fixed network with persistent individuals and temporal independence can lead to inconsistencies. For example, the outcomes of participants in a vaccine trial depend on their treatment statuses in earlier periods. In this work, we relax the assumption that the network remains fixed.

Existing algorithms fail to scale because they attempt

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

to capture arbitrary interference instead of leveraging common structural assumptions from the causal inference literature. For example, in the case of L treatment levels, the linear reformulation of Agarwal et al. (2024) means that a node of degree d is represented by $(L+1)^{d+1}$ unique parameters. A network with maximum degree $d_{\max}$ thus has up to $n(L+1)^{d_{\max}}$ unknown parameters that need to be estimated. Thus, with $n = 1000$, $d_{\max} = 10$, and $L = 2$, the reward function would have over a million unknowns. Estimating this many parameters, which grows linearly with n and exponentially with node degree $d_{\max}$, leads to computational complexity cubic in n and prohibitive data requirements.

The causal inference literature has studied interference on networks with particular emphasis on specifying assumptions that are both realistic and practical. However, this work is almost exclusively in the context of static experimentation with a focus on identification and inference as opposed to dynamic decision making. Motivated by applications where networks are large and dynamic, we address the gap between the MAB and causal inference literature by devising a scalable MAB algorithm. Our key insight is that by adopting common assumptions (Manski, 2013) about the additivity and symmetry of interference patterns, rewards form a system of n linear equations, allowing for an efficient Thompson sampling algorithm capable of handling networks that are orders of magnitude larger than competing methods.

Our contributions are as follows: we show that under common interference assumptions, the vector of node-level rewards becomes linear in a parameter vector $\boldsymbol{\theta}$. Within a broad class of reward function formulations, $\boldsymbol{\theta}$ is low-dimensional, enabling a scalable Thompson sampling algorithm that maximizes cumulative rewards. Our framework allows for both nodes and edges to be sampled at each time period, resolving potential contradictions in previous works that require both temporal independence and a fixed population of nodes. We prove both upper and lower bounds on regret, showing our algorithm is near-optimal. Finally, to underscore the flexible nature of our framework, we provide empirical results under a variety of reward functions and network specifications.

2 BACKGROUND

Interference and Peer Influence Effects Researchers in causal inference and econometrics have dedicated considerable work to interference and peer influence effects. Many topics, such as identification (Manski, 1993), inference (Tchetgen and VanderWeele, 2012; Aronow and Samii, 2017; Toulis and Kao, 2013), and experimental design (Hudgens and Halloran, 2008; Eckles et al., 2017) have been covered in a variety of contexts, including networks (e.g., see Bramoullé et al. (2020) for a survey). However, little work combines an inference objective under interference with a maximization problem. Notable exceptions include Wager and Xu (2021) and Viviano (2024). Neither matches our setting, as Wager and Xu (2021) considers a specific form of interference due to marketplace effects and Viviano (2024) is restricted to offline contexts.

Bandits with Interference Interference is in part a dimensionality problem. Without restriction, in the case of a binary treatment and n individuals, it increases the number of arms from n to 2^n. While the literature on combinatorial bandits (e.g., Chen et al. (2013)) provides intuition for the difficulties of our problem related to the curse of dimensionality, assumptions typical of that literature (such as independent rewards) do not hold in our setting. Work on bandits under interference is relatively sparse. Spatial interference has been studied (Laber et al., 2018; Jia et al., 2024) as well as network interference (Agarwal et al., 2024). Work exists on optimal individual treatment regimes with interference (Su et al., 2019) as well as linear contextual bandits under interference (Xu et al., 2024). Our algorithm draws its inspiration from the linear bandit literature (Abbasi-Yadkori et al., 2011) and Thompson sampling (Russo and Van Roy, 2014). The work closest to ours is Agarwal et al. (2024). The authors design a MAB algorithm for a static network assuming only "sparse network interference," and formulate the reward function as a linear model. That work does so indirectly through discrete Fourier analysis, while we show that the rewards can be written directly as a linear function of unknown parameters. We show that our method can outperform Agarwal et al. (2024) even when our linear formulation is misspecified.

3 MODEL SETUP

At each time period t, the agent observes an adjacency matrix $\mathbf{A}_t$ for a set of n nodes with maximum degree $d_{\max}$. The agent must choose the treatment allocation vector $\mathbf{Z}_t \in \{0, 1, \ldots, L\}^n$ potentially subject to budget constraint B_t (e.g. an upper limit on the number of treated nodes) where L denotes the number of treatment levels. Each node i has a corresponding reward function $r_i(\mathbf{Z}_t; \mathbf{A}_t)$. The agent attempts to maximize the sum of the reward functions over time.

Without placing assumptions on interference, each node-level reward function r_i depends on the entire treatment vector $\mathbf{Z}_t$ as well as $\mathbf{A}_t$, which we illustrate in Figure 1. For a given network $\mathbf{A}_t$, each reward func-

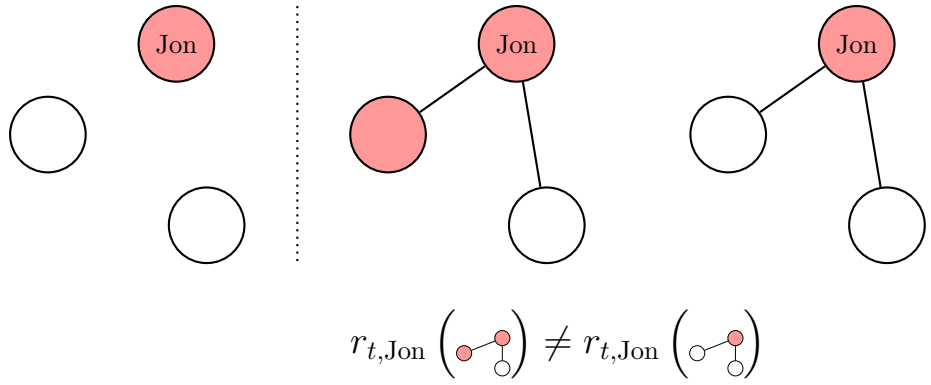

Figure 1: A visualization of network interference, where the reward function of "Jon" depends on the treatment status of his neighbors. On the left is the standard MAB case.

tion is thus a mapping $r_i(\cdot; \mathbf{A}_t) : \{0, 1, \ldots, L\}^n \to \mathbb{R}$. Learning each of these mappings simultaneously becomes computationally infeasible due to the input space growing exponentially in n. Therefore, we adopt a set of common assumptions on the structure of interference to allow for scalable optimization of the treatment policy.

3.1 Interference Assumptions

We adopt the neighborhood interference assumption (Sussman and Airoldi, 2017; Belloni et al., 2025) for the node-level reward functions. Let $\mathcal{N}_{t,i}$ denote the set of neighbors of node i at time t, i.e. for all $j \in \mathcal{N}_{t,i}$, $[\mathbf{A}_t]_{i,j} = 1$.

Assumption 1. *For all nodes i, the reward function r_i satisfies the neighborhood interference assumption (NIA) if for all treatment assignments $\mathbf{Z}_t$, $\mathbf{Z}'_t$ that agree on the set $\mathcal{N}_{t,i} \cup \{i\}$, $r_i(\mathbf{Z}_t; \mathbf{A}_t) = r_i(\mathbf{Z}'_t; \mathbf{A}_t)$.*

NIA requires that a node's reward function depends only on its own treatment and the treatment status of its neighbors, reducing the dimension of the input space. For example, with $n = 100$ and no assumptions on the interference pattern, the reward function of each node depends on the entire 100-length vector of treatment assignments, implying $(L+1)^n$ possible inputs. Under NIA, a node with degree d would have a reward function with $(L+1)^{(d+1)}$ inputs.

Under NIA, the vector of node-level reward functions can be represented as linear in a parameter vector $\boldsymbol{\theta}$ containing treatment and interference effect parameters (Sussman and Airoldi, 2017, eq. 4.1):

$$\mathbf{r}_t = \mathbf{H}(\mathbf{Z}_t; \mathbf{A}_t)\boldsymbol{\theta} + \boldsymbol{\epsilon}_t \tag{1}$$

where $\mathbf{H}(\mathbf{Z}_t; \mathbf{A}_t)$ is an $n \times p$ feature matrix that maps treatment assignments and network structure onto the effect parameters. We emphasize that linearity is not a modeling assumption but a reparameterization that directly follows from the neighborhood interference assumption (see Supplement Section 1 for details). However, without further assumptions, the dimension of $\boldsymbol{\theta}$ still inhibits scalable exploration and exploitation as each node has on the order of $(L+1)^{d+1}$ parameters. The following assumptions further restrict the form of interference.

