Neural Tangent Bayesian Optimization for Accurate and Efficient Influence Maximization

1st Zijian Zhang
CSE
Mississippi State University
Mississippi State, MS, USA
zz242@msstate.edu

2nd Zonghan Zhang
CSE
Mississippi State University
Mississippi State, MS, USA
zz239@msstate.edu

3rd Zhiqian Chen
CSE
Mississippi State University
Mississippi State, MS, USA
zchen@cse.msstate.edu

***Abstract*—Influence Maximization (IM) is a critical area of research with widespread applications in viral marketing, social network recommendations, and disease containment. The primary objective of IM is to identify an optimal seed set that maximizes influence spread across networks. Traditional approaches to IM, including proxy-based, sketch-based, and simulation-based methods, each face specific limitations. Proxy-based methods often fail to capture complex seed interactions and are model-specific, sketch-based methods balance scalability with accuracy but can introduce errors, and simulation-based techniques, while accurate, are computationally intensive, particularly for large-scale graphs. Additionally, the relationship between seed set configurations and their resulting influence spreads remains largely a black box, posing significant challenges in modeling and prediction without extensive computational effort. To address these challenges, we introduce the Neural Tangent BOIM (NT-BOIM) framework, which utilizes Bayesian Optimization (BO) to reduce the number of required simulations significantly. This approach employs the Neural Tangent Kernel (NTK) as the kernel for the Gaussian Processes(GP) in our BO framework, enhancing our ability to model the complex, high-dimensional data typical of social networks. The NTK provides a robust framework to analyze and predict the training dynamics of neural networks, making it particularly effective for understanding and optimizing influence spread across different seed sets. Our NT-BOIM methodology not only enhances the performance of IM tasks but also expedites the optimization process, offering a computationally efficient alternative to traditional methods. Key innovations include designing a specialized NTK that accurately quantifies distances between seed sets in graph structures and implementing a stratified sampling technique, preceded by clustering, to ensure uniform sampling distribution within each BO iteration. Extensive empirical experiments demonstrate that our approach outperforms standard simulation methods in both effectiveness and computational speed, bridging the gap between computational efficiency and approximation accuracy. [Code](https://github.com/XGraph-Team/NT-BOIM).**

***Index Terms*—Influence Maximization, Bayesian Optimization, Gaussian Process, Neural Tangent Kernel**

I. Introduction

In an increasingly networked world, the concept of IM has risen to prominence, attracting sustained interest from both the academic and industrial communities [6], [14]. The significance of IM is underscored by its wide-ranging applications, including viral marketing [4], personalized recommendations in social networks [30], rumor mitigation [9], and the containment of infectious diseases [16]. The core objective of IM is to identify a seed set of size k that optimizes the extent of influence propagation, commonly referred to as influence spread. Given that solving IM problems optimally is NP-hard, existing research often resorts to approximation techniques, primarily greedy algorithms [10], [26], [29], [31]. However, the suboptimal performance or computational inefficiency of current IM algorithms can lead to significant repercussions. As such, the timely identification of the most effective seed set remains a research imperative.

Current research on IM faces two major pain points: **Formulation of an Optimal Algorithm for IM:** IM methodologies predominantly bifurcate into two paradigms: proxy-based and simulation-based approaches. Proxy-based techniques emphasize computational efficiency and have evolved over time, employing heuristic estimations of nodal influence, while subsequent advancements strive for a more nuanced capture of propagation dynamics. Despite their computational advantages, these methods often exhibit limitations in capturing intricate seed interactions and are highly model-specific, thereby compromising approximation quality. Conversely, simulation-based methods prioritize approximation fidelity by iteratively selecting nodes that maximize marginal influence, though the computational burden escalates exponentially for large-scale graphs. **Modeling the Interplay between Seed Set and Influence Propagation:** The prevailing focus in existing IM literature has predominantly been the identification of influential seed sets, often neglecting the intricate relationship between seed selection and resultant influence spread. Recent attempts to dissect the influence contributions of individual seeds and their interdependencies using global sensitivity analysis still overlook the latent correlations between seed set configurations and ultimate influence outcomes. Another line of inquiry employs Graph Neural Networks as a predictive model for individual seed influence but is constrained by model transferability and computational overhead.

To address these challenges, we introduce the NT-BOIM, a method that leverages the sample efficiency of BO to significantly reduce the number of simulations required. We employ the NTK as the kernel for the GP in our BO framework. NTK represents a convergence point between neural networks and kernel methods, offering a framework to analyze the training dynamics of neural networks in the infinite-width

limit. This allows us to model the complex, high-dimensional data of social networks effectively, enhancing the prediction of influence spread across different seed sets.

To facilitate graph-level BO, we design a specialized NTK that accurately quantifies the distance between seed sets within the graph structure. Moreover, we implement a stratified sampling technique, preceded by clustering, to ensure a uniform distribution of sampled instances within each BO iteration. This methodology not only enhances performance but also expedites the optimization process, offering a more computationally efficient alternative to traditional simulation-based approaches for IM. Our primary contributions include:

- Proposing an efficient and effective simulation-based method that deviates from conventional methods by incorporating the consideration of seed interactions, enhancing the feasibility of the algorithm by substantially decreasing the number of simulations required.
- Providing theoretical support for the proposed NTK and sampling in IM, rigorously validating the kernel function through theoretical analysis, and providing a comprehensive theoretical framework that significantly mitigates variance.
- Conducting extensive empirical experiments to prove the superiority of NT-BOIM, employing a suite of both real-world and synthetic datasets that not only attain performance metrics on par with traditional methods but also exhibit a computational speedup.

