Perfect Sampling from Pairwise Comparisons

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work, we study how to efficiently obtain perfect samples from a discrete distribution $\mathcal{D}$ given access only to pairwise comparisons of elements of its support. Specifically, we assume access to samples (x, S), where S is drawn from a distribution over sets $\mathcal{Q}$ (indicating the elements being compared), and x is drawn from the conditional distribution $\mathcal{D}_S$ (indicating the winner of the comparison) and aim to output a clean sample y distributed according to $\mathcal{D}$. We mainly focus on the case of pairwise comparisons where all sets S have size 2. We design a Markov chain whose stationary distribution coincides with $\mathcal{D}$ and give an algorithm to obtain exact samples using the technique of Coupling from the Past. However, the sample complexity of this algorithm depends on the structure of the distribution $\mathcal{D}$ and can be even exponential in the support of $\mathcal{D}$ in many natural scenarios. Our main contribution is to provide an efficient exact sampling algorithm whose complexity does not depend on the structure of $\mathcal{D}$. To this end, we give a parametric Markov chain that mixes significantly faster given a good approximation to the stationary distribution. We can obtain such an approximation using an efficient learning from pairwise comparisons algorithm (Shah et al., JMLR 17, 2016). Our technique for speeding up sampling from a Markov chain whose stationary distribution is approximately known is simple, general and possibly of independent interest.

1 Introduction

Machine Learning questions dealing with pairwise comparisons (PCs) constitute a fundamental and broad research area with multiple real-world applications [SBB$^+$16, Kaz11, LOAF12, SBGW16, FH10, WJJ13]. For instance, the preference of a consumer to choose one product over another constitutes a pairwise comparison between the two products. The two bedrocks of modern ML applications are inference/learning and sampling. The area of learning from pairwise comparisons has a vast amount of work [RA14, SBGW16, DKNS01, NOS12, Hun04, JLYY11, APA18, LSR21, HOX14, SBB$^+$16, VY16, NOS17, VYZ20] . In this problem, there exist n alternatives and an unknown weight vector $\mathcal{D} \in \mathbb{R}^n$ where $\mathcal{D}(i)$ is the weight of the i-th element with $\mathcal{D}(i) \geq 0$ and $\sum_{i\in[n]} \mathcal{D}(i) = 1$, i.e., $\mathcal{D}$ is a probability distribution over $[n]$. Roughly speaking, the learner observes pairwise comparisons (or more generally k-wise comparisons) of the form $(w, \{u, v\}) \in [n] \times [n]^2$ that compare the alternatives $u \neq v$ and w equals either u or v indicating the winner of the comparison. The most fundamental distribution over pairwise comparisons is the BTL model [BT52, Luc59] where the probability that the learner observes $w = u$ when the items u, v are compared is $\mathcal{D}(u)/(\mathcal{D}(u) + \mathcal{D}(v))$. The learner's goal is mainly to estimate the underlying distribution $\mathcal{D}$ using a small number of pairwise comparisons.

In this work, we focus on questions concerning sampling, the second bedrock of modern ML. In fact, we will deal with a fundamental area of the sampling literature, namely *perfect sampling*. In contrast to approximate sampling, perfect (or exact) sampling from a probabilistic model requires an exact sample from this model. Problems in this vein of research lie in the intersection of computer science

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

and probability theory and have been extensively studied [PW96, PW98, Hub04, Hub16, FGY19, JSS21, AJ21, GJL19, BC20, HSW21]. Our work poses questions concerning perfect sampling in the area of pairwise comparisons. We ask the following simple yet non-trivial question: Is it possible to obtain a perfect sample x distributed as in $\mathcal{D}$ by observing only pairwise comparisons from $\mathcal{D}$?

Apart from the algorithmic and mathematical challenge of generating exact samples, perfect simulation has also practical motivation, noticed in various previous works [Hub16, JSS21, AJ21]. Crucially, its value does not stem from the fact that the output of the sampling distribution is perfect; usually, one can come up with a Markov chain whose evolution rapidly converges to the stationary measure. Hence, the distance between the (approximate) sampling distribution from the desired one can become small efficiently. This approximate sampling approach has a clear drawback: there is no termination criterion, i.e., the Markov chain has to be run for sufficiently long time and this requires an a priori known bound on the chain's mixing time, which may be much larger that necessary. On the contrary, perfect sampling algorithms come with a stopping rule and this termination condition can be attained well ahead of the worst-case analysis time bounds in practice.