Assumption 2. *The reward function $r_{t,i}$ satisfies additivity of main effects if*

$$r_i(Z_{t,i}, \mathbf{Z}_{\mathcal{N}_{t,i}}) = r_i(Z_{t,i}, \mathbf{0}) + r_i(0, \mathbf{Z}_{\mathcal{N}_{t,i}}).$$

Assumption 3. *The reward function r_i satisfies symmetrically received interference if, for all permutations τ, $r_i(Z_{t,i}, \mathbf{Z}_{\mathcal{N}_{t,i}}) = r_i(Z_{t,i}, \tau(\mathbf{Z}_{\mathcal{N}_{t,i}}))$.*

We have dropped the dependence of r_i on $\mathbf{A}_t$ for notational simplicity. Assumption 2 states that the direct effect does not interact with the indirect effects. Assumption 3 implies that a node's reward depends only on the number of treated neighbors, not which specific neighbors get treated. Together, Assumptions 1 to 3 are termed SANIA (Symmetric and Additive NIA).

3.2 Reward Function Specification

The SANIA assumptions allow for a large class of reward functions; the specific node-level form can be tailored to the application, including information about the underlying network structure and the nodes themselves.

A simple starting point is to assume that each node-level reward function has unique direct and indirect effect parameters that do not depend on context or network structure beyond the adjacency matrix. Allowing $d^1_{t,i}$ to be the number of treated neighbors of

node i at time t, this results in a reward function of the form

$$r_i(\mathbf{Z_t}; \mathbf{A_t}) = Z_{t,i} \cdot \mu_i + \sum_{k=1}^{d_i} \gamma_{i,k} \cdot \mathbf{1}_{\{d^1_{t,i}=k\}} + \epsilon_{t,i} \quad (2)$$

where we leave out an intercept parameter for simplicity and assume treatment $Z_{t,i}$ is binary. Here, d_i denotes the degree of node i and d_i^1 the number of treated neighbors of node i. The parameters $\gamma_{i,k}$ determine the spillover effect of having k treated neighbors. This is the form taken by Agarwal et al. (2024) with further restrictions due to SANIA.

We note two issues with this approach. First, the number of parameters scales linearly with the size of the network ($O(n \cdot d_{\text{avg}})$). While this improves upon the exponential scaling under NIA, it can be restrictive for large networks. Second, the use of fixed node-specific parameters implicitly assumes a static population observed over time. It thus becomes difficult to justify temporal independence in errors.

We provide alternative parameterizations that allow for information sharing across nodes and do not rely on fixed node identities. Our method does not rely on a specific parameterization but instead defines a flexible framework that researchers can use to construct models tailored to their application.

3.2.1 Shared Parameters

The simplest parameterization within this framework assumes that all nodes share a single set of parameters: $\mu_i = \mu_j$ and $\gamma_{i,k} = \gamma_{j,k}$ for all i, j, k. If all interference parameters γ_k are 0, this reduces to the standard individualistic treatment response assumption. With nonzero interference, it becomes the constant treatment response in the depth of interference model (Manski, 2013).

This approach is highly scalable but has the potential to be misspecified. Despite this sensitivity, it is a common approach in the causal inference literature (Toulis and Kao, 2013; Eckles et al., 2017) and can be aided by including covariates in the model.

3.2.2 Grouped Parameters

A natural extension of the previous parameterization assumes that groups of units (either observed or latent) share parameters. This model is based on the observations in sociology and network science that similarly behaving nodes tend to share connections (McPherson et al., 2001). All nodes in group g share a parameter vector $\boldsymbol{\theta}_g = \{\mu_g, \gamma_{g,1}, \ldots, \gamma_{g,m}\}$. These combine to form the full parameter vector: $\{\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_G^\top\}^\top$.

3.2.3 Latent Features and Other Generalizations

A further generalization embeds the networks in a Euclidean space. The distance between nodes in this space determines the probability of forming links. Using the latent space model of Hoff et al. (2002), positions in the unobserved space can be estimated for each node and included in their reward function. For example, assuming each node has a corresponding two-dimensional location vector $\boldsymbol{\zeta}$, the reward functions could take the form

$$r_i(\mathbf{Z}_t; \mathbf{A}_t, \boldsymbol{\zeta}_{t,i}) = \boldsymbol{\zeta}_{t,i}^\top \boldsymbol{\mu} + \sum_{k=1}^{d_i} (\boldsymbol{\zeta}_{t,i}^\top \boldsymbol{\alpha}_k) \cdot \mathbf{1}_{\{d^1_{t,i}=k\}} + \epsilon_{t,i}.$$

3.3 Design Matrix

The structure of the design matrix $\mathbf{H}(\mathbf{Z}_t; \mathbf{A}_t)$ is determined by the chosen reward parameterization. Its rows consist of indicator variables derived from the treatment vector $\mathbf{Z}_t$, features summarizing the interactions between $\mathbf{Z}_t$ and $\mathbf{A}_t$ (e.g. the counts of treated neighbors), and additional features such as group labels.

For example, under the shared parameters model, the portion of $\mathbf{H}$ corresponding to the direct effect μ would be a single column containing the indicators $Z_{t,i}$.

Under the grouped parameters model, there would be a set of columns for the direct effects; the column for group g would have an entry of 1 for node i if it is in group g and is treated, and 0 otherwise.

While each row is simple to compute, the mapping itself can be complex and thus maximizing the rewards with respect to $\mathbf{Z}_t$ conditional on a set of parameters can be difficult. We discuss this issue further in later sections.

3.4 Regret

The agent seeks to maximize the cumulative node-level rewards with respect to the treatment vector. To measure the agent's performance, we use the Bayesian regret—the expected gap in cumulative rewards under the optimal policy and the agent's actions with respect to the prior distribution over $\boldsymbol{\theta}$. The optimal treatment at time t, denoted $\mathbf{Z}_t^*$, maximizes the sum of expected node-level rewards:

$$\mathbf{Z}_t^* = \arg\max_{\mathbf{Z}_t} \sum_{i=1}^{n} \mathbb{E}\left[r_i(\mathbf{Z}_t; \mathbf{A}_t)\right]$$

The regret is then the cumulative expected difference in rewards between the optimal policy and the agent's

policy:

$$\text{Reg}_T = \sum_{t=1}^{T} \sum_{i=1}^{n} \mathbb{E}\left[r_i(\mathbf{Z}_t^*; \mathbf{A}_t) - r_i(\mathbf{Z}_t; \mathbf{A}_t)\right].$$

In the following section, we define an algorithm that achieves low regret with high probability and provide upper and lower bounds on regret.

4 THOMPSON SAMPLING UNDER NETWORK INTERFERENCE

We devise a scalable Thompson sampling algorithm that can accommodate any reward function satisfying the neighborhood interference assumption, such as those discussed in the previous section. The parameterization of the reward functions will determine the dimension of $\boldsymbol{\theta}$, which will in turn govern the algorithm's bounds and computational complexity. We discuss in detail the impact of $\boldsymbol{\theta}$ on the theoretical and empirical results of our algorithm, thus providing a flexible and theoretically grounded framework for maximizing treatment policies under network interference.

4.1 Thompson Sampling Algorithm

We begin with the standard assumption on the noise $\boldsymbol{\epsilon}_t$. This assumption facilitates the theoretical development in our paper and naturally maps onto the linear specification in Sussman and Airoldi (2017), which includes an individualized baseline effect (see additional details in the Supplement):

Assumption 4. *The terms $\epsilon_{t,i}$ are independent 1-sub-Gaussian random variables for all t, i.*

Thompson sampling requires a prior distribution over the parameter vector $\boldsymbol{\theta}$ of dimension D. Motivated by normal-normal conjugacy in Bayesian linear regression, we specify that $\boldsymbol{\theta} \sim \text{N}\left(\mathbf{0}, \frac{1}{\lambda}\mathbf{I}_D\right)$. Having received an observation $\mathbf{r}_t$, the posterior is updated using the standard conjugate posterior formulas (Gelman et al., 2013). The hyperparameter λ is effectively equivalent to the penalty term in ridge regression. Because its effect diminishes in the number of observations, it can be ignored in theoretical analysis, but it can have considerable impact on regret in early time periods. We discuss this further in Section 3 of the Supplement.

Our algorithm thus maintains a posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{r}_1, \ldots, \mathbf{r}_t)$ over the unknown parameters $\boldsymbol{\theta}$. At each time period, the agent draws $\boldsymbol{\theta}^{(t)}$ from the posterior and chooses action $\mathbf{Z}_t$ that maximizes the total reward conditional on $\boldsymbol{\theta}^{(t)}$.

Although our algorithm resembles Thompson sampling for classical linear bandits in its form, its application to our setting is not straightforward. In classical linear bandits, a single reward observation is modeled as $r_t = \langle X_t, \boldsymbol{\theta} \rangle + \epsilon_t$. In contrast, our setting involves n interdependent reward observations per round, each influenced by network interference. The interdependence means that it is impossible to simply perform n independent linear bandit updates. Our reformulation enables the extension of scalable linear bandit algorithms to a regime where prior work has failed to scale beyond the smallest of networks ($n \approx 10$). The algorithm requires new regret analysis, which we provide in the following section.