II. Related Work

A. Influence Maximization

IM is recognized as a NP-hard problem, prompting researchers to explore feasible solutions with optimal performance. The first approximation approach proposed a simulation-based greedy algorithm, which, however, suffered from scalability issues [10]. Subsequent simulation-based methods aimed to improve performance or reduce complexity but still faced prohibitive computational costs, limiting their application to massive online networks [1], [8], [13], [25]. To alleviate the burden of simulations, proxy-based approaches emerged, approximating node spreading power using various proxies. Early proxies were simple heuristics like degree, PageRank [21], and eigen-centrality [33]. Later developments introduced influence-aware and diffusion model-aware proxies that offered more accurate estimations of seed influence spread [3], [11], [29], [31]. Common diffusion models, such as the Independent Cascade (IC) and Linear Threshold (LT), describe activation dynamics but typically abstract away the stochastic elements for analytical traceability. A comprehensive survey of influence maximization models, covering both static and dynamic networks, provides a detailed taxonomy and highlights new trends and challenges in detecting influential nodes [42]. Recently, deep graph representation learning methods have been proposed to address the challenges of IM, offering accelerated inference and improved scalability [43].

B. Bayesian Optimization

BO is employed for optimizing black-box functions that are costly to evaluate, constructing a probabilistic model of the objective function to guide the search for optimal solutions [15]. It has become a staple in hyperparameter tuning and optimization of complex simulations and models [24]. Bayesian probability theory underpins the method, where a prior distribution over function spaces is updated with observations, and an acquisition function determines subsequent evaluation points by balancing exploration with exploitation [22]. Researchers have extended BO to accommodate constraints, parallel evaluations, and high dimensions [5], [7]. Although BO has been applied over graph search spaces, most studies have focused on node-level tasks and developed specific kernels for node smoothing, which do not directly address IM optimization challenges [2], [17], [19], [20], [28]. Recent work has applied BO techniques to influence maximization, addressing the complexity of multiplex diffusion processes through scalable surrogate models [44].

C. Neural Tangent Kernel in Bayesian Optimization

The NTK bridges the gap between neural networks and kernel methods by describing the evolution of a neural network during training, particularly in infinitely wide networks where it becomes deterministic and time-independent [34]–[36]. This property allows for both theoretical analysis and practical applications. NTK's application within Bayesian Optimization (BO) has shown improvements in sample efficiency and scalability across various optimization tasks [37]–[39]. He et al. [40] explore the connection between deep ensembles and GP through the lens of NTK, providing a Bayesian interpretation for deep ensembles. This development is particularly relevant to IM, as it suggests potential avenues for quantifying uncertainty in influence spread predictions and improving robustness in dynamic network environments.

Recent work has extended the NTK framework to GNNs, leading to the development of Graph Neural Tangent Kernels (GNTKs). Krishnagopal and Ruiz [46] investigate the training dynamics of large-graph GNNs using GNTKs, providing theoretical insights into their convergence properties on large graphs. This advancement is significant for IM, as it enhances our understanding of how GNNs behave in large network settings, which is crucial for modeling influence propagation. The relevance of NTK to IM stems from its ability to capture complex relationships between seed sets and influence spread, modeling IM as a function optimization problem over the space of seed sets. By leveraging the NTK's capacity to approximate neural network behaviors in kernel space, we can more accurately model the non-linear dynamics of influence propagation in social networks. Despite its potential, challenges remain in designing appropriate kernels that effectively capture the structural properties of graphs and the dynamics of influence propagation. Integrating NTK with BO for IM offers a promising direction for enhancing the efficiency and effectiveness of IM algorithms, potentially leading to more accurate and robust solutions in complex network settings.

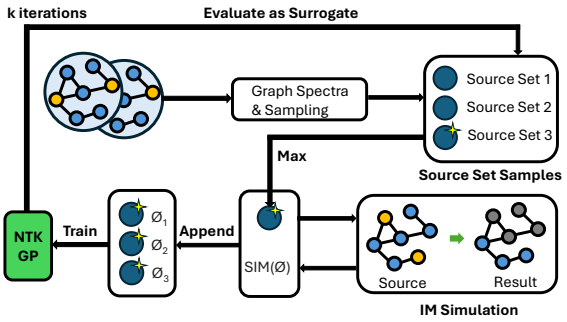

Fig. 1. Overview of NT-BOIM

III. Problem Setup

A graph is represented as a bidirectional structure $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the set of nodes and edges, respectively, and $|\mathcal{V}| = N$. Given this graph G, a predefined seed budget $k \in \mathbb{N}^+$, and a specific diffusion model d, the objective of an IM algorithm is to identify a seed set Ω of size k that approximately maximizes the expected influence spread $\phi(\Omega)$ (i.e., numbers of covered nodes). Mathematically, this can be formulated as:

$$\Omega = \arg\max_{\Omega} \phi(\Omega), \quad \text{s.t.} \quad |\Omega| \leq k. \tag{1}$$

To capture the underlying relationship between the seed set Ω and the influence spread $\phi(\Omega)$, we aim to approximate a function f such that $\phi(\Omega) \approx f(\Omega; G, d)$. Consequently, Equation 1 can be reformulated as:

$$\Omega = \arg\max_{\Omega} f(\Omega; G, d), \quad \text{s.t.} \quad |\Omega| \leq k. \tag{2}$$

