# 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.

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 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 $\mathbf{Z}_t \in \{0, 1, \ldots, K\}^n$ potentially subject to

budget constraint $B_t$ (e.g. an upper limit on the number of treated nodes). 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 $\mathbf{A}_t$. For a given network $\mathbf{A}_t$, each reward function is thus a mapping $r_i(\cdot; \mathbf{A}_t) : \{0, 1, \ldots, K\}^n \to \mathbb{R}$. 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.

### 2.1 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 $(K+1)^n$ possible inputs. Under NIA, a node with degree $d$ would have a reward function with $(K+1)^{(d+1)}$ inputs.

Under NIA, the vector of node-level reward functions can be represented as linear in a parameter vector $\boldsymbol{\theta}$ containing direct and indirect effects (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 $(K+1)^{d+1}$ 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 $\tau$, $r_i(Z_{t,i}, \mathbf{Z}_{\mathcal{N}_{t,i}}) = r_i(Z_{t,i}, \tau(\mathbf{Z}_{\mathcal{N}_{t,i}}))$.*

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).

### 2.2 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

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

where we leave out an intercept parameter for simplicity. 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.

#### 2.2.1 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 $\gamma_k$ are 0, this reduces to the standard individualistic treatment response assumption. With non-zero 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.

#### 2.2.2 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: $\{\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_G^\top\}^\top$.

### 2.3 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.

### 2.4 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:

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

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

## 3 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.

### 3.1 Thompson Sampling Algorithm

We begin with the standard assumption on the noise $\boldsymbol{\epsilon}_t$.

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

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 $\boldsymbol{\theta}^{(t)}$.

**Algorithm 1** Thompson Sampling under Interference

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

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

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

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

**end for**

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.

### 3.2 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*

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

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 $\pi$, 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.

## 4 Simulations

We validate the flexibility and scalability of our method through simulation studies.

### 4.1 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 1 we evaluate our algorithm using the mean regret of the iterations and include 95% confidence bands.

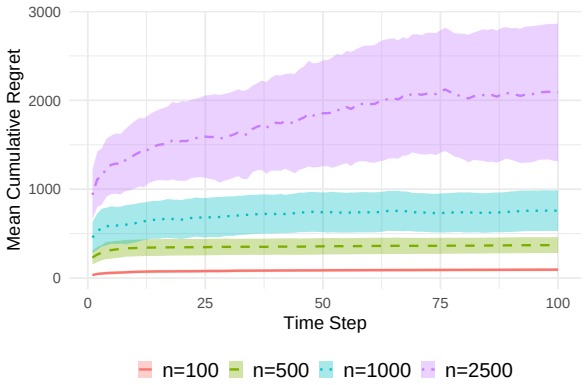


Figure 1: Cumulative regret plot under the shared parameters model.

## 5 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.

## 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.

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.

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.

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

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.

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

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.

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


## 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.