Algorithm 1 Thompson Sampling under Interference

1: **Input:** Prior mean $\boldsymbol{\mu}_0$, prior covariance $\boldsymbol{\Sigma}_0$, regularization λ, noise variance σ^2
2: **for** $t = 1$ to T **do**
3: Observe network $\mathbf{A}_t$ and any features
4: Sample $\boldsymbol{\theta}^{(t)} \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$
5: Choose treatment vector $\mathbf{Z}_t$:

$$\mathbf{Z}_t = \arg\max_{\mathbf{Z}} \mathbf{1}_n^\top [\mathbf{H}(\mathbf{Z}; \mathbf{A}_t)\boldsymbol{\theta}^{(t)}]$$

6: Observe rewards $\mathbf{r}_t$
7: Compute $\mathbf{H}_t = \mathbf{H}(\mathbf{Z}_t; \mathbf{A}_t)$
8: Update posterior:

$$\boldsymbol{\Sigma}_t = \left(\mathbf{H}_t^\top \mathbf{H}_t/\sigma^2 + \boldsymbol{\Sigma}_{t-1}^{-1}\right)^{-1}$$
$$\boldsymbol{\mu}_t = \boldsymbol{\Sigma}_t \left(\mathbf{H}_t^\top \mathbf{r}_t/\sigma^2 + \boldsymbol{\Sigma}_{t-1}^{-1}\boldsymbol{\mu}_{t-1}\right)$$

9: **end for**

In practice, the agent may be constrained by a budget. For example, they may only be able to treat $B < n$ units per round. This is easily accommodated by our algorithm and does not impact the theoretical results, as it simply redefines the optimal policy as a maximum over the constrained action space.

Computational Complexity Allowing D to denote the dimension of $\boldsymbol{\theta}$, line 8 of Algorithm 1 has time complexity $O(nD^2 + D^3)$. For comparison, a regression on the most general NIA formulation has complexity $O(n^3 2^{2d_{\max}} + n^3 2^{3d_{\max}})$. This scales with n^3 (and exponentially in n if $d_{\max}$ increases with n), meaning that existing explore-then-commit algorithms (e.g. Agarwal et al. (2024)) fail to scale with n.

The maximization problem at Line 5 of Algorithm 1 is nontrivial and is the primary computational bottleneck during implementation. We used the Gurobi optimization software (Gurobi Optimization, LLC, 2024) in our experiments. This problem has not yet been discussed in the network bandits literature, as existing algorithms fail to scale beyond small networks where maximization is easy.

4.2 Regret Analysis

We first provide an upper bound on the Bayesian regret of Algorithm 1.

Theorem 1. *Assume there exist positive constants c_1, c_2 such that $\sup_{\boldsymbol{\theta}\in\Theta} \|\boldsymbol{\theta}\|_2 \leq c_1$ and $\sup_{\mathbf{H}\in\mathcal{H}^n} \|\mathbf{H}\|_2 \leq c_2$, and suppose Assumptions 1–4 hold. Algorithm 1 then satisfies*

$$\textit{BayesRegret}(n,T) = O\left(D\sqrt{nT}\log(nT)\right)$$

We prove Theorem 1 by adapting the results on frequentist regret of Abbasi-Yadkori et al. (2011) to our setting and then applying Proposition 5 of Russo and Van Roy (2014). Our assumptions on $\boldsymbol{\theta}$ and $\mathbf{H}$ match those of Russo and Van Roy (2014). Full proofs are provided in Section 2 of the Supplement.

We show that our approach is near-optimal by deriving a lower bound on regret.

Theorem 2. *Under the same assumptions as Theorem 1, for any policy π, there exists a $\theta \in \Theta$ such that $R_T(\mathcal{A},\theta) \geq \Omega\left(D\sqrt{nT}\right)$.*

The proof adapts arguments from Section 24.1 of Lattimore and Szepesvári (2020). We see that the upper bound provided in Theorem 1 matches this lower bound up to a logarithmic factor, thus proving that our algorithm is near-optimal.

Interestingly, our regret bounds resemble those for standard linear bandits but include a factor of n. The intuition for this difference lies in the feedback structure: for a network of size n, by time T we have accumulated information for nT nodes while the standard linear bandit algorithms will have only received T observations. However, the increased complexity of the action space partially offsets the increase in information at each round. The balance of these opposing forces mirrors the exploration-exploitation tradeoff in standard linear bandits, resulting in our bounds.

As discussed in Section 3.2, a potential pitfall is that D may scale with n. For example, if each node has a unique direct effect, then the dimension of $\boldsymbol{\theta}$ scales at least linearly in n. We focus our theoretical analysis on the case of a fixed D and discuss the impact of T and n on regret as this allows a clear comparison to other algorithms in the bandit literature. We provide results from a variety of choices of the reward function through simulation studies in the following section.

Comparison to previous approaches

- **Classic MAB Approach.** Standard MAB algorithms fail in this context for two reasons. First, they would attempt to learn each of the $(L+1)^n$ arms independently, which quickly becomes infeasible (e.g. TS has a regret bound of $O\left(\sqrt{2^n T\log 2^n}\right)$ (Agrawal and Goyal, 2017)). Second, they can only be applied to a static network, as the node reward functions are fixed over time.

- **Linear Bandits.** Collapsing Equation (1) by summing over the rows results in a standard linear bandit formulation. However, this turns n data points into 1, losing a significant amount of information about $\boldsymbol{\theta}$. By learning more from the same data, our algorithm performs strictly better.

- **Network Bandits.** Existing approaches for bandits under network interference assume only NIA. The algorithm in Agarwal et al. (2024) attempts to estimate the Fourier coefficients of the reward function for each node. The length of the resulting parameter vector is exponential in the size of the network, severely limiting scaling ability. The inability of existing algorithms to scale is a primary motivation for our work.

5 SIMULATIONS

We validate the flexibility and scalability of our method through simulation studies. We structure our experiments around three reward function parameterizations: a shared parameter model, a block-structured model, and a more complex latent features model.

5.1 Linear Spillovers

We begin with a shared-parameter model. Each node has reward function

$$r_i(\mathbf{Z}_t;\mathbf{A}_t) = \mu\cdot Z_{t,i} + \sum_{k=1}^{d_i}\gamma_k\cdot\mathbf{1}_{\{d^1_{t,i}=k\}} + \epsilon_{t,i}.$$

We set $\boldsymbol{\mu}_0 = 0, \Sigma_0 = \mathbf{I}$, and use budget $B = \frac{n}{5}$ (i.e. $\|\mathbf{Z}_t\|_1 \leq B$). For each iteration, we draw $\mu \sim \mathrm{N}(1, 0.2)$ and $\gamma_k \sim \mathrm{N}(k, 0.5)$. At each time period, we simulate $\mathbf{A}_t$ from a stochastic block model (SBM) of sizes $n \in \{100, 500, 1000\}$ with $K = \frac{n}{10}$ groups having uniform group membership probabilities, the probability $p_{\text{within}} = 0.25$ of sharing a link with nodes in the same group and probability $p_{\text{between}} = \frac{1}{n}$ of a link between nodes in different groups. Errors are distributed $\epsilon_{i,t} \sim \mathrm{N}(0,1)$.

For each n, we run 25 iterations with $T = 100$. In Figure 2 we evaluate our algorithm using the mean regret of the iterations and include 95% confidence bands.

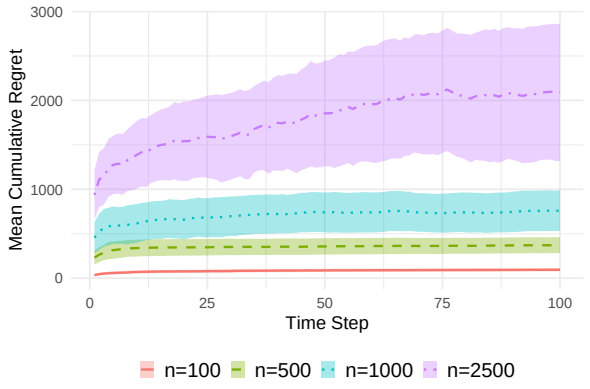

Figure 2: Cumulative regret plot under the shared parameters model.

5.2 Grouped Effects

We now consider the grouped model of Section 3.2.2. For each group, the reward function follows Equation (2), with μ_g representing the direct effect for group g and γ_g representing the vector of potential spillover effects for units in group g:

$$r_i(\mathbf{Z_t}; \mathbf{A_t}, G_{t,i} = g) = Z_{t,i} \cdot \mu_g + \sum_{k=1}^{d_i} \gamma_{g,k} \cdot \mathbf{1}_{\{d^1_{t,i}=k\}} + \epsilon_{t,i}.$$

We use the same data generating process for $\mathbf{A_t}$ as the previous section. The parameter groupings at each time t are defined by the K groups of the SBM which we for now assume the agent observes.