IV. Method

NT-BOIM is a learning framework designed to identify influential nodes in a network, as depicted in Figure 1. This framework leverages BO, utilizing a GP model with different kernel functions to predict influence spread based on a given seed set of nodes. To enhance the accuracy of the GP model and ensure efficient exploration of the search space, we employ various sampling strategies, including stratified sampling, cluster-based sampling, normal (Gaussian) sampling, and random sampling. These strategies aim to address the challenges of high computational cost and diverse representation in selecting influential nodes. Initially, we perform simulations and utilize these sampling techniques to initialize the GP model effectively. Subsequent model updates are conducted iteratively, guided by the EI criterion to balance exploration and exploitation. The iterative process continues until the budget constraint is exhausted, and the final exploration of candidate cases reveals the optimal collection of influential nodes. This approach addresses the shortcomings of traditional methods, which often ignore graph structure and node attributes, leading to unreliable estimates of influence spread. By incorporating graph information through specialized kernels and using advanced sampling strategies, NT-BOIM provides a robust and efficient solution for IM in complex networks.

A. Reduce Search Space

Consider a graph with N nodes. Node sets are typically associated with a binary vector, where they are labeled as 1 if they are sources and 0 otherwise. This vector is represented as $s = \{0, 1\}^N$. With k sources, the total possible source configurations is $\binom{N}{k}$. Recognizing that not all nodes are equally significant in diffusion, like major cities in transport networks or key influencers in social networks, we focus on reducing the potential source combinations using various strategies tailored for different variants of our NT-BOIM approach. In general, the following techniques are employed across different NT-BOIM variants to manage and reduce the search space: Focus on Significant Nodes: Many variants prioritize nodes based on their degree centrality or other centrality measures. For instance, a common strategy involves selecting the top a nodes by degree, thereby reducing the search space to $\binom{a}{k}$ where $a \ll N$. Distance Maximization: To mitigate the overlap of influence regions, we maximize the inner distance among selected seed sets. This is done by ensuring that the shortest path between nodes in a seed set s is maximized:

$$d_s = \max_{s} \min_{u,v} d(u, v), \quad \forall u, v \in s \tag{3}$$

where $d(u, v)$ is the shortest node distance. Top d_s seed sets are chosen as candidates.

B. Kernel Design for Gaussian Process

A kernel that is appropriate and valid ensures that GPs reliably estimate the extent of influence propagation, given a specific seed set. One significant issue lies in the absence of graph structural information in the binary seed vector representation s. To illustrate, consider two 3-node sets: one original and the other formed by shifting each node by one hop based on the original set. Although it is anticipated that the final influence spread would be quite similar for these two sets due to their structural similarity, the similarity of the binary representations is actually quite low (0 in this case). This binary representation inadequately characterizes the similarity between two sets of nodes, as it disregards the underlying graph structure that significantly impacts influence propagation. Furthermore, the binary representation violates the smoothness assumption of GP, which posits that similar inputs should produce similar outputs. Without incorporating structural information, the GP model cannot leverage this assumption effectively, leading to unreliable estimates of influence spread. Previous work on graph kernels, including the Radial Basis Function (RBF) kernel, has primarily focused on structural comparisons, often ignoring node attributes [12], [18], [23], [27].

In order to overcome this constraint, we propose novel kernels that effectively combine graph structure information and attributes with theoretical validity. We introduce a new kernel based on NTK. This kernel leverages the expressiveness

of neural networks to capture complex patterns in graph-structured data. The NTK model in our implementation leverages the Neural Tangents library [41], a powerful tool that enables the computation of the NTK for infinitely wide neural networks. This infinite-width approximation offers a computationally efficient way to capture the complex interactions between graph structure and node features that influence the dynamics of information diffusion.

The NTK can be formulated as follows:

First, the input graph data is transformed using a neural network, where the transformation is parameterized by the neural network's weights and biases. Let $\phi_\theta(s)$ represent the output of the neural network given the seed vector s and parameters θ.

The NTK is then defined as:

$$K_{\text{NTK}}(s, s') = \mathbb{E}_\theta \left[\frac{\partial \phi_\theta(s)}{\partial \theta} \frac{\partial \phi_\theta(s')}{\partial \theta}^\top \right], \tag{4}$$

where the expectation is taken over the distribution of the neural network parameters θ.

To incorporate the graph Fourier transformation, we first transform the source vector s into its Fourier counterpart $\tilde{s}$ as follows:

$$\tilde{s} = U^\top s, \quad \tilde{s}(i) = \sum_{i=1}^{n} s_i U^\top(i), \tag{5}$$

where $U^\top$ is the matrix of eigenvectors of the graph Laplacian. Then, the Fourier-transformed NTK (FNTK) is defined as:

$$K_{\text{FNTK}}(s, s') = \mathbb{E}_\theta \left[\frac{\partial \phi_\theta(\tilde{s})}{\partial \theta} \frac{\partial \phi_\theta(\tilde{s'})}{\partial \theta}^\top \right]. \tag{6}$$

C. Proof of Validity as a Mercer Kernel

Theorem: The FNTK $K_{\text{FNTK}}(s, s')$ is a valid Mercer kernel.
Proof:

- Definition of the Kernel: The FNTK is defined as the expected inner product of the gradients of the neural network outputs with respect to the parameters θ after the input vectors have been transformed by the graph Fourier transformation. This expectation over the gradients can be seen as a form of averaging over multiple feature maps, where each feature map is defined by the neural network's gradients.
- Positive Semi-Definiteness:
  Assumption: We assume that the expectation $\mathbb{E}_\theta[\cdot]$ is well-defined and results in a finite kernel matrix.
  Consider any finite set of points $\{s_1, s_2, \ldots, s_n\}$ and form the kernel matrix K with entries $K_{ij} = K_{\text{FNTK}}(s_i, s_j)$. Let $\Phi_\theta(S)$ be the matrix where each row is the vectorized gradient (Jacobian) of $\phi_\theta(\tilde{s_i})$ with respect to θ, i.e., $\Phi_\theta(S) = \left[\frac{\partial \phi_\theta(\tilde{s_1})}{\partial \theta}^\top, \frac{\partial \phi_\theta(\tilde{s_2})}{\partial \theta}^\top, \ldots, \frac{\partial \phi_\theta(\tilde{s_n})}{\partial \theta}^\top \right]^\top$.
  The kernel matrix can be expressed as:

$$K = \mathbb{E}_\theta \left[\Phi_\theta(S) \Phi_\theta(S)^\top \right]. \tag{7}$$

For any vector $\mathbf{v} \in \mathbb{R}^n$, consider the quadratic form:

$$\begin{aligned} \mathbf{v}^\top K \mathbf{v} &= \mathbf{v}^\top \mathbb{E}_\theta \left[\Phi_\theta(S) \Phi_\theta(S)^\top \right] \mathbf{v} \\ &= \mathbb{E}_\theta \left[\mathbf{v}^\top \Phi_\theta(S) \Phi_\theta(S)^\top \mathbf{v} \right]. \end{aligned} \tag{8}$$

The term inside the expectation, $\mathbf{v}^\top \Phi_\theta(S) \Phi_\theta(S)^\top \mathbf{v}$, is the squared norm (Euclidean length) of the vector $\Phi_\theta(S)^\top \mathbf{v}$, and is therefore non-negative.
Since expectations preserve non-negativity, we have $\mathbf{v}^\top K \mathbf{v} \geq 0$ for all $\mathbf{v}$, proving that K is positive semi-definite.
Since the FNTK matrix K is positive semi-definite for any finite set of input points, the FNTK $K_{\text{FNTK}}(s, s')$ is a valid Mercer kernel.
Next, we set up a GP with the FNTK kernel to realize Equation 2. The purpose of this GP is to estimate the expected influence spread ϕ of the provided seed set s evaluated by simulation.

$$GP : s \to \phi(s). \tag{9}$$

D. Data Acquisition

To train our GP model effectively, we need pairs of seed sets (s_i) and their corresponding influence spreads $\phi(s_i)$, acquired through simulations. The objective is to select diverse and representative seed sets to minimize variance. We employ four principal sampling Techniques to initialize and iteratively train the GP model, balancing exploration and exploitation:

- Stratified Sampling: We employ stratified sampling to ensure diverse, representative selections from candidate seed sets. Mathematically, let $S = \{s_1, s_2, \ldots, s_n\}$ be all candidate sets. Partition S into k strata $S = \bigcup_{i=1}^{k} S_i$, where each S_i contains sets of similar size. For each stratum S_i, sample m_i sets, determined by the desired representation level.
- Clustered Sampling: Group S into ℓ clusters C_j based on a size-divisibility criterion, where $S = \bigcup_{j=1}^{\ell} C_j$. Each cluster C_j includes seed sets where the size modulo a constant factor (e.g., 5) is the same. Sample p_j seed sets from each cluster C_j, ensuring a varied size representation within the samples.
- Normal Distribution Sampling: Seed sets are selected based on a normal distribution with mean $\mu = \frac{N-1}{2}$ and standard deviation $\sigma = \frac{N}{6}$. Indices x_i are clipped to the range $[0, N-1]$ to ensure valid selections.
- Random Sampling: As a baseline, seed sets are randomly selected from S with equal probability, providing a comparison to structured sampling methods.

The selection of seed sets in each iteration is driven by Expected Improvement (EI), optimizing:

$$s^* = \arg\max_{s_i} \text{EI}(s_i) = \mathbb{E}[\delta(s_i, s^+) \cdot I(s_i)], \tag{10}$$

where $\delta(s_i, s^+) = f(s_i; o^*) - f(s^+; o^*)$ and $I(s_i)$ is an indicator function. Initial seed sets are chosen based on degree centrality. Influence spreads ϕ are estimated using simulations with models such as IC or LT, providing data for GP training.

Our integrated sampling strategies within a BO significantly enhance the efficiency and performance of IM algorithms.

E. Algorithm

NT-BOIM with Fourier transformation is outlined in Algorithm 1. Starting with a graph G and various configuration parameters, it aims to find a k-sized node set Ω that maximizes the expected influence spread ϕ. The algorithm computes the normalized Laplacian and its eigenvectors to create Fourier and inverse Fourier transform matrices. Candidate sets are generated and clustered based on the allowed shortest distance between nodes. A subset of candidate sets is sampled to form the training dataset. For each candidate set, a Fourier signal is created, and the diffusion model (IC or LT) calculates the influence spread, forming the training labels. An NTK model is initialized with the training data, using EI as the acquisition function. In each iteration, new candidate sets are sampled, and their Fourier signals are evaluated using the NTK model. The most promising candidate is selected, and its source set is found via the inverse Fourier transform. This set is evaluated using the diffusion model, and the NTK model is updated. This process repeats for the specified number of iterations. Finally, the best-performing source set is selected as the seed set Ω. This approach uses Fourier transforms to efficiently handle candidate sets in the spectral domain, while clustering and the NTK model balance exploration and exploitation during BO.

F. Time Complexity

We analyze the time complexity of NT-BOIM based on the provided code, focusing on the shared components of both NT-BOIM cluster and NT-BOIM stratified methods, as well as their unique elements, including the added Fourier transformations.

