SILVAN: Estimating Betweenness Centralities with Progressive Sampling and Non-uniform Rademacher Bounds[⋆]

Leonardo Pellegrina[0000−0002−6601−5526] and Fabio Vandin[0000−0003−2244−2320]

University of Padova, Department of Information Engineering, Padova, Italy
{leonardo.pellegrina,fabio.vandin}@unipd.it

Abstract. Betweenness centrality is a popular centrality measure with applications in several domains, and whose exact computation is impractical for modern-sized networks. We present SILVAN, a novel, efficient algorithm to compute, with high probability, accurate estimates of the betweenness centrality of all nodes of a graph and a high-quality approximation of the top-k betweenness centralities. SILVAN follows a progressive sampling approach, and builds on novel bounds based on Monte-Carlo Empirical Rademacher Averages, a powerful and flexible tool from statistical learning theory. SILVAN relies on a novel estimation scheme providing *non-uniform* bounds on the deviation of the estimates of the betweenness centrality of all the nodes from their true values, and a refined characterisation of the number of samples required to obtain a high-quality approximation. Our extensive experimental evaluation shows that SILVAN extracts high-quality approximations while outperforming, in terms of number of samples and accuracy, the state-of-the-art approximation algorithm with comparable quality guarantees.

Keywords: Betweenness Centrality · Rademacher Averages · Random Sampling.

1 Introduction

The computation of node centrality measures, which are scores quantifying the importance of nodes, is a fundamental task in graph analytics [18]. *Betweenness centrality* is a popular centrality measure, defined first in sociology [1, 12], that quantifies the importance of a node as the fraction of shortest paths in the graph that go through the node.

The computation of the *exact* betweenness centrality for all nodes in a graph $G = (V, E)$ can be obtained with Brandes' algorithm [8] in time $\mathcal{O}(|V||E|)$ for unweighted graphs and in time $\mathcal{O}\left(|V||E| + |V|^2 \log |V|\right)$ for graphs with positive weights, which is impractical for modern networks with up to hundreds of millions of nodes and edges. Several works (e.g., [11, 32]) proposed heuristics to

[⋆] The extended version of this work is available online at [22].

improve Brandes' algorithm, but they do not improve on its worst-case complexity. In fact, for unweighted graphs a corresponding lower bound (based on the Strong Exponential Time Hypothesis) was proved in [5]. The impracticality of the exact computation for modern networks, and the use of betweenness centrality mostly in exploratory analyses of the data, have motivated the study of efficient algorithms to compute approximations of the betweenness centrality, trading precision for efficiency.

Several works [24, 27, 6, 9], have recently proposed sampling approaches to approximate the betweenness centrality of all nodes in a graph. The main idea is to sample shortest paths uniformly at random and use such paths to estimate the betweenness centrality of the nodes. As for all sampling approaches, the main difficulty is then to relate the estimates obtained from the samples with the corresponding exact quantities, providing tight trade-offs between guarantees on the quality of the estimates and the required computational work. To do so, these methods rely on sophisticated probabilistic and statistical learning concepts, such as the *VC-dimension* [35], the *pseudodimension* [23], or *Rademacher Averages* [14, 2], which have been successfully used to obtain rigorous approximations for other data mining tasks (e.g., pattern mining [25, 26, 28, 20]).

We defer to the full version of this work [22] a more detailed introduction of previous works to approximate the betweenness centrality and related problems due to space constraints. However, despite all these contributions, computing accurate approximations of the betweenness centrality on large graphs is still expensive and demanding in practice. This is the challenge we tackle in our work.

Our contributions In this work we study the problem of approximating the betweenness centralities of nodes in a graph. We propose SILVAN (*eStimatIng betweenness centraLities with progressiVe sAmpling and Non-uniform rademacher bounds*), a novel, efficient, progressive sampling algorithm to approximate betweenness centralities while providing rigorous guarantees on the quality of various approximations.

- Our first contribution is *empirical peeling*, a novel technique that we introduce to obtain sharp *non-uniform data-dependent bounds* on the maximum deviation of families of functions (Section 3.1). Empirical peeling is based on the Monte Carlo Empirical Rademacher Average (MCERA, defined in Section 2.2) and relies on an effective *data-dependent* approach to *partition* a family of functions according to their empirically estimated variance; this allows to fully exploit *variance-dependent* bounds at the core of the technique. Our algorithm SILVAN (Section 3.2) relies on such novel bounds to provide guarantees on the approximation of the betweenness centrality that are much sharper than the ones obtained by previous works; these new contributions make SILVAN a practical algorithm for obtaining different approximations of the betweenness centrality. In fact, we show that combining the MCERA with empirical peeling allows us to design flexible algorithms with different guarantees (e.g., additive or relative) and for different tasks (e.g., estimating all betweenness centralities or the top-k ones). This is the first work that obtains different types of approximation guarantees based on the MCERA. Most importantly, our approach is general and of independent interest, as it may apply to other problems, even outside of data mining applications.

- We derive a new bound on the sufficient number of samples to approximate the betweenness centrality for all nodes (presented in the full version [22]), that naturally combines with the progressive sampling strategy of SILVAN by introducing an upper limit to the number of samples required to converge. Our new bound is governed by key quantities of the underlying graph, not considered by previous works, such as the *average shortest path length*, and the *maximum variance* of betweenness centrality estimators, significantly improving the state-of-the-art bounds for the task. Our proof combines techniques from combinatorial optimization and key results from theory of concentration inequalities. While previous results were tailored to analyse a specific estimator of the betweenness centrality, our result is general, since it applies to all available estimators of the betweenness centrality. Furthermore, we extend this result to obtain sharper *relative* deviation bounds from a random sample.
- We perform an extensive experimental evaluation (Section 4), showing that SILVAN improves the state-of-the-art by requiring a fraction of the sample sizes and running times to achieve a given approximation quality or, equivalently, sharper guarantees for the same amount of work. Our experimental evaluations shows that SILVAN's guarantees, provided by our theoretical analysis, hold with a true approximation error close to its probabilistic upper bound, confirming the sharpness of our analysis. For the extraction of the top-k betweenness centralities, our algorithm provides faster approximations, using less samples, and with fewer false positives.

In this short version of the paper we mainly focus on an high-level presentation and empirical evaluation of SILVAN. The full version of this work [22] includes all details, theoretical results and proofs, and the presentation of SILVAN-TopK, a variant of SILVAN to obtain a relative approximation of the k nodes with highest betweenness centrality.

2 Preliminaries

In this section we introduce the basic notions used in the remaining of the paper.

2.1 Graphs and Betweenness Centrality

Let $G = (V, E)$ be a graph. For ease of exposition, we focus on unweighted graphs, however our algorithms can be easily adapted to weighted graphs. For any pair (u, z) of different nodes ($u \neq z$), let σ_{uz} be the number of shortest paths between u and z, and let $\sigma_{uz}(v)$ be the number of shortest paths between u and z that *pass through* (i.e., contain) v, with $u \neq v \neq z$. Equivalently, v is *internal* to such shortest paths. The (normalized) *betweenness centrality* $b(v)$ of a node $v \in V$ is defined as

$$b(v) = \frac{1}{|V|(|V|-1)} \sum_{u \neq v \neq z} \frac{\sigma_{uz}(v)}{\sigma_{uz}}. \tag{1}$$

Intuitively, a node v has high betweenness centrality $b(v)$ if it is traversed by a large fraction of shortest paths of the graph G.