For each iteration, we draw parameters $\mu_g \sim \mathrm{N}(1, 0.2)$, $\gamma_{g,k} \sim \mathrm{N}(k, 1)$. The priors, budget, and number of iterations are the same as the previous section. We plot the cumulative regret for multiple network sizes in Figure 3.

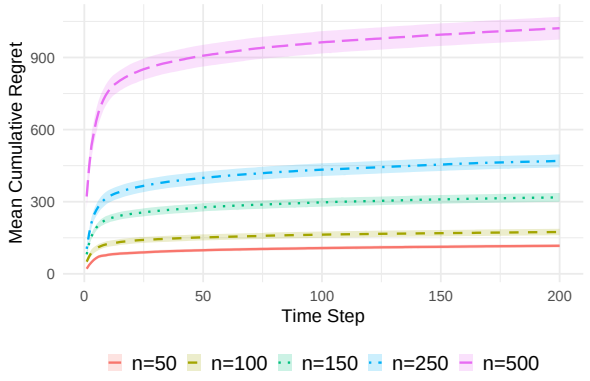

Figure 3: Cumulative regret for a variety of network sizes under grouped SANIA reward functions.

5.2.1 Estimating Group Labels

We now consider the setting where the agent does not observe the group labels of all nodes. We assume that the agent observes the true label of one "anchor" node per group and must estimate all other group labels.

To do so, the agent aggregates all previously observed networks into a cumulative adjacency matrix $\mathbf{A}^{(t)}_{\text{cumulative}} = \sum_{i=1}^{t} \mathbf{A}_i$. An SBM with a Poisson likelihood is then fit to this matrix to cluster the nodes into K groups, producing an estimated group label for each node. The anchor nodes are key to avoid label switching, as the labels themselves are arbitrary once the groups have been defined.

While our regret bounds assume that the reward functions are correctly specified (and thus in this case node labels are observed), we show in Figure 4 that our algorithm still performs well when some labels are latent and must be estimated.

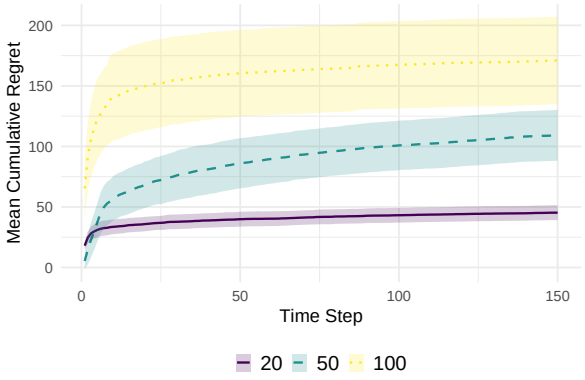

Figure 4: Cumulative regret under grouped SANIA reward function where group labels are estimated using one anchor node per group.

5.3 Heterogeneous Effects with Latent Features

We now consider a model where treatment effects are heterogeneous and depend on latent node characteristics. We use the latent feature parameterization from Section 3.2, where the reward for node i with latent features $\boldsymbol{\zeta}_i \in \mathbb{R}^2$ is given by:

$$r_i(\mathbf{Z}_t; \mathbf{A}_t, \boldsymbol{\zeta}_i) = Z_{t,i} \cdot \boldsymbol{\zeta}_i^\top \boldsymbol{\mu} + \sum_{k=1}^{d_{\max}} (\boldsymbol{\zeta}_i^\top \boldsymbol{\alpha}_k) \cdot \mathbf{1}_{\{d^1_{t,i}=k\}} + \epsilon_{t,i}.$$

The data-generating process uses the same underlying features to drive both network structure and rewards:

1. **Latent Feature Generation:** For each iteration, we generate latent features for $n \in$

$\{100, 500, 1000\}$ nodes by drawing $\boldsymbol{\zeta}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$ for $i = 1, \ldots, n$. These features are fixed for all t.

2. **Parameter and Reward Generation:** The true parameters are drawn once per run as $\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$ and $\boldsymbol{\alpha}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$ for $k = 1, \ldots, d_{\max}$. The noise is standard normal, $\epsilon_{t,i} \sim \mathcal{N}(0, 1)$.

3. **Network Generation:** At each time step t, the network $\mathbf{A}_t$ is generated from a latent space model. The probability of an edge between nodes i and j depends on the Euclidean distance between their features: $P([\mathbf{A}_t]_{ij} = 1) = \text{logit}^{-1}(\alpha - \|\boldsymbol{\zeta}_i - \boldsymbol{\zeta}_j\|_2)$. This process creates networks with community structures where nearby nodes in the latent space are more likely to be connected.

For this experiment, we assume the latent features $\boldsymbol{\zeta}_i$ are observable to the agent at decision time. The agent's task is to learn the unknown parameter vectors $\boldsymbol{\mu}$ and $\{\boldsymbol{\alpha}_k\}$. We run the simulation for $T = 500$ steps with a budget to treat 20% of nodes. For each n, we set α so that the expected degree of each node is approximately 5. We show the cumulative regret for multiple network sizes in Figure 5.

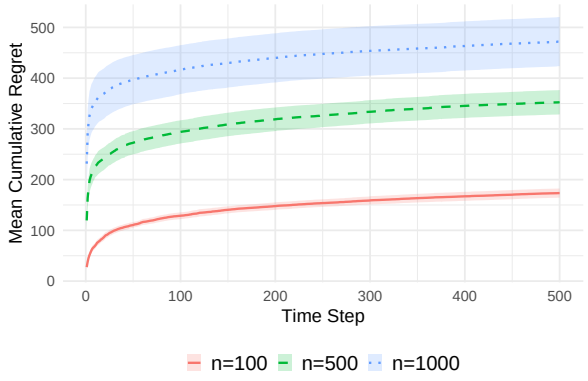

Figure 5: Cumulative regret plots for latent feature reward model.

5.4 Comparisons to Existing Methods

To compare our method to Agarwal et al. (2024) we postulate a most favorable data generating process (DGP) to their approach: the reward functions follow Equation 2 and remain fixed over time, and the network is small ($n = 8$) and static. We use the same parameter generation scheme as the grouped effects simulations.

In Figure 6, we see that our method significantly outperforms the more general Agarwal et al. (2024). We also compare our methods under misspecification of the SANIA assumptions, specifically by omitting Assumption 2 from the DGP . In Figure 7, we see that our method continues to outperform, despite misspecification of the SANIA assumptions.

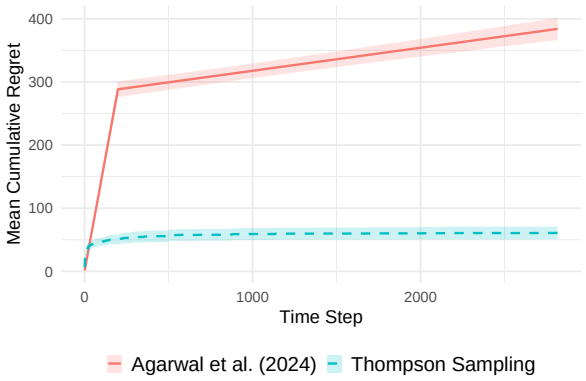

Figure 6: Comparison to Agarwal et al. (2024) on a static network of size $n = 8$.

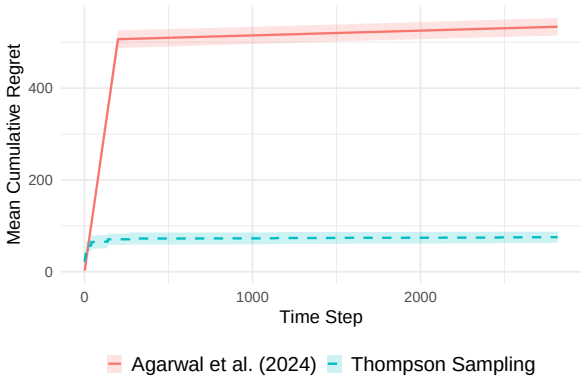

Figure 7: Comparison to Agarwal et al. (2024) under misspecification of the additivity assumption. Our method remains robust and continues to outperform.

5.5 SANIA Misspecification

We investigate the sensitivity of our algorithm to violations of the SANIA assumptions. The following reward function violates Assumption 3 as the contribution of a treated neighbor depends on that neighbors' own degree:

$$r_i = \mu_i Z_i + \sum_{j \in \mathcal{N}_i} A_{ij} Z_j \deg(j) + \epsilon_i.$$

We generate rewards from this model and fix an Erdös-Renyi graph with $n = 8$ and $p = 0.3$. We use the reward model from Section 5.1 which assumes symmetric spillovers and is thus misspecified.