Second, the question that we pose appears to have links with the important literature of truncated statistics [Gal97, DGTZ18]. Truncation is a common biasing mechanism under which samples from an underlying distribution $\mathcal{D}$ over $\mathbb{X}$ are only observed if they fall in a given set $S \subseteq \mathbb{X}$, i.e., one only sees samples from the conditional distribution $\mathcal{D}_S$. There exists a vast amount of applications that involve data censoring (that go back to at least [Gal97]) where one does not have direct sample access to $\mathcal{D}$. Recent work in this literature [FKT20] asked when it is possible to obtain perfect samples from some specific truncated discrete distributions (namely, truncated Boolean product distributions). The area of pairwise comparisons can be identified as an extension of the truncated setting where the truncation set can change (i.e., each subset of alternatives corresponds to some truncation set); while perfect sampling has been a subject of interest in truncated statistics, the neighboring area of pairwise comparisons lacks of efficient algorithms for this important question.

Problem Formulation. Let us define our setting formally (which we call Local Sampling Scheme). This setting is standard (without this terminology) and can be found e.g., at [RA14].

Definition 1 ([RA14])**.** *Let $\mathcal{Z}$ be a finite discrete domain. Consider a target distribution $\mathcal{D}$ supported on $\mathcal{Z}$ and a distribution $\mathcal{Q}$ supported on subsets of $\mathcal{Z}$. The sample oracle* $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ *is called* Local Sampling Scheme (LSS) *and each time it is invoked, it returns an example (x, S) such that: (i) the set $S \subseteq \mathcal{Z}$ is drawn from $\mathcal{Q}$ and (ii) x is drawn from the conditional distribution $\mathcal{D}_S$, where $\mathcal{D}_S(x) = \mathcal{D}(x)\mathbf{1}\{x \in S\}/\mathcal{D}(S)$ and $\mathcal{D}(S) = \sum_{x\in S} \mathcal{D}(x)$.*

For simplicity[1], we mostly focus on the case where $\mathcal{Q}$ is supported on a subset of $\mathcal{Z} \times \mathcal{Z}$, and often refer to $\mathcal{Q}$ as the *pair distribution*. Then, $\mathcal{Q}$ naturally induces an (edge weighted) undirected graph $G_{\mathcal{Q}}$ on $\mathcal{Z}$, where $\{x, y\} \in E(G_{\mathcal{Q}})$, if $\{x, y\}$ is supported on $\mathcal{Q}$, and has weight equal to the probability $\mathcal{Q}(x, y)$ that $\{x, y\}$ is drawn from $\mathcal{Q}$. When we deal with a pair distribution $\mathcal{Q}$, the above generative model lies in the heart of the pairwise comparisons [FV86, Mar96, WJJ13]. The comparisons are generated according to the Bradley-Terry-Luce (BTL) model (where the comparisons have size 2, [BT52, Luc59]): there is an unknown weight vector $\mathcal{D} \in \mathbb{R}^n_+$ ($\mathcal{D}(u)$ indicates the quality of item u) and, for two items u, v, the algorithm observes that u beats v with probability $\frac{\mathcal{D}(u)}{\mathcal{D}(u)+\mathcal{D}(v)}$. Motivated by the importance of perfect simulation, we ask the following question:

Question 1. *Is there an efficient algorithm that draws i.i.d. samples (x, S) from* $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ *and generates a single sample $y \sim \mathcal{D}$?*