2.2 Rademacher Averages

Rademacher averages are a core concept in statistical learning theory [14] and in the study of empirical processes [7]. We now present the main notions and results used in our work, while additional details are given by [7, 34, 17]. Let $\mathcal{X}$ be a finite domain and consider a probability distribution γ over the elements of $\mathcal{X}$. Let $\mathcal{F}$ be a family of functions from $\mathcal{X}$ to $[0,1]$, and let $\mathcal{S} = \{\tau_1, \dots, \tau_m\}$ be a collection of m independent and identically distributed samples from $\mathcal{X}$ taken according to γ. Note that while we focus on functions $\in [0,1]$ for simplicity, all the results of this paper applies to functions in a bounded codomain $[a,b]$ by scaling and shifting. For each function $f \in \mathcal{F}$, define its average value over the sample $\mathcal{S}$ as $\mu_{\mathcal{S}}(f) = \frac{1}{m}\sum_{i=1}^{m} f(\tau_i)$ and its expectation, taken w.r.t. $\mathcal{S}$, as $\mu_\gamma(f) = \mathbb{E}_{\mathcal{S}}[\mu_{\mathcal{S}}(f)]$. Note that, by definition, $\mu_{\mathcal{S}}(f)$ is an *unbiased* estimator of $\mu_\gamma(f)$.

Given $\mathcal{S}$, we are interested in bounding the *supremum deviation* $\mathsf{D}(\mathcal{F}, \mathcal{S})$ of $\mu_{\mathcal{S}}(f)$ from $\mu_\gamma(f)$ among all $f \in \mathcal{F}$, that is

$$\mathsf{D}(\mathcal{F}, \mathcal{S}) = \sup_{f \in \mathcal{F}} |\mu_{\mathcal{S}}(f) - \mu_\gamma(f)|. \tag{2}$$

For the task of betweenness centrality approximation, different estimators can be defined with different notions of the domain $\mathcal{X}$, the family $\mathcal{F}$, and the sampling distribution γ. The simplest example is the `rk` estimator [24], where $\mathcal{X}$ is the set of shortest path of the graph, γ is the categorical distribution over $\mathcal{X}$ (where a shortest path $\pi \in \mathcal{X}$ from u to z has weight $(|V|(|V|-1)\sigma_{uz})^{-1}$), and the functions in $\mathcal{F}$ estimate the betweenness centrality $b(v)$ of the node $v \in V$ as the fraction of shortest paths of $\mathcal{S}$ that traverse v. More formally, $\mathcal{S}$ is a collection of m shortest paths taken independently at random from $\mathcal{X}$ according to γ (by choosing uniformly at random a pair u, z of nodes, and a shortest path from u to z), and the family $\mathcal{F} = \{f_v : v \in V\}$ is composed of indicator functions $f_v : \mathcal{X} \to \{0,1\}$ with $f_v(\tau) = \mathbb{1}[v \text{ is internal to } \tau]$, and $\mathbb{E}_\tau[f_v(\tau)] = b(v)$ for all $v \in V$. SILVAN employs a more refined estimator that we describe in more details in Section 3.2.

The *Empirical Rademacher Average* (ERA) $\hat{\mathsf{R}}(\mathcal{F}, \mathcal{S})$ of $\mathcal{F}$ on $\mathcal{S}$ is a key quantity to obtain a data-dependent upper bound to the supremum deviation $\mathsf{D}(\mathcal{F}, \mathcal{S})$. Let $\boldsymbol{\sigma} = \langle \boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_m \rangle$ be a collection of m i.i.d. Rademacher random variables (r.v.'s), each taking value in $\{-1, 1\}$ with equal probability. The ERA

$\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)$ of $\mathcal{F}$ on $\mathcal{S}$ is

$$\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{\sigma}_i f(\tau_i)\right]. \tag{3}$$

Computing the ERA $\hat{\mathsf{R}}\left(\mathcal{F},\mathcal{S}\right)$ is usually intractable, since there are 2^m possible assignments for $\boldsymbol{\sigma}$ and for each such assignment a supremum over the functions in $\mathcal{F}$ must be computed. A useful approach to obtain sharp probabilistic bounds on the ERA is given by Monte-Carlo estimation [2]. For $c \geq 1$, let $\boldsymbol{\sigma} \in \{-1,1\}^{c\times m}$ be a $c\times m$ matrix of i.i.d. Rademacher r.v.'s. The *c-trials Monte-Carlo Empirical Rademacher Average (c-MCERA)* $\hat{\mathsf{R}}^c_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma})$ of $\mathcal{F}$ on $\mathcal{S}$ using $\boldsymbol{\sigma}$ is:

$$\hat{\mathsf{R}}^c_m(\mathcal{F},\mathcal{S},\boldsymbol{\sigma}) = \frac{1}{c}\sum_{j=1}^{c}\sup_{f\in\mathcal{F}}\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{\sigma}_{j,i} f(\tau_i). \tag{4}$$

The c-MCERA allows to obtain sharp *data-dependent* probabilistic upper bounds to the supremum deviation, as they directly estimate the expected supremum deviation of sets of functions by taking into account their correlation. For this reason, they are often significantly more accurate than other methods [20], such as the ones based on often loose *deterministic upper bounds* to Rademacher averages (e.g., Massart's Lemma [16]), or other *distribution-free* notions of complexity, such as the VC-dimension. In general, the c-MCERA may be hard to compute, due to the supremums over $\mathcal{F}$ [2]. However, for the case of betweenness centralities, we show that all quantities relevant to the c-MCERA can be efficiently and incrementally updated by SILVAN as shortest paths are randomly sampled.

3 SILVAN: Efficient Progressive Estimation of Betweenness Centralities

In this section we introduce SILVAN (*eSt̲imatI̲ng betweenness centraL̲ities with progressiV̲e sA̲mpling and N̲on-uniform rademacher bounds*) and the techniques at its core.

We start, in Section 3.1, by presenting the empirical peeling technique and the related main technical results, which provide sharp data-dependent non-uniform approximation bounds supporting our algorithms. We then describe, in Section 3.2, our algorithm SILVAN that builds on such improved bounds to obtain an approximation within *additive* error ε of the betweenness centrality for all nodes via progressive sampling. In the full version of this work [22] we present novel improved bounds on the number of sufficient samples to achieve absolute approximations with high probability. These bounds are naturally combined with the progressive sampling scheme of SILVAN. The full version also includes SILVAN-TopK, the extension of SILVAN to obtain a relative approximation of the k nodes with highest betweenness centrality.

3.1 Non-uniform Bounds via Empirical Peeling

In this section we introduce *empirical peeling*, a new data-dependent scheme based on the c-MCERA to obtain sharp non-uniform bounds to the supremum deviation. The main idea behind empirical peeling is to *partition* the set of functions $\mathcal{F}$ in order to obtain the sharp bounds for different subsets of $\mathcal{F}$.