In Figure 8 we compare directly to both Agarwal et al. (2024) and a standard MAB, as these more general algorithms do not suffer from misspecification. We fix a network over time as required by Agarwal et al.

(2024). Despite being misspecified, our algorithm significantly outperforms both alternatives. This is because the SANIA-based parameterization, while unable to capture the degree-dependent spillover effects exactly, learns an effective approximation with far fewer parameters. In contrast, Agarwal et al. (2024) must estimate an exponentially larger parameter vector, and the standard MAB must explore 2^n arms independently, resulting in substantially higher variance that dominates any bias advantage from correct specification.

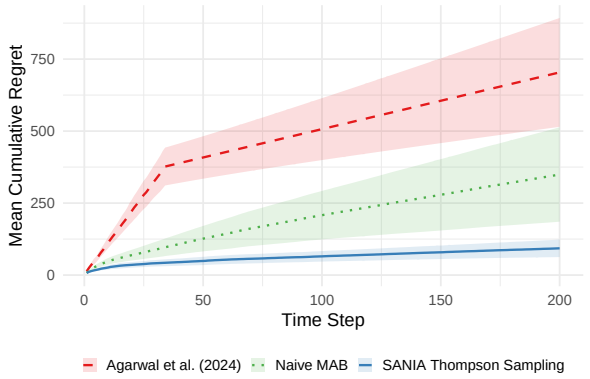

Figure 8: Comparison of Algorithm 1 to Agarwal et al. (2024) under symmetry misspecification.

5.6 Network Misspecification

Our algorithm and analysis assumes that the agent observes $\mathbf{A}_t$ at each time period. However, issues with data collection can cause existing edges to be unobserved or nonexistent edges falsely set to 1. We test our algorithms sensitivity to misspecified adjacency matrices by rerunning the experiment from Section 5.1 while randomly flipping a portion $\epsilon \in \{0, 0.05, 0.1, 0.2\}$ of the edges in $\mathbf{A}_t$. In Figure 9 we see that regret scales well even with 20% of the edges flipped.

6 CONCLUSION

We have introduced a scalable Thompson sampling algorithm for regret minimization under network interference. We show that the SANIA assumptions can be leveraged for scalable policy maximization, bridging a gap between the causal inference and bandits literature. We prove Bayesian regret bounds and provide strong evidence of performance and scalability through simulation experiments. Our work suggests future research into linear optimization algorithms specific to network interference. Extensions of our method to partially observed networks could greatly impact their practicality. Empirical validations beyond simulation experiments can test the impact on real-world outcomes.

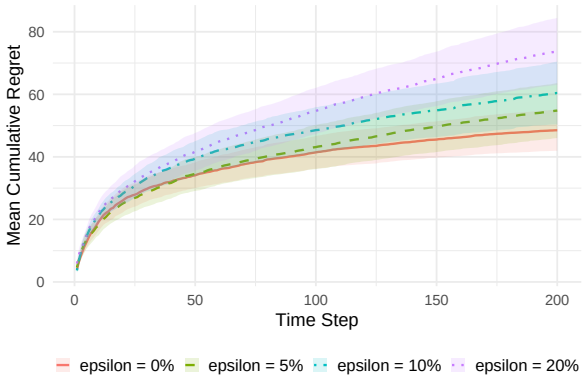

Figure 9: Cumulative regret under the linear spillover model for $n = 8$ under various noise levels that result in misspecifed $\mathbf{A}_t$.

Acknowledgments

We gratefully acknowledge funding from the National Science foundation under grants DMS 2346292, DMS 2230074, DMS CAREER 2046880, and the National Institute of Health under grant R01CA280970.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Agarwal, A., Agarwal, A., Masoero, L., and Whitehouse, J. (2024). Multi-armed bandits with network interference.

Agrawal, S. and Goyal, N. (2017). Near-optimal regret bounds for thompson sampling. *J. ACM*, 64(5).

Aronow, P. M. and Samii, C. (2017). Estimating average causal effects under general interference, with application to a social network experiment. *The Annals of Applied Statistics*, 11(4):1912–1947.

Belloni, A., Fang, F., and Volfovsky, A. (2025). Neighborhood adaptive estimators for causal inference under network interference.

Benjamin-Chung, J., Arnold, B. F., Berger, D., Luby, S. P., Miguel, E., Colford Jr, J. M., and Hubbard, A. E. (2017). Spillover effects in epidemiology: parameters, study designs and methodological considerations. *International Journal of Epidemiology*, 47(1):332–347.

Bramoullé, Y., Djebbari, H., and Fortin, B. (2020). Peer effects in networks: A survey. *Annual Review of Economics*, 12(Volume 12, 2020):603–629.

Chen, W., Wang, Y., and Yuan, Y. (2013). Combinatorial multi-armed bandit: General framework and applications. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 151–159, Atlanta, Georgia, USA. PMLR.

Eckles, D., Karrer, B., and Ugander, J. (2017). Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1):20150021.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman & Hall/CRC Texts in Statistical Science Series. CRC, Boca Raton, Florida, third edition.

Gurobi Optimization, LLC (2024). Gurobi Optimizer Reference Manual.

Hoff, P. D., Raftery, A. E., and and, M. S. H. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.

Hudgens, M. G. and Halloran, M. E. (2008). Toward causal inference with interference. *Journal of the American Statistical Association*, 103(482):832–842.

Jamshidi, F., Shahverdikondori, M., and Kiyavash, N. (2025). Graph-dependent regret bounds in multi-armed bandits with interference.

Jia, S., Frazier, P., and Kallus, N. (2024). Multi-armed bandits with interference.

Laber, E. B., Meyer, N. J., Reich, B. J., Pacifici, K., Collazo, J. A., and Drake, J. M. (2018). Optimal treatment allocations in space and time for on-line control of an emerging infectious disease. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 67(4):743–789.

Lattimore, T. and Szepesvári, C. (2020). *Bandit Algorithms*. Cambridge University Press.

Manski, C. F. (1993). Identification of endogenous social effects: The reflection problem. *The Review of Economic Studies*, 60(3):531–542.

Manski, C. F. (2013). Identification of treatment response with social interactions. *The Econometrics Journal*, 16(1):S1–S23.

McPherson, M., Smith-Lovin, L., and Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(Volume 27, 2001):415–444.

Munro, E., Kuang, X., and Wager, S. (2025). Treatment effects in market equilibrium.

Murphy, S. A. (2003). Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 65(2):331–355.

Ogburn, E. L., Sofrygin, O., Díaz, I., and van der Laan and, M. J. (2024). Causal inference for social network data. *Journal of the American Statistical Association*, 119(545):597–611.

Rothschild, M. (1974). A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 9(2):185–202.

Russo, D. and Van Roy, B. (2014). Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243.

Su, L., Lu, W., and Song, R. (2019). Modelling and estimation for optimal treatment decision with interference. *Stat*, 8(1):e219. e219 sta4.219.

Sussman, D. L. and Airoldi, E. M. (2017). Elements of estimation theory for causal effects in the presence of network interference.

Tchetgen, E. J. T. and VanderWeele, T. J. (2012). On causal inference in the presence of interference. *Statistical Methods in Medical Research*, 21(1):55–75. PMID: 21068053.

Toulis, P. and Kao, E. (2013). Estimation of causal peer influence effects. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1489–1497, Atlanta, Georgia, USA. PMLR.

Vaart, A. W. v. d. (1998). *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Viviano, D. (2024). Policy targeting under network interference. *The Review of Economic Studies*, 92(2):1257–1292.

Wager, S. and Xu, K. (2021). Experimenting in equilibrium. *Management Science*, 67(11):6694–6715.

Xu, Y., Lu, W., and Song, R. (2024). Linear contextual bandits with interference.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes (Sections 3 and 4)
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes (Section 4)
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes, in Supplement.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes (Section 4, and proofs in Supplement)
   (b) Complete proofs of all theoretical results. Yes (Supplement)
   (c) Clear explanations of any assumptions. Yes (Sections 3 and 4)

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes (Supplement)

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes.
   (b) The license information of the assets, if applicable. Yes.
   (c) New assets either in the supplemental material or as a URL, if applicable. Yes.
   (d) Information about consent from data providers/curators. Not Applicable.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable.

Supplementary Materials

A Linearity Under NIA

The neighborhood interference assumption (NIA) states that a node's outcome depends only on its own treatment status $Z_{t,i}$ and the treatment status of its neighbors $Z_{\mathcal{N}_{t,i}}$. As shown in Section 4.1.1 of Sussman and Airoldi (2017), this directly implies that the reward functions can be written as

$$r_i(\mathbf{Z}_t; \mathbf{A}_t) = \alpha_i + \beta_i Z_{t,i} + \Gamma_i(\mathbf{Z}_{\mathcal{N}_{t,i}}) + Z_{t,i}\Delta_i(\mathbf{Z}_{\mathcal{N}_{t,i}})$$

where

1. α_i is the baseline effect when neither the unit or any of its neighbors is treated.
2. β_i is the direct treatment effect.
3. Γ_i is the interference effect determined by the treatment status of the neighboring units of i.
4. Δ_i is the interaction effect between unit i's treatment status and the treatment of its neighbors.