Preparation Phase:

- **Candidate Generation:** This process involves several steps: Calculating degrees of all nodes and sorting them: $O(N \log N)$. Generating all combinations of k from C candidates: $O\left(\binom{C}{k}\right)$. Calculating shortest paths between pairs of nodes in each combination, dominated by: $O\left(\binom{C}{k} \cdot (k^2 \cdot (M + N \log N))\right)$. Applying Fourier transforms to each candidate set to create signals: $O\left(\binom{C}{k} \cdot N \log N\right)$ for all candidate sets. Thus, the overall complexity for candidate generation is:
$$O\left(N \log N + \binom{C}{k} \cdot (k^2 \cdot (M + N \log N) + N \log N)\right)$$
- **Stratification or Clustering Candidate Sets:** This step groups candidate sets based on their size, which has a linear complexity relative to the number of candidate sets n: $O(n)$.
- **Model Initialization:** Initializing the model involves computing the kernel matrix, which is polynomial in the input size and number of parameters. We denote this complexity as $O(g(N, P))$, where P is the number of model parameters. Here, $g(N, P)$ represents the complexity of computing the kernel matrix, which is polynomial in the number of nodes (N) and model parameters (P).

GP Training and Prediction (Iterative Loop):

- **Sampling:** The complexity of sampling from strata or clusters is negligible compared to other steps, so we can consider it $O(1)$.
- **Influence Spread Simulation:** The complexity here remains $O(T)$, where T is the simulation time for one seed set. Perform |strata| or |clusters| simulations per iteration.
- **Model Update and Prediction:** This involves updating the model or NTK and making predictions. The complexity is $O(g(N, P))$, the same as initialization, as the kernel matrix needs to be recomputed.
- **EI Optimization:** The complexity of optimizing EI involves evaluating it for each sampled seed set, which is dominated by the simulation cost and Fourier transform application.

Considering these factors, the overall time complexity for both methods can be expressed as:

$$O\left(N \log N + \binom{C}{k} \cdot (k^2 \cdot (M + N \log N) + N \log N) + g(N, P) + \beta \cdot (|\text{strata/clusters}| \cdot (T + g(N, P)))\right)$$

where: $O(N \log N)$ is for sorting degrees and applying Fourier transforms. $\binom{C}{k} \cdot (k^2 \cdot (M + N \log N) + N \log N)$ is for generating and filtering candidate sets and signal creation. $g(N, P)$ is the complexity of the model initialization and update. β is the BO budget (number of iterations). |strata/clusters| is the number of strata or clusters. T is the simulation time for one seed set. If we assume the simulation time T dominates the other terms (as is often the case), the complexity simplifies to $O(\beta \cdot |\text{strata/clusters}| \cdot T)$. If the number of strata or clusters |strata/clusters| is a constant, the complexity further simplifies to $O(\beta T)$. Compared to the original simulation-based greedy method (GRD) [10], which performs N simulations to find the first seed, for each following seed, the number of simulations reduces by 1. Thus, the total number of simulations is $N + (N-1) + \cdots + (N-k+1) = kN - \frac{k^2-k}{2}$ and the time complexity is $O(kNT)$. CELF++ [13] improves the efficiency of the greedy algorithm by reducing the number of evaluations. CELF++ maintains a priority queue to keep track of the marginal gains of nodes and reduces unnecessary evaluations by leveraging lazy forward optimization. The theoretical time complexity of CELF++ is $O(N \log N + kNT)$. However, in practice, CELF++ performs fewer evaluations than the standard greedy method, making it faster in real-world scenarios.The time complexity for the fastest simulation-based IM, which is Sobol IM [32] (SIM), is $O(M)$, where M is a proxy-based IM algorithm that combines with the following simulations. It is claimed that SIM combined with Degree Discount (SIM-DD) could provide a good enough solution. Thus, the time complexity can be regarded as $O(k \log N)$. However, this time complexity considers the evaluation time negligible. As the evaluation is #P-hard, which is at least as hard as NP problems, we need to consider the evaluation time. Therefore,

the actual time complexity for SIM combined with degree discount is $O(k \log N + 2kT)$. Assuming that the complexity of T dominates N^3, we can put the time complexities of the four methods together: NT-BOIM: $O(\beta T)$, GRD: $O(kNT)$, SIM-DD: $O(2kT)$, CELF++: $O(N \log N + kNT)$.

V. Experimental Setup

We assess the NT-BOIM approach using the Cora and CiteSeer datasets, representing citation networks in machine learning and computer science, respectively. These datasets are comprised of 2,708 and 3,312 publications and model real-world social network characteristics, such as clustering and power-law degree distributions. Influence propagation is simulated using both the IC and LT models to reflect different mechanisms of information spread in networks. The performance of NT-BOIM is benchmarked against state-of-the-art IM methods, including greedy algorithms and heuristic-based approaches. Evaluation metrics focus on influence spread and computational efficiency. Experiments are conducted with various random initializations to ensure robustness, and statistical tests determine the significance of performance differences. Additional tests explore the impact of parameters like candidate set size, number of initial simulations, and the type of acquisition function used in BO, providing insights into the optimal configuration of NT-BOIM. The setup aims to comprehensively evaluate the efficacy of NT-BOIM in academic citation networks and understand its sensitivity to different design choices. The **Influence Spread (eval)** measures the coverage achieved by the seed nodes, where higher values indicate greater IM, and the **Computational Time (runtime)** assesses the method's efficiency, recorded in seconds.