Classical concentration inequalities, such as Bernstein's and Bennet's [7], are well suited to control the deviation $\mathsf{D}(\{f\}, \mathcal{S})$ of a single function f, and to derive an approximation whose accuracy depends on its variance $Var(f)$. Instead, when *simultaneously* bounding the deviation of multiple functions belonging to a set of functions $\mathcal{F}$, the accuracy of the probabilistic bound on the supremum deviation $\mathsf{D}(\mathcal{F}, \mathcal{S})$ has a strong but natural dependence on the *maximum* variance $\sup_{f \in \mathcal{F}} Var(f)$. However, when the variances of the members of $\mathcal{F}$ are highly heterogenous, this leads to a significant loss of accuracy in the approximation of functions with variance much smaller than the maximum (i.e., we obtain a "blurred" approximation of functions f' with $Var(f') \ll \sup_{f \in \mathcal{F}} Var(f)$).

We propose an intuitive solution to achieve a higher granularity in the approximation: we partition $\mathcal{F}$ into $t \geq 1$ subsets $\{\mathcal{F}_j, j \in [1,t]\}$ with $\bigcup_j \mathcal{F}_j = \mathcal{F}$, such that functions with similar variance belong to the same subset $\mathcal{F}_j$; this allows to control the supremum deviations $\mathsf{D}(\mathcal{F}_j, \mathcal{S})$ for each $\mathcal{F}_j$ separately, exploiting the fact that the maximum variance is now computed on each subset $\mathcal{F}_j$ instead on the entire set $\mathcal{F}$. This idea leads to sharp *non-uniform bounds* (Theorem 1) that are locally valid for each subset $\mathcal{F}_j$ of $\mathcal{F}$, and it is the main motivation and intuition behind empirical peeling. Define the empirical wimpy variance $w_{\mathcal{F}_j}(\mathcal{S}) = \sup_{f \in \mathcal{F}_j} \frac{1}{m} \sum_{i=1}^{m} \left(f(\tau_i)\right)^2$. We state the following result, key to derive the guarantees of SILVAN.

Theorem 1. *Let $\mathcal{F} = \bigcup_{j=1}^{t} \mathcal{F}_j$ be a family of functions with codomain in $[0,1]$. Let $\mathcal{S}$ be a sample of size m taken i.i.d. from a distribution γ. Denote $\nu_{\mathcal{F}_j}$ such that $\sup_{f \in \mathcal{F}_j} Var(f) \leq \nu_{\mathcal{F}_j}$. For any $\delta \in (0,1)$, define*

$$\tilde{\mathsf{R}}_j \doteq \hat{\mathsf{R}}^c_m(\mathcal{F}_j, \mathcal{S}, \boldsymbol{\sigma}) + \sqrt{\frac{4 w_{\mathcal{F}_j}(\mathcal{S}) \ln\left(\frac{4t}{\delta}\right)}{cm}},$$

$$\mathsf{R}_j \doteq \tilde{\mathsf{R}}_j + \frac{\ln\left(\frac{4t}{\delta}\right)}{m} + \sqrt{\left(\frac{\ln\left(\frac{4t}{\delta}\right)}{m}\right)^2 + \frac{2\ln\left(\frac{4t}{\delta}\right)\tilde{\mathsf{R}}_j}{m}},$$

$$\varepsilon_{\mathcal{F}_j} \doteq 2\mathsf{R}_j + \sqrt{\frac{2\ln\left(\frac{4t}{\delta}\right)\left(\nu_{\mathcal{F}_j} + 4\mathsf{R}_j\right)}{m}} + \frac{\ln\left(\frac{4t}{\delta}\right)}{3m}. \tag{5}$$

With probability at least $1-\delta$ over the choice of $\mathcal{S}$ and $\boldsymbol{\sigma}$, it holds $\mathsf{D}(\mathcal{F}_j, \mathcal{S}) \leq \varepsilon_{\mathcal{F}_j}$ for all $j \in [1,t]$.

From Theorem 1 we observe that, since each $\nu_{\mathcal{F}_j}$ strongly affects $\varepsilon_{\mathcal{F}_j}$, as it typically dominates (5), partitioning $\mathcal{F}$ according to different stratifications of $\nu_{\mathcal{F}_j}$ is very beneficial to obtain sharp *non-uniform* bounds. We remark that recent works based on Monte Carlo Rademacher Averages [20, 9] used bounds that apply to the particular case $t = 1$ (without any partitioning of $\mathcal{F}$) to obtain a *uniform* variance-dependent bound that can be very loose for most functions as it ignores any heterogeneity of variances within $\mathcal{F}$ (see Thereom 3.2 of [20] and Theorem 3.1 of [9]).

We note that in many cases, appropriate values for variance upper bounds $\nu_{\mathcal{F}_j}$ are not known. The following result upper bounds every supremum variance $\sup_{f \in \mathcal{F}_j} Var(f)$ of all sets of functions $\{\mathcal{F}_j\}$ using the empirical wimpy variances $w_{\mathcal{F}}(\mathcal{S})$. This bound conveniently defines sharp data-dependent values of $\nu_{\mathcal{F}_j}$ that we plug in (5).

Proposition 1. *With probability at least* $1 - \delta$ *it holds, for all* $j \in [1, t]$,

$$\sup_{f \in \mathcal{F}_j} Var(f) \leq \nu_{\mathcal{F}_j} \doteq w_{\mathcal{F}_j}(\mathcal{S}) + \frac{\ln\left(\frac{t}{\delta}\right)}{m} + \sqrt{\left(\frac{\ln\left(\frac{t}{\delta}\right)}{m}\right)^2 + \frac{2 w_{\mathcal{F}_j}(\mathcal{S}) \ln\left(\frac{t}{\delta}\right)}{m}}. \quad (6)$$

Theorem 1 and Proposition 1 are easily combined by replacing $4/\delta$ by $5/\delta$ in Theorem 1, and $1/\delta$ by $5/\delta$ in (6); with this adjustment we obtain that both statements hold simultaneously with probability at least $1 - \delta$.

3.2 SILVAN

In this Section we give an high-level description of SILVAN, our algorithm, based on the contributions of Section 3.1, to compute rigorous approximations of the betweenness centrality of all nodes in a graph.

Sampling Shortest Paths SILVAN works by sampling shortest paths in G uniformly at random and using the fraction of shortest paths containing v as an unbiased estimator of its betweenness centrality $b(v)$. The first estimator following this idea was introduced by [24] (the `rk` estimator). The idea is to first samples two uniformly random nodes u, z, and then a uniformly distributed shortest path π between u and z. A more refined approach was proposed by [25] (the `ab` estimator), which considers *all* shortest paths between u and z instead of only one, approximating the betweenness centrality $b(v)$ as the *fraction* of such shortest paths passing through v. The `ab` estimator has been shown to have smaller variance and to provide higher quality approximations than the `rk` in practice [9]; this is because, intuitively, it updates estimations among all nodes involved in shortest paths between u and z, and thus, informally, provides "more information per sample". Computationally, the set Π_{uz} of shortest paths between u and z, required by both the `rk` and `ab` estimators, can be obtained in time $\mathcal{O}(|E|)$ using a (truncated) BFS, initialized from u and expanded until z is found. For the `rk` estimator, a faster approach based on a *balanced bidirection BFS* was proposed and analysed by [6]: they show that all information required to sample one shortest path between two vertices u and z can be obtained in time $\mathcal{O}(|E|^{\frac{1}{2}})$ with high probability on several random graph models, and experimentally on real-world instances. While this approach drastically speeds-up betweenness centrality approximations via the `rk` estimator [6], an analogous extension of this technique to the `ab` estimator is desirable but currently lacking.