Given the treatment vector $\mathbf{Z}_t$ and adjacency matrix $\mathbf{A}_t$, the ith row of the design matrix $H_i(\mathbf{Z_t}; \mathbf{A}_t)$ is simply a list of indicators that select the appropriate parameters from $\boldsymbol{\theta}$ corresponding to the linear formulation of $r_i(\mathbf{Z}_t; \mathbf{A}_t)$. Thus, the design matrix $\mathbf{H}$ can be constructed such that $\mathbf{r}_t(\mathbf{Z}_t; \mathbf{A}_t) = \mathbf{H}(\mathbf{Z}_t; \mathbf{A}_t)\boldsymbol{\theta} + \boldsymbol{\epsilon_t}$.

B Missing Proofs

B.1 Proof of Theorem 1

We prove Theorem 1 by first proving frequentist regret bounds for a linUCB style algorithm.

B.1.1 NetworkUCL Algorithm

We adapt the linUCB algorithm from Abbasi-Yadkori et al. (2011) to our setting. We prove novel regret bounds by an argument similar to Abbasi-Yadkori et al. (2011), adapting where necessary. We call this UCL-style algorithm networkUCL. It works by constructing confidence sets C_t for $\boldsymbol{\theta}$ using data observed up to time t. It then maximizes the rewards jointly over the decision set $\{0,1\}^n$ and the confidence set. We provide basic pseudocode in Algorithm 2.

Algorithm 2 networkUCL algorithm.

1: **for** $t = 1$ to T **do**
2: $\quad(\mathbf{Z}_t, \tilde{\boldsymbol{\theta}}) = \text{argmax}_{\mathbf{Z}\in\{0,1\}^n,\ \boldsymbol{\theta}\in C_{t-1}} \mathbf{H}(\mathbf{Z}_t; \mathbf{A}_t)\boldsymbol{\theta}$
3: $\quad$Play $\mathbf{Z}_t$, observe $\mathbf{r}_t$
4: $\quad$Update C_t
5: **end for**

B.1.2 NetworkUCL Regret Bound

We first require a bound on $\|\mathbf{H_t}^\top \boldsymbol{\epsilon}\|_{\overline{\mathbf{V}}_\mathbf{t}^{-1}}$, where $\overline{\mathbf{V}}_t = \lambda I + \sum_{s=1}^{t} \mathbf{H}_s^\top \mathbf{H}_s$, and $\boldsymbol{\epsilon}$ is a vector of independent sub-gaussian random variables. We follow a similar approach to Lemma 8, Lemma 9, and Theorem 1 of Abbasi-

Yadkori et al. (2011). However, these proofs work with the martingale

$$M_t^\lambda = \exp\left(\sum_{s=1}^{t}\left[\frac{\epsilon_s\langle\lambda, H_s\rangle}{R} - \frac{1}{2}\langle\lambda, H_s\rangle^2\right]\right).$$

Our setting requires working with a different martingale:

$$M_t^\lambda = \exp\left(\sum_{s=1}^{t}\left[\sum_{i=1}^{n}\frac{\epsilon_{s,i}\langle\lambda, [H_s]_i\rangle}{R} - \frac{1}{2}\langle\lambda, [H_s]_i\rangle^2\right]\right).$$

Throughout, we take $\{\mathcal{F}_t\}_{t=0}^\infty$ to be a filtration of σ-algebras such that $\mathbf{H}_t$ is $\mathcal{F}_{t-1}$ measurable and ϵ_t is $\mathcal{F}_t$ measurable. We begin by adapting Lemma 8 of Abbasi-Yadkori et al. (2011) to our model.

Lemma 1. *Let $\lambda \in \mathbb{R}^d$ be arbitrary and for any $t \geq 0$ consider*

$$M_t^\lambda = \exp\left(\sum_{s=1}^{t}\left[\sum_{i=1}^{n}\frac{\epsilon_{s,i}\langle\lambda, [X_s]_i\rangle}{R} - \frac{1}{2}\langle\lambda, [X_s]_i\rangle^2\right]\right).$$

Let τ be a stopping time with respect to the filtration $\{\mathcal{F}\}_{t=0}^\infty$. Then under Assumption 4, M_τ^λ is almost surely well-defined and $\mathbb{E}[M_\tau^\lambda] \leq 1$.

Proof. To prove that M_t^λ is a supermartingale, we must show that $\mathbb{E}[M_t^\lambda|\mathcal{F}_{t-1}] \leq M_{t-1}^\lambda$. We denote the overall contribution at time t as D_t^λ and the contribution of the ith row of $\mathbf{H}_t$ as $D_{t,i}^\lambda$: Let

$$D_t^\lambda = \exp\left(\sum_{i=1}^{n}\frac{\epsilon_{t,i}\langle\lambda, [X_t]_i\rangle}{R} - \frac{1}{2}\langle\lambda, [X_t]_i\rangle^2\right)$$
$$D_{t,i}^\lambda = \exp\left(\frac{\epsilon_{t,i}\langle\lambda, [X_t]_i\rangle}{R} - \frac{1}{2}\langle\lambda, [X_t]_i\rangle^2\right)$$

Notice that $D_t^\lambda = D_{t,1}^\lambda \cdot D_{t,2}^\lambda \cdot \ldots \cdot D_{t,n}^\lambda$. Now, by the 1-sub Gaussianity of $\epsilon_{t,i}$, we have that $\mathbb{E}[D_{t,i}^\lambda|\mathcal{F}_{t-1}] \leq 1$. Combining this with the independence of the ϵ's, we get

$$\mathbb{E}[D_t^\lambda|\mathcal{F}_{t-1}] = \mathbb{E}[D_{t,1}^\lambda|\mathcal{F}_{t-1}]\cdots\mathbb{E}[D_{t,i}^\lambda|\mathcal{F}_{t-1}] \leq 1.$$

We thus have

$$\begin{aligned}\mathbb{E}\left[M_t^\lambda|\mathcal{F}_{t-1}\right] &= \mathbb{E}\left[D_1^\lambda\cdots D_t^\lambda|\mathcal{F}_{t-1}\right]\\ &= D_1^\lambda\cdots D_{t-1}^\lambda\mathbb{E}\left[D_t^\lambda|\mathcal{F}_{t-1}\right]\\ &= M_{t-1}^\lambda\mathbb{E}\left[D_t^\lambda|\mathcal{F}_{t-1}\right] \leq 1.\end{aligned}$$

The fact that M_τ^λ is well-defined follows from the proof of Lemma 8 of Abbasi-Yadkori et al. (2011). □

With this result in hand, Lemma 9 and Theorem 1 from Abbasi-Yadkori et al. (2011) follow directly. We next need to extend Lemma 11 from Abbasi-Yadkori et al. (2011) to our setting. Our proof differs due to the batch structure of our updates to $\overline{V}_t$, however our result is effectively equivalent to theirs.

Lemma 2 (Adapted from Lemma 11 of Abbasi-Yadkori et al. (2011))**.** *Assume that for all t, the spectral norm of the design matrix is bounded such that $\|\mathbf{H}_t\|_2 \leq c_2$. Under the condition that the regularization parameter $\lambda \geq c_2^2$, it holds that*

$$\sum_{t=1}^{T}\sum_{i=1}^{n}\|[\mathbf{H}_t]_i\|^2_{\overline{\mathbf{V}}_{t-1}^{-1}} \leq 2(\log\det\overline{\mathbf{V}}_T - \log\det\mathbf{V})$$

where $\overline{\mathbf{V}}_t = \mathbf{V} + \sum_{s=1}^{t}\mathbf{H}_s^\top\mathbf{H}_s$ and $\mathbf{V} = \lambda\mathbf{I}$.

Proof. The proof adapts the core argument from Abbasi-Yadkori et al. (2011) to our setting, where the update term $\mathbf{H}_t^\top \mathbf{H}_t$ may have a rank greater than one.