To put our contribution into context, let us ask another question whose answer is clear from previous work. What is the sample complexity of *learning* $\mathcal{D}$ from a Local Sampling Scheme $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$? Here, the goal is to estimate $\mathcal{D}$ in some L_p norm given such comparisons. Some works focus on learning the re-parameterization $\boldsymbol{z} \in \mathbb{R}^n$ with $z_x = \log(\mathcal{D}(x))$. Depending on the context, either the normalization condition $\langle \mathbf{1}, \mathcal{D}\rangle = 1$ or $\langle \mathbf{1}, \boldsymbol{z}\rangle = 0$ is used. [SBB+16] deal with the problem of learning the re-parameterization $\boldsymbol{z}$ in L_2 norm when $|S| = 2$ (pairwise comparisons) or $|S| = k$ (k-wise comparisons with $k = O(1)$). The sample complexity for learning $\boldsymbol{z}$ in L_2 norm from pairwise comparisons is $n/(\lambda(\boldsymbol{Q})\epsilon^2)$, where $\lambda(\boldsymbol{Q})$ is the second smallest eigenvalue of a Laplacian matrix $\boldsymbol{Q}$, induced by the pair distribution $\mathcal{Q}$ (the learning result is tight in a minimax sense for

[1] In Appendix G, we discuss we extend our results sets of size larger than 2.

some appropriate semi-norm [SBB+16]). In particular, the matrix $\boldsymbol{Q}$ is defined as $\boldsymbol{Q}_{xy} = -\mathcal{Q}(x, y)$ and $\boldsymbol{Q}_{xx} = \sum_{y \neq x} \mathcal{Q}(x, y)$. Hence, in the pairwise comparisons setting, the sample complexity of learning incurs an overhead associated with $\lambda(\boldsymbol{Q})$ (Table 1). The algorithm of [SBB+16] is a pivotal component for our main result, as we will see later. [APA18] provide a similar result for learning $\mathcal{D}$ in TV distance using a random-walk approach. We remark that learning from pairwise comparisons requires some mild conditions concerning the sampling process (LSS in our case). We review them shortly below.

Table 1: Learning and Exact Sampling from PCs of size 2. $\widetilde{O}(\cdot)$ subsumes logarithmic factors.

Sample Access	**Learning** $(\epsilon > 0)$	**Exact Sampling**
Definition 1 with Pairwise Comparisons	$(\boldsymbol{z}, L_2)$ $O(n/\epsilon^2) \cdot \frac{1}{\lambda(\boldsymbol{Q})}$ [SBB+16]	$\widetilde{O}(n^2) \cdot \frac{1}{\lambda(\boldsymbol{Q})}$ (Theorem 3)

Conditions for Local Sampling Schemes. We provide a standard pair of conditions for Local Sampling Schemes (these or similar conditions are also needed in the learning problem). We are interested in LSSs that satisfy both of these conditions. However, some of our results still hold when only the first condition is true; this will be more clear later. Assumption 1 is a necessary information-theoretic condition for the pair distribution $\mathcal{Q}$.

Assumption 1 (Identifiability)**.** *The support of the pair distribution $\mathcal{Q}$ of the Local Sampling Scheme contains a spanning tree, i.e., the induced graph $G_{\mathcal{Q}}$ is connected. We define $\mathcal{E} := E(G_{\mathcal{Q}})$.*

Intuitively, the pair distribution $\mathcal{Q}$ needs to be supported on a spanning tree. Equivalently, the associated Laplacian matrix $\boldsymbol{Q}$ should have positive Fiedler eigenvalue $\lambda(\boldsymbol{Q}) > 0$. Assumption 2 is a condition about the support of $\mathcal{Q}$ *and* the target distribution $\mathcal{D}$; it allows efficient learning of $\mathcal{D}$ from pairwise comparisons and is related to the difficulty of estimating small $\mathcal{D}$-probabilities.

Assumption 2 (Efficiently Learnable)**.** *There exists a constant $\phi > 1$ such that target distribution $\mathcal{D}$ satisfies $\frac{1}{\phi} \leq \max_{(x,y)\in\mathcal{E}} \frac{\mathcal{D}(x)}{\mathcal{D}(y)} \leq \phi$, where $\mathcal{E}$ is the support of the distribution $\mathcal{Q}$.*

From a dual perspective, Assumption 2 says that $\mathcal{Q}$ is supported only on edges where the two corresponding $\mathcal{D}$-ratios are well controlled and implies that any conditional distribution $\mathcal{D}_{\{x,y\}}$ has bounded variance, i.e., $(1/\phi)/(1+\phi)^2 \leq \frac{\mathcal{D}(x)\mathcal{D}(y)}{(\mathcal{D}(x)+\mathcal{D}(y))^2} \leq \phi/(1+1/\phi)^2$. A quite similar property has been proposed in the problem of learning from pairwise comparisons [SBB+16].