Our sampling algorithm extends the balanced bidirection BFS to the `ab` estimator; this allows to combine superior statistical properties of `ab` with the much faster balanced bidirection BFS enjoyed by `rk`. Our main idea is that, once the set of all shortest paths $\mathit{\Pi}_{uz}$ between u and z is *implicitly* computed by the two BFSs, then it is very efficient to sample multiple shortest paths uniformly at random from $\mathit{\Pi}_{uz}$ (while [6] only sampled one shortest path).

SILVAN samples shortest paths with the following procedure:

1. sample two uniformly random nodes u, z;
2. performs a balanced bidirection BFS starting from u and z, until the two BFSs “meet”;
3. sample uniformly at random $\lceil \alpha \sigma_{uz} \rceil$ shortest paths from the set $\mathit{\Pi}_{uz}$ of shortest paths between u and z, where $\sigma_{uz} = |\mathit{\Pi}_{uz}|$ is the number of shortest paths between u and z and $\alpha \geq 1$ a positive constant.

It is easy to see that the expected fraction of shortest paths sampled using this procedure containing v is equal to the betweenness centrality $b(v)$ of v. In particular, for each node $v \in V$ and a bag of shortest paths τ obtained from this sampling procedure, define the function $f_v(\tau)$, with $f_v(\tau) = |\tau|^{-1} \sum_{\pi \in \tau} \mathbb{1}\left[v \in \pi\right]$ where $\mathbb{1}\left[v \in \pi\right] = 1$ if v is internal to the shortest path $\pi \in \tau$, 0 otherwise. Consequently, the set of functions we use for betweenness centrality approximation contains all f_v with $v \in V$, so that $\mathcal{F} = \{f_v, v \in V\}$. By considering a sample $\mathcal{S}$ of size m taken as described above, we define the estimate $\tilde{b}(v)$ of the betweenness centrality $b(v)$ of v as $\tilde{b}(v) = \mu_{\mathcal{S}}(f_v) = \frac{1}{m} \sum_{\tau \in \mathcal{S}} f_v(\tau)$. We have that $\tilde{b}(v)$ is an unbiased estimator of $\mu_\gamma(f_v) = b(v)$, so that $\mathbb{E}_{\mathcal{S}}[\tilde{b}(v)] = b(v)$. Regarding α, from standard Poisson approximation to the balls and bins model [17], we obtain that the expected fraction of shortest paths that are not sampled from the set $\mathit{\Pi}_{uz}$ in step (3) is $\sigma_{uz}(1 - 1/\sigma_{uz})^{\alpha \sigma_{uz}} \approx e^{-\alpha}$. Consequently, to ensure that the set of sampled shortest paths well represents $\mathit{\Pi}_{uz}$, we set α to $\ln \frac{1}{\lambda}$ where λ is a small value (e.g., in practice we use $\lambda = 0.1$).

SILVAN algorithm We now describe our algorithm SILVAN to compute an accurate approximation of the betweenness centrality. The goal of SILVAN is to achieve an ε-*approximation* (or ε absolute approximation) of the set $BC(G) = \{b(v) : v \in V\}$, defined as follows.

Definition 1. *A set* $\tilde{BC}(G) = \{\tilde{b}(v) : v \in V\}$ *is an* ε-approximation *of* $BC(G) = \{b(v) : v \in V\}$ *if it holds, for all* $v \in V$*, that* $|b(v) - \tilde{b}(v)| \leq \varepsilon$.

Algorithm 1 describes the algorithm SILVAN to compute an ε-approximation of $BC(G)$ by employing the techniques introduced in Section 3.1. SILVAN can be logically divided into two phases: in the first phase (lines 1-4), SILVAN generates a sample $\mathcal{S}'$ that is used for empirical peeling (Section 3.1) to partition $\mathcal{F}$

into t subsets $\{\mathcal{F}_j, j \in [1,t]\}$. The second phase (lines 5-15) describes the main operations of the algorithm to approximate the betweenness centrality.

The second phase of SILVAN is based on a *progressive sampling* approach. At a high level, the algorithm works in iterations, and in iteration i SILVAN extracts an approximation $\tilde{b}(v)$ of the values $b(v)$ for all $v \in V$ from a sample $\mathcal{S}_i$, which is a collection of $m_i = |\mathcal{S}_i|$ randomly sampled bags of shortest paths. The progressive sampling scheme considers samples sizes $\{m_i\}$ that form an increasing sequence, following a suitable *sampling schedule.* At the end of each iteration, SILVAN checks whether a suitable *stopping condition* is satisfied. This stopping condition is based on estimating the c-MCERA of each partition $\mathcal{F}_j$ on the sample $\mathcal{S}_i$, and obtaining bounds to the supremum deviation $\mathsf{D}(\mathcal{F}_j, \mathcal{S})$ for each $\mathcal{F}_j$ (via the empirical peeling technique of Theorem 1). When all the deviations are small enough (i.e., all are $\leq \varepsilon$), the stopping condition is satisfied and the algorithm reports the achieved approximation. It is important that the stopping condition is satisfied as soon as possible, as each sample is expensive to compute, in particular for large graphs.

We leave to the full version (Section 4.2.2 of [22]) a detailed description of the operations done by Algorithm 1. We prove that its output is an accurate approximation of the betweenness centrality.

Proposition 2. *With probability $\geq 1 - \delta$, the output $\tilde{b}$ of* SILVAN *is a ε-approximation of $BC(G)$.*

We now describe a simple but effective criteria to partition $\mathcal{F}$, implementing the `empiricalPeeling` method, used in Algorithm 1. First, we denote with $\tilde{w}_v$ the estimated wimpy variance of the function f_v on sample $\mathcal{S}'$ as $\tilde{w}_v = \frac{1}{|\mathcal{S}'|} \sum_{\tau \in \mathcal{S}'} \left(f_v\left(\tau\right)\right)^2$. We assign each function f_v for each node $v \in V$ to the set $\mathcal{F}_j$ with index $j = \lceil \log_a(\min\{\tilde{w}_v^{-1}, |\mathcal{S}'|\}) \rceil$ for a constant $a > 1$. Intuitively, this allows to split $\mathcal{F}$ into (at most) $t = \lceil \log_a(|\mathcal{S}'|) \rceil$ partitions, such that each set $\mathcal{F}_j$ groups functions with variances in $[1/a^{j+1}, 1/a^j]$, therefore within a multiplicative factor a. Our main intuition is that the empirical wimpy variances $w_{\mathcal{F}_j}(\mathcal{S})$ control the accuracy of the bounds on the supremum deviations $\mathsf{D}(\mathcal{F}_j, \mathcal{S})$ (as it estimates $\nu_{\mathcal{F}_j}$ in Theorem 1); this partitioning scheme fully exploits the non-uniform variance-dependent bounds at the core of SILVAN since the empirical wimpy variances $w_{\mathcal{F}_j}(\mathcal{S})$ are approximated by $w_{\mathcal{F}_j}(\mathcal{S}')$ and are $w_{\mathcal{F}_j}(\mathcal{S}') \leq 1/a^j$, which decrease exponentially with j.