VI. Results Analysis

This section evaluates various IM methods on the CITESEER and CORA datasets under IC and LTmodels, analyzing performance via Eval Mean and Eval Std metrics, with a focus on the proposed NT-BOIM method.

A. CITESEER Dataset

- In the LT Model, NT-BOIM methods with cluster sampling demonstrate potential at 913.962, though with higher variability, suggesting room for optimization. CELF++ remains the top performer with 1014.300, while IMRank closely follows at 979.220.
- In the IC Model, NT-BOIM methods show strong performance. Notably, the NT-BOIM stratified achieves 646.584, closely approaching the effectiveness of CELF++, which leads with the highest performance of 658.720 and low variability. IMRank also shows solid performance at 619.630.

B. CORA Dataset

- In the LT Model, NT-BOIM methods underperform compared to traditional methods such as degree-based strategies. However, they still offer adaptability and potential for further optimization. CELF++ again leads with top performance at 1719.560.
- In the IC Model, NT-BOIM methods demonstrate robust performance with cluster (1146.188) and stratified (1164.200) sampling, closely trailing CELF++, which has the highest performance at 1227.310. IMRank also shows strong performance at 1143.440.

C. Sampling Strategies

Cluster Sampling often shows the highest mean among NT-BOIM methods but with considerable variability, indicating a potential for capturing network structure effectively. Stratified Sampling exhibits strong performance, especially in IC models, suggesting it achieves a good balance in seed selection. Random and Normal Sampling are lower performing, providing baselines for assessing more sophisticated strategies. NT-BOIM methods are competitive, particularly in IC models, where the cascading nature of influence spread can be effectively modeled. While CELF++ consistently outperforms across datasets and models, NT-BOIM shows significant potential, especially with cluster and stratified sampling strategies. The performance of NT-BOIM in IC models versus LT models highlights its suitability for scenarios that align closely with the probabilistic nature of IC models. This suggests that NT-BOIM offers a valuable and flexible tool for IM tasks where rapid, cascading influence patterns are prevalent.

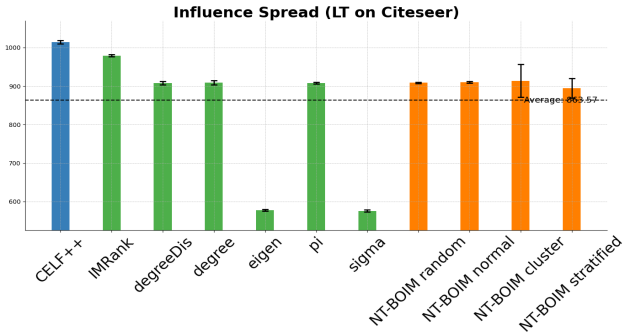

Fig. 2. NT-BOIM performs well with LT on CiteSeer

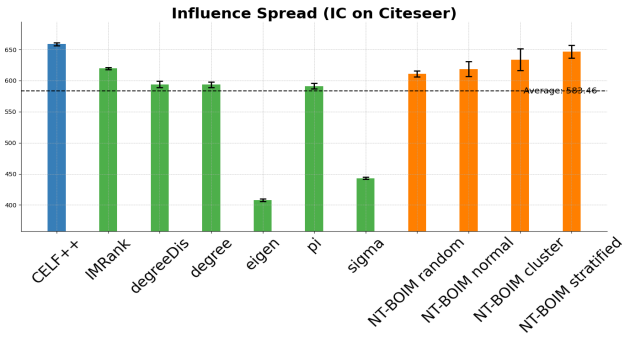

Fig. 3. NT-BOIM demonstrates competitiveness with IC on CiteSeer

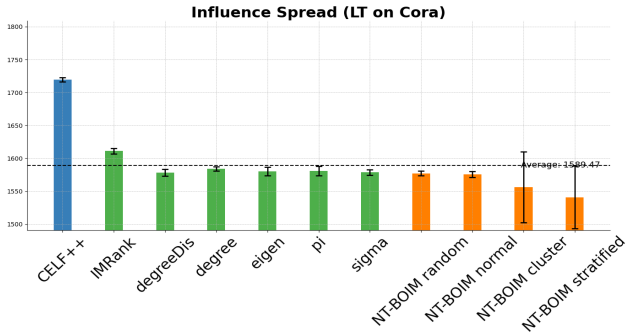

Fig. 4. NT-BOIM needs improvement with LT on Cora

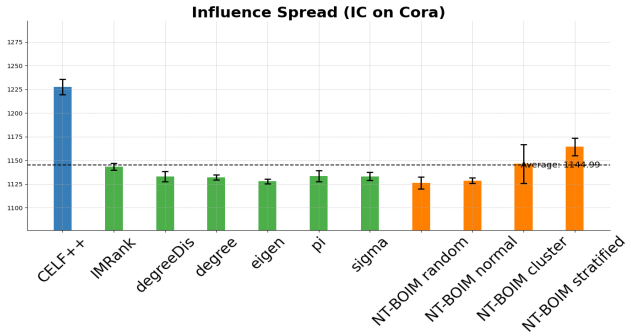

Fig. 5. NT-BOIM shows robust performance with IC on Cora

D. Runtime

According to the Table 1, the NT-BOIM variants consistently achieve performance close to CELF++ in terms of evaluation mean, with an average of 96% on the CiteSeer dataset and 93% on the Cora dataset. Additionally, they significantly outperform CELF++ in runtime, being on average 59% faster on CiteSeer and 72% faster on Cora.