Contributions and Techniques. We provide an algorithmic answer to Question 1 regarding perfect sampling (under the above mild conditions) by showing the following:

There exists an efficient algorithm that draws $\widetilde{O}(n^2/\lambda(\boldsymbol{Q}))$ samples from $\mathrm{Samp}(\mathcal{Q}; \mathcal{D})$ and outputs a clean sample $x \sim \mathcal{D}$.

The tool that makes this possible (and algorithmic) is a novel variant of the elegant Coupling from the Past (CFTP) algorithm of [PW96]. CFTP is a technique developed by Propp and Wilson ([PW96, PW98]), that provides an exact random sample from a Markov chain, that has law the (unique) stationary distribution (we refer to Appendix A.2.3 for an introduction to CFTP). To make our contribution clear, we provide two theorems; Informal Theorem 1 that directly and "naively" applies CFTP (indicating the challenges behind efficient sample correction) and a more sophisticated Informal Theorem 2 that *combines CFTP and learning from pairwise comparisons* in order to get the result of Table 1. We show how to efficiently generate "global" samples from the true distribution $\mathcal{D}$ from few samples from a Local Sampling Scheme $\mathrm{Samp}(\mathcal{Q}; \mathcal{D})$. To this end, our novel algorithm combines the CFTP method [PW96, PW98], a remarkable technique that generates exact samples from the stationary distribution of a given Markov chain, with (i) an efficient algorithm that estimates the parameters of Bradley-Terry-Luce ranking models from pairwise (and, in general, k-wise) comparisons [SBB+16] and (ii) procedures based on Bernoulli factories [DHKN17, MP05, NP05] and rejection sampling.

At a technical level, we observe (Section 2) that a Local Sampling Scheme $\mathrm{Samp}(\mathcal{Q}; \mathcal{D})$ naturally induces a canonical ergodic Markov chain on $[n]$ with transition matrix $\boldsymbol{M}$. The probability of a transition from a state x to a state y is equal to the probability $\mathcal{Q}(x, y)$, that the pair $\{x, y\}$ is drawn

from $\mathcal{Q}$, times the probability $\mathcal{D}(y)/(\mathcal{D}(x)+\mathcal{D}(y))$, that y is drawn from the conditional distribution $\mathcal{D}_{\{x,y\}}$. Then, the unique stationary distribution of the resulting Markov chain coincides with the true distribution $\mathcal{D}$. Applying (a non-adaptive version of) the Coupling from the Past algorithm, we obtain the following result under Assumption 1 and Assumption 2 for a target distribution $\mathcal{D}$ over $[n]$. For the matrix $\boldsymbol{M}$, we let $\Gamma(\boldsymbol{M})$ be its absolute spectral gap (see Appendix A.1 for extensive notation).

Informal Theorem 1 (Direct Exact Sampling from LSS)**.** *With an expected number of* $\widetilde{O}\left(\frac{n^2}{\Gamma(\boldsymbol{M})}\right)$ *samples from a Local Sampling Scheme* $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$*, there exists an algorithm (Algorithm 4) that generates a sample distributed as in* $\mathcal{D}$*.*

The sample complexity of Informal Theorem 1 is, for instance, attained by the instance of Figure 1, which we will discuss shortly.[2] We remark that Informal Theorem 1 holds more generally for Local Sampling Schemes that only satisfy Assumption 1 and not Assumption 2; for a formal version of the above result, we refer to Theorem 2 and for a discussion on the resulting sample complexity, we refer also to Remark 1. Let us now see why Informal Theorem 1 is quite unsatisfying.