4 Experiments

We implemented SILVAN and tested it on several real-world graphs. In our experimental evaluations we assess the effectiveness of the progressive sampling approach of SILVAN to approximate the betweenness centrality of all nodes.

Experimental Setup We implemented SILVAN in `C++`. Our implementation of SILVAN, with automated scripts to reproduce all experiments, is available online.[1] We compare SILVAN with KADABRA [6][2], that has been shown to uniformly and significantly outperform previous methods, and with BAVARIAN [9][3], the most recent method for betweenness centrality approximation. When referring to BAVARIAN, we consider its variant based on progressive sampling (denoted BAVARIAN-P, see Alg. 2 and Sect. 5.2 of [9]) which addresses the same problem solved by SILVAN and KADABRA, and we tested it using all different estimators for the betweenness centrality described in [9] (called rk, ab, and bp). In this short version we only show results for the ab estimators, while the complete comparison is available in the extended version [22]. All the code was compiled with GCC 8 and run on a machine with 2.30 GHz Intel Xeon CPU, 512 GB of RAM, on Ubuntu 20.04, with a total of 64 cores. All experiments were performed using multithreading on all threads.

Graphs. We tested SILVAN on 7 undirected and 11 directed real-world graphs from SNAP[4] and KONECT[5], most of them previously analysed by KADABRA [6] and other previous methods [24, 27, 9]. The characteristics of the graphs are described in detail in Table 1 (in the Appendix).

For every graph, we ran all algorithms to compute an ε-approximation with parameter $\varepsilon \in \{0.01, 0.005, 0.0025, 0.001, 0.0005\}$, chosen to have comparable magnitude to the betweenness centrality of the most central nodes (i.e., see col. $\hat{\xi}$ of Table 1); this is required to compute meaningful approximations (i.e., an ε absolute approximation is useless when the centralities of the most central nodes are much smaller than ε). We fix $\delta = 0.05$, and use $c = 25$ Monte Carlo Rademacher vectors for SILVAN and BAVARIAN (note that $c = k$ in [9]).[6] We ran all algorithms 10 times and report averages ± stds. We limit the execution time of each run to 6 hours; we terminate the algorithm when exceeding this threshold.

For the empirical peeling scheme of SILVAN, we sample $m' = \log(1/\delta)/\varepsilon$ shortest paths to generate $\mathcal{S}'$, always a very small fraction of the overall samples analysed by SILVAN. Regarding the sampling schedule followed in the second phase, we increase the sample size m_i with a geometric progression, such that $m_{i+1} = \theta \cdot m_i$, with $\theta = 1.2$. The empiricalPeeling procedure of SILVAN follows the scheme described at the end of Section 3.2 using $a = 2$. For the progressive sampling schedule of BAVARIAN, we use the same geometric progression parameter $\theta = 1.2$ of SILVAN.

Figure 1 shows the results for this set of experiments comparing SILVAN to KADABRA, while Figure 3 shows the results comparing SILVAN to BAVARIAN for the estimator ab (Figures in the Appendix).

[1] https://github.com/VandinLab/SILVAN

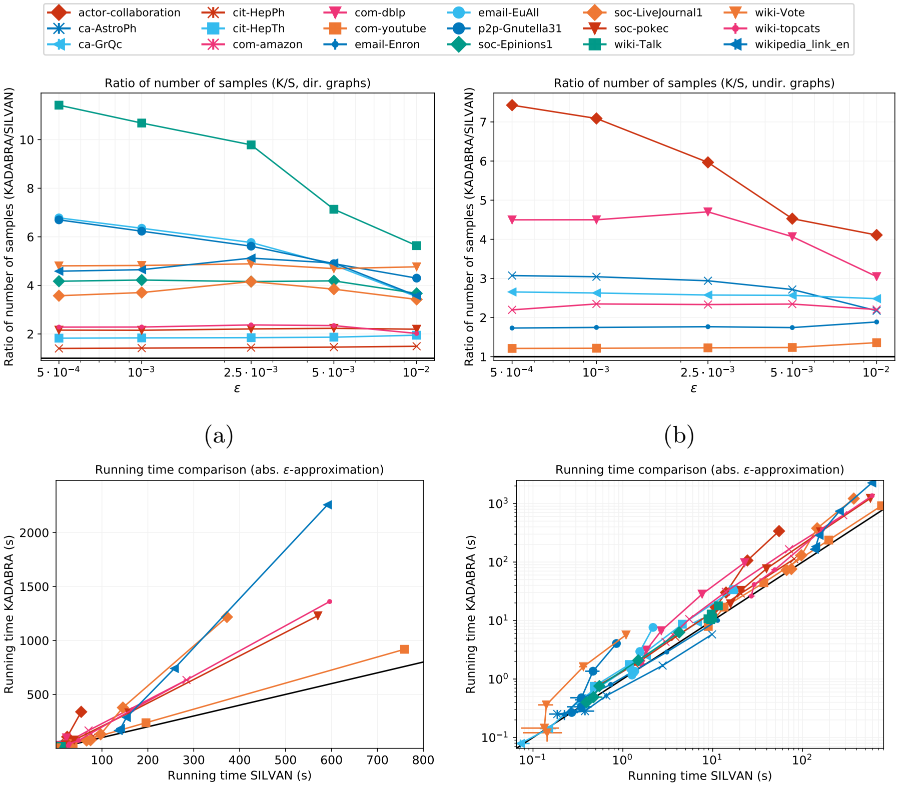
[2] https://github.com/natema/kadabra

[3] https://github.com/acdmammoths/Bavarian-code

[4] http://snap.stanford.edu/data/index.html

[5] http://konect.cc/networks/

[6] We follow [20] and [9], that have shown that sharp bounds are obtained even with a low number of Monte Carlo trials, and that there are minimal improvements using $c > 30$.



(a) (b)

(c) (d)

Fig. 1: Comparison between the performance of `KADABRA` and SILVAN for obtaining absolute approximations. (a): ratios of the number of samples required by `KADABRA` and the number of samples required by SILVAN for directed graphs (black line drawn at $y = 1$). (b): as (a) for undirected graphs. (c): comparison of the running times of `KADABRA` (y axis) and SILVAN (x axis) for all graphs (black line drawn at $y = x$). (d): as (c) with axes in log scale.