First, we express the inner sum over the n nodes as a matrix trace. The term $\|[\mathbf{H}_t]_i\|^2_{\overline{\mathbf{V}}_{t-1}^{-1}}$ is a scalar and is thus equal to its own trace.

$$\begin{aligned}
\sum_{i=1}^{n} \|[\mathbf{H}_t]_i\|^2_{\overline{\mathbf{V}}_{t-1}^{-1}} &= \sum_{i=1}^{n} \operatorname{tr}\left([\mathbf{H}_t]_i \overline{\mathbf{V}}_{t-1}^{-1} [\mathbf{H}_t]_i^\top\right) \\
&= \sum_{i=1}^{n} \operatorname{tr}\left([\mathbf{H}_t]_i^\top [\mathbf{H}_t]_i \overline{\mathbf{V}}_{t-1}^{-1}\right) && \text{by the cyclic property of the trace} \\
&= \operatorname{tr}\left(\left(\sum_{i=1}^{n} [\mathbf{H}_t]_i^\top [\mathbf{H}_t]_i\right) \overline{\mathbf{V}}_{t-1}^{-1}\right) && \text{by the linearity of the trace} \\
&= \operatorname{tr}\left(\mathbf{H}_t^\top \mathbf{H}_t \overline{\mathbf{V}}_{t-1}^{-1}\right)
\end{aligned}$$

The quantity to bound is therefore $\sum_{t=1}^{T} \operatorname{tr}(\mathbf{H}_t^\top \mathbf{H}_t \overline{\mathbf{V}}_{t-1}^{-1})$.

Next, we relate this trace to the growth of the log-determinant of $\overline{\mathbf{V}}_t$. The determinant update rule is

$$\det(\overline{\mathbf{V}}_t) = \det(\overline{\mathbf{V}}_{t-1} + \mathbf{H}_t^\top \mathbf{H}_t) = \det(\overline{\mathbf{V}}_{t-1}) \cdot \det(\mathbf{I} + \overline{\mathbf{V}}_{t-1}^{-1} \mathbf{H}_t^\top \mathbf{H}_t).$$

Taking the logarithm, we have

$$\log\det(\overline{\mathbf{V}}_t) - \log\det(\overline{\mathbf{V}}_{t-1}) = \log\det(\mathbf{I} + \mathbf{A}_t),$$

where $\mathbf{A}_t = \overline{\mathbf{V}}_{t-1}^{-1} \mathbf{H}_t^\top \mathbf{H}_t$. Note that $\operatorname{tr}(\mathbf{A}_t) = \operatorname{tr}(\overline{\mathbf{V}}_{t-1}^{-1} \mathbf{H}_t^\top \mathbf{H}_t) = \operatorname{tr}(\mathbf{H}_t^\top \mathbf{H}_t \overline{\mathbf{V}}_{t-1}^{-1})$.

We use the inequality $x \le 2\log(1+x)$, which holds for $x \in [0, 1]$. This can be extended to matrices: for a positive semi-definite matrix $\mathbf{A}_t$ with all its eigenvalues in $[0, 1]$, we have $\operatorname{tr}(\mathbf{A}_t) \le 2\log\det(\mathbf{I} + \mathbf{A}_t)$. The condition on the eigenvalues of $\mathbf{A}_t$ is met if its largest eigenvalue, $\lambda_{\max}(\mathbf{A}_t)$, is at most 1. We bound this eigenvalue as follows:

$$\lambda_{\max}(\mathbf{A}_t) \le \|\mathbf{A}_t\|_2 \le \|\overline{\mathbf{V}}_{t-1}^{-1}\|_2 \|\mathbf{H}_t^\top \mathbf{H}_t\|_2 = \lambda_{\max}(\overline{\mathbf{V}}_{t-1}^{-1}) \|\mathbf{H}_t\|_2^2.$$

Since $\lambda_{\max}(\overline{\mathbf{V}}_{t-1}^{-1}) = 1/\lambda_{\min}(\overline{\mathbf{V}}_{t-1}) \le 1/\lambda$ and $\|\mathbf{H}_t\|_2 \le c_2$ by assumption, we have $\lambda_{\max}(\mathbf{A}_t) \le c_2^2/\lambda$. Our stated condition $\lambda \ge c_2^2$ ensures that $\lambda_{\max}(\mathbf{A}_t) \le 1$. Note that a similar assumption is made in Abbasi-Yadkori et al. (2011).

Thus, the inequality holds, and we have

$$\operatorname{tr}(\mathbf{H}_t^\top \mathbf{H}_t \overline{\mathbf{V}}_{t-1}^{-1}) \le 2\log\det(\mathbf{I} + \mathbf{A}_t) = 2\left(\log\det(\overline{\mathbf{V}}_t) - \log\det(\overline{\mathbf{V}}_{t-1})\right).$$

Summing this inequality from $t = 1$ to T yields a telescoping series:

$$\begin{aligned}
\sum_{t=1}^{T} \operatorname{tr}(\mathbf{H}_t^\top \mathbf{H}_t \overline{\mathbf{V}}_{t-1}^{-1}) &\le 2 \sum_{t=1}^{T} \left(\log\det(\overline{\mathbf{V}}_t) - \log\det(\overline{\mathbf{V}}_{t-1})\right) \\
&= 2\left(\log\det(\overline{\mathbf{V}}_T) - \log\det(\mathbf{V})\right).
\end{aligned}$$

This completes the proof. □

We now prove an upper bound on frequentist regret for our UCL algorithm.

Theorem 3. *Under the conditions of Lemma 1 and Lemma 2 above as well as Theorem 2 from Abbasi-Yadkori et al. (2011), then with probability* $1 - \delta$ *the networkUCL algorithm satisfies*

$$R_t \le \sqrt{nTD\log(\lambda + nTL/D)}(\lambda^{1/2}S + R\sqrt{2\log(1/\delta) + D\log(1 + nTL/(\lambda D)}$$

Proof. First, define $r_{t,i}$ as the regret accumulated by node i at time t. The following inequality comes directly from the proof of Theorem 3 from Abbasi-Yadkori et al. (2011). Following their notation, denote $[H_t]_i^*$ as the optimal feature vector for node i at time t. Additionally, θ^* denotes the true parameter vector, $\tilde{\theta}_t$ the optimistic choice, and $\hat{\theta}_t$ the OLS estimate.

$$\begin{aligned}
r_{t,i} &= \langle [H_t]_i^*, \theta^* \rangle - \langle [H_t]_i, \theta^* \rangle \\
&\leq \langle [H_t]_i, \tilde{\theta}_t \rangle - \langle \theta^*, [H_t]_i \rangle \\
&= \langle [H_t]_i, \tilde{\theta}_t - \theta^* \rangle \\
&= \langle [H_t]_i, \hat{\theta}_{t-1} - \theta^* \rangle + \langle \tilde{\theta}_t - \hat{\theta}_{t-1}, [H_t]_i \rangle \\
&\leq \|\hat{\theta}_{t-1} - \theta^*\|_{\overline{V}_{t-1}} \|[H_t]_i\|_{\overline{V}_{t-1}^{-1}} + \|\tilde{\theta}_t - \hat{\theta}\|_{\overline{V}_{t-1}} \|[H_t]_i\|_{\overline{V}_{t-1}^{-1}} \\
&\leq 2\sqrt{\beta_{t-1}(\delta)} \|[H_t]_i\|_{\overline{V}_{t-1}^{-1}} \qquad (3)
\end{aligned}$$

Using our previous lemma, we can now bound total cumulative regret. A simple extension of Lemma 10 from Abbasi-Yadkori et al. (2011) gives us that $\log\det(\overline{V}_T) \leq D\log(\lambda + nTL/D)$, and Theorem 2 directly gives us that $\beta_t(\delta) \leq R\sqrt{D\log((1+nTL/\lambda)/\delta)} + \lambda^{1/2}S$.

$$\begin{aligned}
R_t &= \sum_{t=1}^{T}\sum_{i=1}^{n} r_{t,i} = \sum_{t=1}^{T} \mathbf{r}_t^\top \mathbf{1}_n \\
&\leq \sum_{t=1} \sqrt{\|\mathbf{r}_t\|_2^2 \|\mathbf{1}_n\|_2^2} = \sqrt{n} \sum_{t=1}^{T} \|\mathbf{r}_t\|_2 \\
&\leq \sqrt{nT \sum_{t=1}^{T} \|\mathbf{r_t}\|_2^2} \\
&= \sqrt{nT \sum_{t=1}^{T}\sum_{i=1}^{n} r_{t,i}^2} \\
&\leq \sqrt{4nT\beta_{t-1}(\delta) \sum_{t=1}^{T}\sum_{i=1}^{n} \|[H_t]_i\|_{\overline{V}_{t-1}^{-1}}} && \text{by (1)} \\
&\leq \sqrt{8nT\beta_{t-1}(\delta) \log\det(\overline{V}_T)} && \text{by Lemma 2} \\
&\leq \sqrt{8nT\beta_{t-1}(\delta) D\log(\lambda + nTL/D)} \\
&\leq \sqrt{nTD\log(\lambda + nTL/D)}(\lambda^{1/2}S + R\sqrt{2\log(1/\delta) + D\log(1 + nTL/(\lambda D)}
\end{aligned}$$

□

B.2 Proof of Theorem 1

With Theorem 3, our result now follows directly from Russo and Van Roy (2014).

Proof. Given our frequentist regret bound, Theorem 1 follows directly from the proof of Proposition 3 in Russo and Van Roy (2014).