In Informal Theorem 1, the matrix $\boldsymbol{M}$ contains the transitions induced by $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ and the transition $u \to v$ is performed with probability $\mathcal{Q}(u,v)\mathcal{D}(v)/(\mathcal{D}(u)+\mathcal{D}(v))$ and $\Gamma(\boldsymbol{M})$ is the absolute spectral gap of the Markov chain's transition matrix $\boldsymbol{M}$ on which CFTP is performed. Note that both $\boldsymbol{M}$ and $\Gamma(\boldsymbol{M})$ depend on $\mathcal{Q}$ and the target distribution $\mathcal{D}$. In many natural cases, e.g., if $\mathcal{D}$ is multimodal or includes many low probability points, the CFTP-based algorithm of Informal Theorem 1 can be quite inefficient! For instance, Figure 1 is a bad instance for the standard CFTP algorithm. It corresponds to an instance where the support of $\mathcal{Q}$ is the path graph over $[n]$ and the target distribution $\mathcal{D}$ is bimodal, satisfying Assumption 1 and Assumption 2 with $\phi = 2$. In this case, $\boldsymbol{M}$ captures the transitions of a negatively biased nearest-neighbor random walk on the path, i.e., the probability of moving to the boundary is larger than moving to the interior. The coalescence probability of the two extreme points is of order α^n for some $0 < \alpha < 1$. This probability is connected to the maximum hitting time and, consequently, we get that the mixing time is $\Omega(\alpha^{-n})$ [LP17] and so the coalescence time is exponentially large. Informal Theorem 1 depends on $\Gamma(\boldsymbol{M})$ and hence the *exponential dependence on* n is reflected in the conductance of the chain, which is exponentially small. So, can we eliminate the bias due to Definition 1 efficiently?

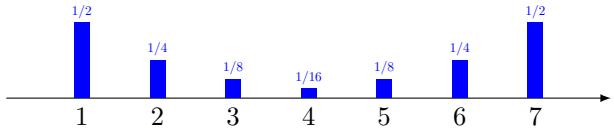

Figure 1: A bad instance for Informal Theorem 1.

As a technical contribution, we provide a novel exact sampling algorithm whose convergence does not depend on $\mathcal{D}$. We show how to improve the efficiency of CFTP by removing the dependence of its sample (and computational) complexity on the structure of the true distribution $\mathcal{D}$. In particular, in Section 3, we show the following under Assumption 1 and Assumption 2:

Informal Theorem 2 (Exact Sampling from LSS using Learning)**.** *There exists an algorithm (Algorithm 1 & Algorithm 2) that draws an expected number of* $\widetilde{O}\left(\frac{n^2}{\lambda(\boldsymbol{Q})}\right)$ *samples from a Local Sampling Scheme* $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$*, and generates a sample distributed as in* $\mathcal{D}$*.*

In the above, $\lambda(\boldsymbol{Q})$ is the second smallest eigenvalue (a.k.a., the Fiedler eigenvalue) of $\boldsymbol{Q}$, the weighted Laplacian matrix of the graph $G_{\mathcal{Q}}$ induced by $\mathcal{Q}$; this is the same matrix that appears in the learning result of Table 1. *Crucially, the sample complexity in the above result is polynomial in* n, provided that the pair distribution $\mathcal{Q}$ is reasonable. For the path graph of Figure 1 with $\mathcal{Q}$ the uniform distribution, Informal Theorem 1 yields an exponential sample complexity but the spectrum of the matrix of Informal Theorem 2 is $\mathrm{spec}(\boldsymbol{Q}) = \{\frac{1}{n}\Theta(1-\cos(\pi i/n))\}_{i=0}^{n-1}$. Hence, since $1-\cos(x) = \sin^2(x/2) \approx x^2$, we get that the sample complexity is $\mathrm{poly}(n)$.

[2]For the instance of Figure 1 one can reduce the sample complexity of our results by a factor of n, since the state space has a natural ordering and one could apply monotone CFTP (it suffices to coalesce two random processes and not n).