Sample sizes In Figures 1 (a) and (b) we show the ratios between the number of samples required by KADABRA and SILVAN to converge (we sum the number of samples of both phases, for both algorithms) for directed and undirected graphs. We can see that the number of samples needed by SILVAN is always smaller than KADABRA, by at least 20%; for 14 out of 18 graphs, SILVAN finished after processing *less than half* of the samples considered by KADABRA, and may require up to an order of magnitude less samples. By inspecting the graphs' statistics (Table 1), the largest improvements are obtained for graphs with smallest $\sup_{v \in V} b(v) \leq \hat{\xi}$ (a simple bound to the maximum variance). In fact, the number of samples required by SILVAN (Figure 2 (a), in the Appendix) varies significantly among graphs, with strong dependence on $\hat{\xi}$. Notice that $\sup_{v \in V} b(v)$ upper bounds the maximum variance $\sup_{v \in V} Var(f_v)$. A potential cause of the gap between SILVAN and KADABRA may depend on the use of the VC-dimension based bound in the adaptive sampling analysis of KADABRA; such bound is indeed required for its correctness, but it is agnostic to any property of the underlying graph (apart from the vertex diameter) and thus results in overly conservative guarantees in such cases. This confirms the significance of SILVAN's sharp *variance-adaptive bounds*. In addition, the fact that SILVAN obtains simultaneous and non-uniform data-dependent approximations for *sets of nodes*, exploting correlations among nodes through the use of the c-MCERA, leads to refined guarantees.

We now compare SILVAN with BAVARIAN in terms of sample sizes. We remark that the plots for sample sizes only show the results for cases in which BAVARIAN terminates in reasonable time (i.e, in less than 6 hours), while figures for running times show a lower bound for such cases. From Figures 3 (a) and (b) (in the Appendix), we can see that SILVAN always requires a fraction of the samples needed by BAVARIAN: at most *half* of the samples for all graphs, up to 1/4 of the samples.

Overall, SILVAN obtains high-quality approximations at a fraction of the samples required by state-of-the-art methods; this highlights the significance of SILVAN's non-uniform approximation approach via empirical peeling and its novel improved bounds on the number of sufficient samples (presented in the full version [22]).

Running times We now discuss how the reduction in the number of samples impacts the overall running times. We observed that, generally, the running time roughly increases linearly with the sample size (Figure 2 (b) shows that the relationship between the sample sizes and the running times of SILVAN is essentialy linear). In fact, the time spent on sampling shortest paths is usually the dominating cost of the algorithms.

In Figure 1 (c) we compare the running times of SILVAN (x axis) and KADABRA (y axis). While for smaller graphs both SILVAN and KADABRA terminate very quickly (e.g., in < 10 seconds), for the largest and most demanding graphs the reduction on the number of samples achieved by SILVAN has a sensible and significant impact on the running times, as clearly shown in Figure 1 (c). For instance, SILVAN analyses the most demanding graph (`wikipedia-link-en`) in less than 1/3 of the time required by `KADABRA` when $\varepsilon \leq 10^{-3}$. This is a consequence of significantly reducing the required samples, and also reflects the capability of SILVAN to compute the c-MCERA *incrementally* as shortest paths are sampled, incurring in a negligible computational overhead.

In Figure 3 (c)-(d) we compare the running times of SILVAN (x axis) and `BAVARIAN` using the `ab` estimator (y axis). Note that we report a lower bound to the running time of `BAVARIAN` when exceeding 6 hours ($= 2.16 \cdot 10^4$ seconds); `BAVARIAN` exceeded this threshold on most large graphs and for smaller values of ε, while SILVAN never required more than 17 minutes ($= 10^3$ seconds). Overall, we observed SILVAN to be at least one order of magnitude faster than `BAVARIAN`, up to 3 orders of magnitude. We observed very similar results for other estimators. SILVAN's improvements are due to both the significant reduction in the number of samples (as discussed previously) thanks to its non-uniform approximation scheme, and from the fact that SILVAN leverages a more efficient algorithm for sampling shortest paths, based on the balanced bidirectional BFS, drastically reducing the computational requirement for the task.

We conclude that SILVAN requires much fewer resources to obtain rigorous approximations of the betweenness centrality of all nodes of the same quality, or, equivalently, sharper guarantees for the same amount of work.

5 Conclusions

We introduced SILVAN, a novel progressive sampling algorithm to estimate the betweenness centrality of all nodes in a graph. SILVAN relies on new bounds on supremum deviation of functions, based on the c-MCERA and non-uniform approximation scheme via empirical peeling. We present variants of SILVAN to obtain additive approximations, and relative approximations for the top-k betweenness centrality. Our experimental results show that SILVAN significantly outperforms state-of-the-art approaches for approximating betweenness centrality with the same guarantees.

There are multiple interesting directions for future work. While in this work we considered various approximations of the betweenness centrality in a static setting, recent works considered extending the problem to dynamic [4, 3, 13], temporal [30], and uncertain graphs [29], or different types of centralities [15, 19], all settings in which we believe the ideas behind our algorithm SILVAN could lead to improved approximations.

Furthermore, the empirical peeling scheme we introduced in this work is general: it can be applied to sets of functions with arbitrary domains, so it can potentially benefit randomized approximation algorithms in other settings, such as interesting [28, 31, 33] and significant pattern mining [21], and sequential hypothesis testing [10].

Acknowledgments This work was supported, in part, by MIUR of Italy, under PRIN Project n. 20174LF3T8 AHeAD and grant L. 232 (Dipartimenti di Eccellenza), and by the University of Padova under project "SID 2020: RATED-X".

References

1. Anthonisse, J.M.: The rush in a directed graph. Stichting Mathematisch Centrum. Mathematische Besliskunde (BN 9/71) (1971)
2. Bartlett, P.L., Mendelson, S.: Rademacher and Gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research **3**(Nov), 463–482 (2002)
3. Bergamini, E., Meyerhenke, H.: Fully-dynamic approximation of betweenness centrality. In: Algorithms-ESA 2015, pp. 155–166. Springer (2015)
4. Bergamini, E., Meyerhenke, H., Staudt, C.L.: Approximating betweenness centrality in large evolving networks. In: 2015 Proceedings of the Seventeenth Workshop on Algorithm Engineering and Experiments (ALENEX). pp. 133–146. SIAM (2014)
5. Borassi, M., Crescenzi, P., Habib, M.: Into the square: On the complexity of some quadratic-time solvable problems. Electronic Notes in Theoretical Computer Science **322**, 51–67 (2016)
6. Borassi, M., Natale, E.: Kadabra is an adaptive algorithm for betweenness via random approximation. Journal of Experimental Algorithmics (JEA) **24**, 1–35 (2019)
7. Boucheron, S., Lugosi, G., Massart, P.: Concentration inequalities: A nonasymptotic theory of independence. Oxford university press (2013)
8. Brandes, U.: A faster algorithm for betweenness centrality. Journal of mathematical sociology **25**(2), 163–177 (2001)
9. Cousins, C., Wohlgemuth, C., Riondato, M.: Bavarian: Betweenness centrality approximation with variance-aware rademacher averages. ACM Trans. Knowl. Discov. Data **17**(6) (mar 2023). https://doi.org/10.1145/3577021
10. De Stefani, L., Upfal, E.: A rademacher complexity based method for controlling power and confidence level in adaptive statistical analysis. In: 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA). pp. 71–80. IEEE (2019)
11. Erdős, D., Ishakian, V., Bestavros, A., Terzi, E.: A divide-and-conquer algorithm for betweenness centrality. In: Proceedings of the 2015 SIAM International Conference on Data Mining. pp. 433–441. SIAM (2015)
12. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry pp. 35–41 (1977)
13. Hayashi, T., Akiba, T., Yoshida, Y.: Fully dynamic betweenness centrality maintenance on massive networks. Proceedings of the VLDB Endowment **9**(2), 48–59 (2015)
14. Koltchinskii, V., Panchenko, D.: Rademacher processes and bounding the risk of function learning. In: High dimensional probability II, pp. 443–457. Springer (2000)
15. de Lima, A.M., da Silva, M.V., Vignatti, A.L.: Estimating the percolation centrality of large networks through pseudo-dimension theory. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1839–1847 (2020)
16. Massart, P.: Some applications of concentration inequalities to statistics. Annales de la Faculté des sciences de Toulouse: Mathématiques **9**(2), 245–303 (2000)
17. Mitzenmacher, M., Upfal, E.: Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis. Cambridge university press (2017)
18. Newman, M.: Networks. Oxford university press (2018)
19. Pellegrina, L.: Efficient centrality maximization with rademacher averages. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data