Specifically, Proposition 1 of Russo and Van Roy (2014) (combined with the regret decomposition of the Bayesian regret of a UCB algorithm) states that the Bayesian regret of Thompson sampling is less than or equal to the Bayesian regret of a UCB-style algorithm. Because our frequentist regret bounds are for a UBC-style algorithm, we can directly use this result to bound the Bayesian regret of Thompson sampling by integrating our frequentist bounds over the distribution of the unknown parameter vector. Such an integral results in bounds of the same asymptotic order, thus giving us the bound in Theorem 1. □

B.3 Proof of Theorem 2

Proof. The proof proceeds by first constructing a specific hard problem instance and then showing any algorithm must incur a certain amount of regret on that problem.

Step 1: Hard Instance Construction

Denote the set of possible parameter vectors as $\Theta = \{-\delta, \delta\}^D$ for some $\delta > 0$ to be chosen later. We consider a simple network of n disconnected nodes. We assign to each node j a fixed feature vector $w_j \in \{-1, 1\}^D$. These vectors are constructed from the rows of a $D \times D$ Hadamard matrix. The reward for a node j is linear in its features: $r_{t,j} = Z_{t,j}\langle w_j, \theta\rangle + \epsilon_{t,j}$, where $\epsilon_{t,j} \sim \mathcal{N}(0, 1)$. The design matrix $H(Z_t)$ is formed by stacking the row vectors w_j for all treated nodes j (where $Z_{t,j} = 1$).

To analyze the regret, we focus on a specific binary choice for each dimension $i \in \{1, \ldots, D\}$. We partition the nodes into two sets: $G_{i,A} = \{j \mid w_{ji} = +1\}$ and $G_{i,B} = \{j \mid w_{ji} = -1\}$. By the properties of Hadamard matrices, $|G_{i,A}| = |G_{i,B}| = n/2$. We define two actions:

- **Action** $Z_{i,A}$: Treat all nodes in group $G_{i,A}$.
- **Action** $Z_{i,B}$: Treat all nodes in group $G_{i,B}$.

If the true parameter is $\theta = \delta \cdot u_i$ (where u_i is the standard basis vector), the total expected reward for action $Z_{i,A}$ is $\sum_{j \in G_{i,A}} \langle w_j, \delta u_i\rangle = (n/2) \cdot \delta$. The total expected reward for $Z_{i,B}$ is $\sum_{j \in G_{i,B}} \langle w_j, \delta u_i\rangle = (n/2) \cdot (-\delta)$. The regret from making a single error by choosing the wrong action is therefore $n\delta$.

Step 2: Information-Theoretic Bound

We use the Bretagnolle-Huber inequality to lower bound the probability of error. Let's consider distinguishing between two parameter vectors θ and θ', where θ and θ' differ only on the i-th component, i.e., $\theta'_i = -\theta_i$. Let $p_{\theta i}$ be the probability of the "bad event" where an algorithm chooses the suboptimal action corresponding to dimension i for at least half of the T rounds. The inequality states

$$p_{\theta i} + p_{\theta' i} \geq \frac{1}{2} \exp\left(-\mathrm{KL}(\mathbb{P}_\theta || \mathbb{P}_{\theta'})\right).$$

The KL divergence is given by

$$\mathrm{KL}(\mathbb{P}_\theta || \mathbb{P}_{\theta'}) = \frac{1}{2} \sum_{t=1}^{T} \mathbb{E}\left[\|H(Z_t)(\theta - \theta')\|_2^2\right] = 2\delta^2 \sum_{t=1}^{T} \mathbb{E}\left[\|[H(Z_t)]_i\|_2^2\right]$$

In our construction, if the algorithm chooses $Z_{i,A}$ or $Z_{i,B}$, it treats $n/2$ nodes. The i-th column of the design matrix, $[H(Z_t)]_i$, will have $n/2$ entries of ± 1. Its squared L2-norm is $\|[H(Z_t)]_i\|_2^2 = n/2$. The KL divergence is thus bounded by

$$\mathrm{KL}(\mathbb{P}_\theta || \mathbb{P}_{\theta'}) \leq 2\delta^2 \sum_{t=1}^{T} \frac{n}{2} = \delta^2 nT$$

To ensure the KL divergence is a constant (e.g., 1), we choose $\delta = \frac{1}{\sqrt{nT}}$. Then, using the "averaging hammer" argument (Lattimore and Szepesvári, 2020) over all pairs (θ, θ') and all dimensions i, we can show that there exists a "hard" parameter vector θ^* for which $\sum_{i=1}^{D} p_{\theta^* i} \geq \frac{CD}{2}$ for some constant C.

Step 3: Regret Analysis

We now connect the probability of error to the total regret. For the constructed hard instance, the regret incurred from a single error on dimension i is $n\delta$.

$$\begin{aligned}
R_T(\theta^*) &= \mathbb{E}\left[\sum_{t=1}^{T} \text{regret}_t\right] \\
&\geq \sum_{i=1}^{D} \mathbb{E}\left[\sum_{t=1}^{T} n\delta \cdot \mathbf{1}\{\text{error on component } i \text{ at time } t\}\right] \\
&= n\delta \sum_{i=1}^{D} \mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\{\text{error on } i \text{ at time } t\}\right] \\
&\geq n\delta \frac{T}{2} \sum_{i=1}^{D} \mathbb{P}\left(\sum_{t=1}^{T} \mathbf{1}\{\text{error on } i\} \geq \frac{T}{2}\right) && \text{(by Markov's Inequality)} \\
&= \frac{n\delta T}{2} \sum_{i=1}^{D} p_{\theta^* i} \\
&\geq \frac{n\delta T}{2}\left(\frac{CD}{2}\right) && \text{(from the Averaging Hammer)} \\
&= \frac{CnDT\delta}{4} \\
&= \frac{CnDT}{4}\frac{1}{\sqrt{nT}} = \frac{C}{4} D\sqrt{nT}
\end{aligned}$$

Thus, we have shown that $R_T(\mathcal{A}, \theta^*) \geq \Omega(D\sqrt{nT})$. □

C Additional Experiments

C.1 Effect of the Regularization Parameter

The choice of λ does not impact our regret bounds as its effect diminishes asymptotically with $T \to \infty$. This can be seen through the posterior update formulas for the normal-normal Bayesian linear regression model. The posterior precision and mean are given by

$$\begin{aligned}
\Sigma_T^{-1} &= \lambda \mathbf{I}_D + \frac{1}{\sigma^2} \sum_{t=1}^{T} \mathbf{H}_t^T \mathbf{H}_t \\
\mu_T &= \left(\lambda \mathbf{I}_D + \frac{1}{\sigma^2} \sum_{t=1}^{T} \mathbf{H}_t^T \mathbf{H}_t\right)^{-1} \left(\frac{1}{\sigma^2} \sum_{t=1}^{T} \mathbf{H}_t^T \mathbf{r_t}\right)
\end{aligned}$$

As the amount of data increases, $\sum_{t=1}^{T} \mathbf{H}_t^\top \mathbf{H}_t$ "swamps out" the constant term $\lambda \mathbf{I}_d$ (Vaart, 1998).

However, the regularization parameter can have a significant impact on regret in the early stages of the algorithm. In Figure 10, we show the impact of λ on regret for the grouped model of Section 3.2.2. A large λ slows learning as the data struggles to overcome the weight of the prior, while a small λ can cause the algorithm to overfit initially, causing increased regret. We see in Figure 10 that $\lambda = 1$ appears optimal.

D Computing Resources

Figures were produced on a high-performance computing cluster using 50-75 CPUs in parallel, each assigned 1GB of RAM. The CPUs used were Intel(R) Xeon(R) Gold 6336Y. Gurobi software was used for all simulations. The authors received an academic license for free by registering through the Gurobi website.

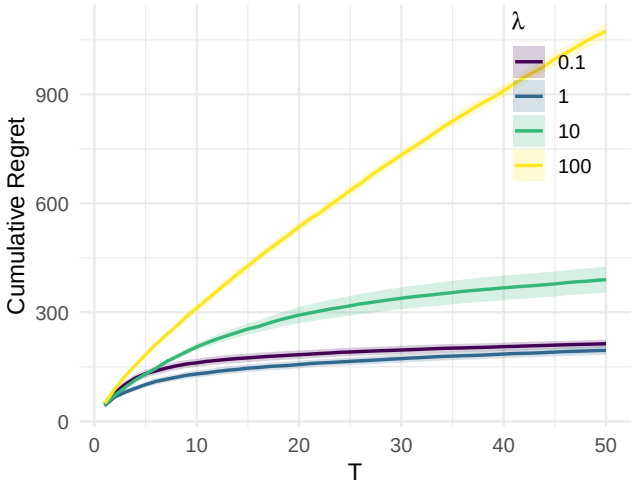

Figure 10: Impact of λ on regret for the grouped effects model.