VII. Conclusion

We introduce NT-BOIM, a novel IM approach utilizing FNTK to enhance the efficiency and accuracy of simulations. This method leverages BO to unravel complex relationships between seed sets and influence spread, demonstrating competitive performance across various datasets. Through theoretical validation and empirical experiments, NT-BOIM proves to be a robust tool for network analysis, simplifying the computational demands while capturing detailed network dynamics more effectively than traditional methods.

TABLE I
Performance Comparison of NT-BOIM Variants and CELF++ on CiteSeer and Cora Datasets

Dataset	Method	Eval Mean ± Std	Runtime (s)
		IC Model	
CiteSeer	NT-BOIM random	610.724 ± 4.866 (93.73%)	770.914 ± 41.235 (40.60%)
	NT-BOIM normal	618.596 ± 12.089 (94.94%)	780.188 ± 22.268 (41.09%)
	NT-BOIM cluster	633.766 ± 17.651 (97.27%)	784.712 ± 13.815 (41.33%)
	NT-BOIM stratified	646.584 ± 10.421 (99.23%)	789.034 ± 11.425 (41.56%)
	CELF++	651.576 ± 5.526 (100%)	1898.712 ± 24.592 (100%)
Cora	NT-BOIM random	1126.260 ± 6.243 (91.77%)	949.515 ± 6.235 (27.77%)
	NT-BOIM normal	1128.568 ± 2.792 (91.95%)	956.291 ± 5.663 (28.00%)
	NT-BOIM cluster	1146.188 ± 20.653 (93.39%)	953.594 ± 4.130 (27.91%)
	NT-BOIM stratified	1164.200 ± 9.067 (94.86%)	954.945 ± 3.082 (27.95%)
	CELF++	1227.312 ± 8.071 (100%)	3418.713 ± 37.876 (100%)
		LT Model	
CiteSeer	NT-BOIM random	908.494 ± 1.512 (89.57%)	1167.650 ± 54.464 (35.00%)
	NT-BOIM normal	909.904 ± 2.240 (89.71%)	1184.070 ± 29.279 (35.49%)
	NT-BOIM cluster	913.962 ± 42.284 (90.11%)	1186.563 ± 18.313 (35.57%)
	NT-BOIM stratified	894.992 ± 25.520 (88.24%)	1191.730 ± 14.691 (35.72%)
	CELF++	1014.302 ± 4.113 (100%)	3335.889 ± 18.772 (100%)
Cora	NT-BOIM random	1577.268 ± 3.743 (91.78%)	1442.369 ± 11.248 (24.79%)
	NT-BOIM normal	1575.928 ± 4.611 (91.70%)	1445.847 ± 5.088 (24.85%)
	NT-BOIM cluster	1556.526 ± 53.785 (90.57%)	1445.673 ± 1.183 (24.84%)
	NT-BOIM stratified	1540.994 ± 47.642 (89.67%)	1447.492 ± 3.123 (24.87%)
	CELF++	1718.500 ± 2.999 (100%)	5819.102 ± 27.527 (100%)

References

[1] Akhil Arora, Sainyam Galhotra, and Sayan Ranu, "Debunking the myths of influence maximization: An in-depth benchmarking study," in *Proceedings of the 2017 ACM international conference on management of data*, pp. 651–666, 2017.

[2] Viacheslav Borovitskiy, Iskander Azangulov, Alexander Terenin, Peter Mostowsky, Marc Deisenroth, and Nicolas Durrande, "Matern gaussian processes on graphs," in *International Conference on Artificial Intelligence and Statistics*, pp. 2593–2601. PMLR, 2021.

[3] Wei Chen, Yajun Wang, and Siyu Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 199–208, 2009.

[4] Wei Chen, Chi Wang, and Yajun Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1029–1038, 2010.

[5] Nando de Freitas and Ziyu Wang, "Bayesian optimization in high dimensions via random embeddings," 2013.

[6] Pedro Domingos and Matt Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 57–66, 2001.

[7] Javier Gonzalez, Zhenwen Dai, Philipp Hennig, and Neil Lawrence, "Batch bayesian optimization via local penalization," in *Artificial intelligence and statistics*, pp. 648–657. PMLR, 2016.

[8] Amit Goyal, Wei Lu, and Laks V.S. Lakshmanan, "Celf++ optimizing the greedy algorithm for influence maximization in social networks," in *Proceedings of the 20th international conference companion on World wide web*, pp. 47–48, 2011.

[9] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang, "Influence blocking maximization in social networks under the competitive lin-

ear threshold model," in *Proceedings of the 2012 SIAM international conference on data mining*, pp. 463–474. SIAM, 2012.

[10] David Kempe, Jon Kleinberg, and Eva Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146, 2003.

[11] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda, "Blocking links to minimize contamination spread in a social network," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 2, pp. 1–23, 2009.

[12] Nils M Kriege, Fredrik D Johansson, and Christopher Morris, "A survey on graph kernels," *Applied Network Science*, vol. 5, no. 1, pp. 1–42, 2020.

[13] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 420–429, 2007.

[14] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan, "Influence maximization on social graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 10, pp. 1852–1872, 2018.

[15] Jonas Mockus, "The application of bayesian methods for seeking the extremum," in *Towards global optimization*, vol. 2, pp. 117, 1998.

[16] Mark EJ Newman, "Spread of epidemic disease on networks," *Physical Review E*, vol. 66, no. 1, 016128, 2002.

[17] Yin Cheng Ng, Nicolo Colombo, and Ricardo Silva, "Bayesian semi-supervised learning with graph gaussian processes," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[18] Giannis Nikolentzos, Giannis Siglidis, and Michalis Vazirgiannis, "Graph kernels: A survey," *Journal of Artificial Intelligence Research*, vol. 72, pp. 943–1027, 2021.