Mining. p. 1872–1884. KDD '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3580305.3599325

20. Pellegrina, L., Cousins, C., Vandin, F., Riondato, M.: Mcrapper: Monte-carlo rademacher averages for poset families and approximate pattern mining. ACM Transactions on Knowledge Discovery from Data (TKDD) **16**, 1 – 29 (2022)
21. Pellegrina, L., Riondato, M., Vandin, F.: SPuManTE: Significant pattern mining with unconditional testing. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1528–1538. KDD '19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3292500.3330978
22. Pellegrina, L., Vandin, F.: SILVAN: Estimating Betweenness Centralities with Progressive Sampling and Non-uniform Rademacher Bounds. arXiv preprint arXiv:2106.03462 (2021)
23. Pollard, D.: Convergence of stochastic processes. Springer Science & Business Media (2012)
24. Riondato, M., Kornaropoulos, E.M.: Fast approximation of betweenness centrality through sampling. Data Mining and Knowledge Discovery **30**(2), 438–475 (2016)
25. Riondato, M., Upfal, E.: Efficient discovery of association rules and frequent itemsets through sampling with tight performance guarantees. ACM Trans. Knowl. Disc. from Data **8**(4), 20 (2014). https://doi.org/10.1145/2629586
26. Riondato, M., Upfal, E.: Mining frequent itemsets through progressive sampling with Rademacher averages. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1005–1014. KDD '15, ACM (2015)
27. Riondato, M., Upfal, E.: ABRA: Approximating betweenness centrality in static and dynamic graphs with Rademacher averages. ACM Trans. Knowl. Disc. from Data **12**(5), 61 (2018)
28. Riondato, M., Vandin, F.: MiSoSouP: Mining interesting subgroups with sampling and pseudodimension. In: Proc. 24th ACM SIGKDD Int. Conf. Knowl. Disc. and Data Mining. pp. 2130–2139. KDD '18, ACM (2018)
29. Saha, A., Brokkelkamp, R., Velaj, Y., Khan, A., Bonchi, F.: Shortest paths and centrality in uncertain networks. Proceedings of the VLDB Endowment **14**(7), 1188–1201 (2021)
30. Santoro, D., Sarpe, I.: Onbra: Rigorous estimation of the temporal betweenness centrality in temporal networks. Proceedings of the ACM Web Conference 2022 (2022)
31. Santoro, D., Tonon, A., Vandin, F.: Mining sequential patterns with vc-dimension and rademacher complexity. Algorithms **13**(5), 123 (2020)
32. Sariyüce, A.E., Saule, E., Kaya, K., Çatalyürek, Ü.V.: Shattering and compressing networks for betweenness centrality. In: Proceedings of the 2013 SIAM International Conference on Data Mining. pp. 686–694. SIAM (2013)
33. Sarpe, I., Vandin, F.: oden: Simultaneous approximation of multiple motif counts in large temporal networks. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 1568–1577 (2021)
34. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press (2014)
35. Vapnik, V.N., Chervonenkis, A.Y.: On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability & Its Applications **16**(2), 264 (1971). https://doi.org/10.1137/1116025

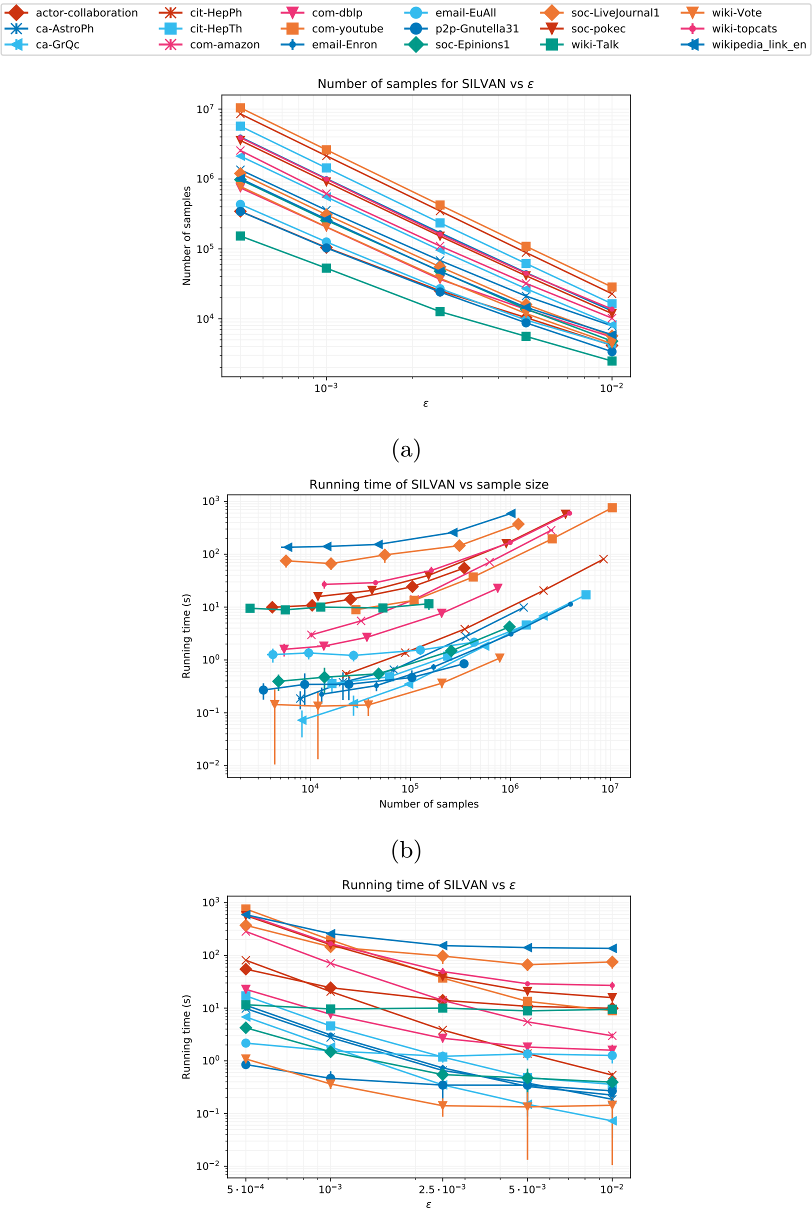
A Appendix

Algorithm 1: SILVAN

```
Input: Graph $G = (V, E)$; $c, m' \geq 1$; $\varepsilon, \delta \in (0, 1)$.
Output: $\varepsilon$-approximation of BC(G) with probability $\geq 1 - \delta$
$\mathcal{S}' \leftarrow$ sampleSPs($m'$);
$\{\mathcal{F}_j, j \in [1, t]\} \leftarrow$ empiricalPeeling($\mathcal{F}, \mathcal{S}'$);
$\hat{m} \leftarrow$ sufficientSamples($\mathcal{F}, \mathcal{S}', \delta/2$);
$\{m_i\}, \{\delta_i\} \leftarrow$ samplingSchedule($\mathcal{F}, \mathcal{S}'$);
forall $j \in [1, t]$ do  $\varepsilon_{\mathcal{F}_j} \leftarrow 1$;
$i \leftarrow 0$; $\mathcal{S}_0 \leftarrow \emptyset$; $\boldsymbol{\sigma} \leftarrow$ empty matrix;
while not stoppingCond($\varepsilon, \{\varepsilon_{\mathcal{F}_j}\}, \hat{m}, m_i$) do
    $i \leftarrow i + 1$; $d_m \leftarrow m_i - m_{i-1}$;
    $\mathcal{S}_i \leftarrow \mathcal{S}_{i-1} \cup$ sampleSPs($d_m$);
    $\boldsymbol{\sigma} \leftarrow$ add columns {sampleRrvs($d_m, c$)} to $\boldsymbol{\sigma}$;
    $\tilde{b}, \tilde{r}, \{\nu_{\mathcal{F}_j}\} \leftarrow$ updateEstimates($\mathcal{S}_i, \boldsymbol{\sigma}, \{\mathcal{F}_j\}$);
    forall $j \in [1, t]$ do
        $\hat{\mathsf{R}}^c_{m_i}(\mathcal{F}_j, \mathcal{S}_i, \boldsymbol{\sigma}) \leftarrow \frac{1}{c} \sum_{x=1}^{c} \max_{v \in V, f_v \in \mathcal{F}_j} \{\tilde{r}(v, x)\}$;
        $\varepsilon_{\mathcal{F}_j} \leftarrow$ epsBound($\hat{\mathsf{R}}^c_{m_i}(\mathcal{F}_j, \mathcal{S}_i, \boldsymbol{\sigma}), \nu_{\mathcal{F}_j}, \delta_i$);
return $\tilde{b}$;
```

Table 1: Statistics of undirected (top section) and directed (bottom section) graphs. D is the vertex diameter, $\hat{\rho}$ is an upper bound of the average shortest path lenth ρ, and $\hat{\xi}$ is an upper bound of $\max_v\{b(v)\}$.

G	$\|V\|$	$\|E\|$	D	$\hat{\rho}$	$\hat{\xi}$
actor-collaboration	3.82e5	3.31e7	13	2.87	0.0090
ca-AstroPh	1.87e4	1.98e5	14	3.20	0.0285
ca-GrQc	5.24e3	1.44e4	17	3.51	0.0450
com-amazon	3.34e5	9.25e5	44	11.97	0.0450
com-dblp	3.17e5	1.04e6	21	6.27	0.0162
com-youtube	1.13e6	2.98e6	20	4.68	0.2573
email-Enron	3.66e4	1.83e5	11	2.78	0.0749
cit-HepPh	3.45e4	4.21e5	12	5.35	0.1817
cit-HepTh	2.77e4	3.52e5	13	2.10	0.1237
email-EuAll	2.65e5	4.20e5	14	0.56	0.0121
p2p-Gnutella31	6.25e4	1.47e5	11	2.16	0.0071
soc-Epinions1	7.58e4	5.08e5	14	2.11	0.0210
soc-LiveJournal1	4.84e6	6.90e7	16	4.58	0.0270
soc-pokec	1.63e6	3.06e7	16	3.94	0.0802
wiki-Talk	2.39e6	5.02e6	9	0.26	0.0037
wiki-topcats	1.79e6	2.85e7	9	5.87	0.0985
wiki-Vote	7.11e3	1.03e5	7	0.66	0.0240
wikipedia-link-en	1.35e7	4.37e8	10	3.21	0.0300

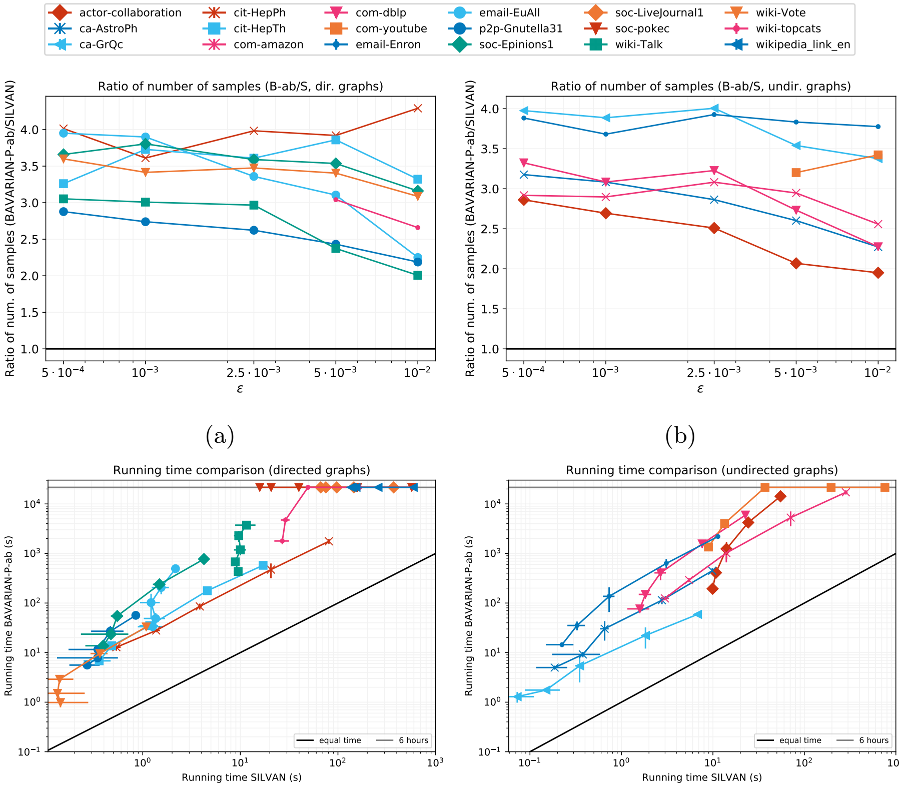


(a)

(b)

(c)

Fig. 2: Resources required by SILVAN for obtaining absolute ε approximations. (a): Number of samples for SILVAN vs. ε. (b): Running times of SILVAN vs. Number of samples. (c): Running times of SILVAN vs. ε.



(a)

(b)

(c)

(d)

Fig. 3: Comparison between the performance of BAVARIAN (ab estimator) and SILVAN for obtaining absolute approximations. (a): ratios of the number of samples required by BAVARIAN and the number of samples required by SILVAN for directed graphs (black line drawn at $y = 1$). (b): as (a) for undirected graphs. (c): comparison of the running times of BAVARIAN (y axis) and SILVAN (x axis) for directed graphs (axes in logarithmic scale) (black line drawn at $y = x$). (d): as (c) for undirected graphs.