[19] Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling, "Combinatorial bayesian optimization using the graph cartesian product," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[20] Felix Opolka, Yin-Cong Zhi, Pietro Lio, and Xiaowen Dong, "Adaptive gaussian processes on graphs via spectral graph wavelets," in *International Conference on Artificial Intelligence and Statistics*, pp. 4818–4834, PMLR, 2022.

[21] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, "The pagerank citation ranking: Bringing order to the web," *Technical report*, Stanford InfoLab, 1999.

[22] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[23] Giannis Siglidis, Giannis Nikolentzos, Stratis Limnios, Christos Giatsidis, Konstantinos Skianis, and Michalis Vazirgiannis, "Grakel: A graph kernel library in python," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 1993–1997, 2020.

[24] Jasper Snoek, Hugo Larochelle, and Ryan P Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[25] Youze Tang, Xiaokui Xiao, and Yanchen Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 75–86, 2014.

[26] Hanghang Tong, B. Aditya Prakash, Charalampos Tsourakakis, Tina Eliassi-Rad, Christos Faloutsos, and Duen Horng Chau, "On the vulnerability of large graphs," in *2010 IEEE International Conference on Data Mining*, pp. 1091–1096, IEEE, 2010.

[27] S. Vichy N. Vishwanathan, Nicol N. Schraudolph, Risi Kondor, and Karsten M. Borgwardt, "Graph kernels," *Journal of Machine Learning Research*, vol. 11, pp. 1201–1242, 2010.

[28] Ian Walker and Ben Glocker, "Graph convolutional Gaussian processes," in *International Conference on Machine Learning*, pp. 6495–6504, PMLR, 2019.

[29] Ruidong Yan, Deying Li, Weili Wu, Ding-Zhu Du, and Yongcai Wang, "Minimizing influence of rumors by blockers on social networks: algorithms and analysis," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1067–1078, 2019.

[30] Mao Ye, Xingjie Liu, and Wang-Chien Lee, "Exploring social influence for recommendation: a generative model approach," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 671–680, 2012.

[31] Zonghan Zhang, Subhodip Biswas, Fanglan Chen, Kaiqun Fu, Taoran Ji, Chang-Tien Lu, Naren Ramakrishnan, and Zhiqian Chen, "Blocking influence at collective level with hard constraints (student abstract)," 2022.

[32] Zonghan Zhang and Zhiqian Chen, "Understanding influence maximization via higher-order decomposition," in *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pp. 766–774, SIAM, 2023.

[33] Lin-Feng Zhong, Ming-Sheng Shang, Xiao-Long Chen, and Shi-Ming Cai, "Identifying the influential nodes via eigen-centrality from the differences and similarities of structure," *Physica A: Statistical Mechanics and its Applications*, vol. 510, pp. 77–82, 2018.

[34] Arthur Jacot, Franck Gabriel, and Clement Hongler, "Neural Tangent Kernel: Convergence and Generalization in Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[35] Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington, "Wide Neural Networks of Any Depth Evolve as Linear Models Under Gradient Descent," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[36] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Russ R. Salakhutdinov, and Ruosong Wang, "On Exact Computation with an Infinitely Wide Neural Net," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[37] Sanjeev Arora, Simon S. Du, Zhiyuan Li, Russ R. Salakhutdinov, Ruosong Wang, and Dingli Yu, "Harnessing the Power of Infinitely Wide Deep Nets on Small-data Tasks," in *International Conference on Learning Representations (ICLR)*, 2019.

[38] Roman Novak, Lechao Xiao, Yasaman Bahri, Jaehoon Lee, Greg Yang, Jiri Hron, Jascha Sohl-Dickstein, "Bayesian Deep Learning and a Probabilistic Perspective of Generalization," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[39] Greg Yang, Sammy Huang, Zihang Cheng, Yichen Li, and Alex Smola, "Bayesian Active Learning with Fully Bayesian Gaussian Processes," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[40] Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh, "Bayesian Deep Ensembles via the Neural Tangent Kernel," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[41] Novak, Roman and Xiao, Lechao and Hron, Jiri and Lee, Jaehoon and Alemi, Alexander A and Sohl-Dickstein, Jascha and Schoenholz, Samuel S, "Neural Tangents: Fast and Easy Infinite Neural Networks in Python," *International Conference on Learning Representations*, 2020.

[42] Jaouadi, Myriam and Romdhane, Lotfi Ben, "A Survey on Influence Maximization Models," *Expert Systems with Applications*, volume 248, page 123429, 2024.

[43] Chowdhury, Tanmoy and Ling, Chen and Jiang, Junji and Wang, Junxiang and Thai, My T and Zhao, Liang, "Deep Graph Representation Learning Influence Maximization with Accelerated Inference," *Neural Networks*, page 106649, 2024.

[44] Yuan, Zirui and Shao, Minglai and Chen, Zhiqian, "Graph Bayesian Optimization for Multiplex Influence Maximization," *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, number 20, pages 22475–22483, 2024.

[45] Wang, Zi and Dahl, George E. and Swersky, Kevin and Lee, Chansoo and Nado, Zachary and Gilmer, Justin and Snoek, Jasper and Ghahramani, Zoubin, "Pre-trained Gaussian Processes for Bayesian Optimization," *Journal of Machine Learning Research*, volume 25, number 212, pages 1–83, 2024.

[46] Krishnagopal, Sanjukta and Ruiz, Luana, "Graph Neural Tangent Kernel: Convergence on Large Graphs," *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 17827–17841, 2023.