Technical Overview. A first key idea towards establishing Informal Theorem 2 is to exploit *learning* in order to accelerate the convergence of CFTP. We first apply (a modification of) the gradient-based algorithm of [SBB$^+$16] to the empirical log-likelihood objective, which estimates the parameters of BTL ranking models from PCs, with samples from $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$. As a result of this learning algorithm (whose efficiency requires Assumption 2, as in [SBB$^+$16]), we get a probability distribution $\widetilde{\mathcal{D}}$, which approximates $\mathcal{D}$ with small relative error, in the sense that $\mathcal{D}(x) \in (1 \pm \epsilon) \cdot \widetilde{\mathcal{D}}(x)$, for any x in $\mathcal{D}$'s support. As we saw in Table 1, the sample complexity will be roughly of order $O(n/(\lambda(\boldsymbol{Q})\epsilon^2)))$. To be precise, this learning phase gives us a sample complexity term of order $O(n^2/\lambda(\boldsymbol{Q}))$ and corresponds to a single execution of the above learning algorithm for the target $\mathcal{D}$ in relative error with accuracy $\epsilon = \Theta(1/\sqrt{n})$ (the accuracy's choice will be more clear later).

The new idea is that we can use the output of the learning algorithm $\widetilde{\mathcal{D}}$ and a Bernoulli factory mechanism to transform the Markov chain induced by $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ into an ergodic chain with almost uniform stationary probability distribution, whose transition probabilities (and mixing time) essentially do not depend on $\mathcal{D}$! To do so, for each pair of states x, y, with $(x, y) \in \mathcal{E}$ and $\widetilde{\mathcal{D}}(x) > \widetilde{\mathcal{D}}(y)$, we can use Bernoulli downscaling (which constitutes an essential building block for general Bernoulli factories, see e.g., [DHKN17, MP05, NP05]) to transform the (unknown) probability of a transition from y to x to $\mathcal{D}(x)/(\mathcal{D}(x) + \mathcal{D}(y)) \cdot \widetilde{\mathcal{D}}(y)/\widetilde{\mathcal{D}}(x)$ (conditional that the edge (x, y) is drawn). Since $\widetilde{\mathcal{D}}(x) \approx \mathcal{D}(x)$, for all states x, this makes the probability of a transition from x to y almost equal to the probability of a transition from y to x, which, in turn, implies that *the stationary distribution of the modified Markov chain is almost uniform*. Thus, we speed up the CFTP algorithm by removing the modes due to the landscape of the target $\mathcal{D}$ and can get an exact sample from the stationary distribution of the modified Markov chain with $O\left(n \log(n)/\lambda(\boldsymbol{Q})\right)$ samples. Namely, the sample complexity of this parameterized variant of the CFTP algorithm does not depend on the structure of $\mathcal{D}$ anymore (of course, it does depend on the size n of $\mathcal{D}$'s support). The above constitute our second key idea and the intuition behind this re-weighting comes from the method of simulated tempering [MP92, GLR18b, GLR18a].

Our third idea deals with the output of the algorithm. Since we want an exact/clean sample from $\mathcal{D}$, we cannot simply print the output of the above process. As a last step, we use rejection sampling, guided by the distribution $\widetilde{\mathcal{D}}$, and apply the "reverse" modification, bringing the distribution of the resulting sample back to $\mathcal{D}$ (see Algorithm 1). With high probability, this last rejection sampling will succeed after $\Theta(n)$ executions of the parameterized CFTP algorithm. However, note that we do not have to execute the learning algorithm again. Hence, the total sample complexity is equal to the cost of the learning phase (that is performed only once and its output is given as input to the parameterized CFTP algorithm; see Algorithm 1) and the cost of the modified CFTP multiplied by $\Theta(n)$ (due to rejection sampling), which yields a total of $\widetilde{O}(n^2/\lambda(\boldsymbol{Q}))$ samples. The above modification of the Markov chain induced by $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$, which improves possibly significantly its mixing time, is simple and general and the parameterized variation of the standard CFTP algorithm (Algorithm 1) might be of independent interest. In Appendix G, we discuss how to extend the analysis of the main Algorithm 1 to sets of size larger than 2.

Related Work. Our results are related to and draws ideas from several areas of TCS:
Exact Sampling. Exact sampling comprises a well-studied field [Ken05, Hub16] with numerous applications in TCS (e.g., to approximate counting [Hub98]). One of our main tools, the Coupling From the Past algorithm, is the breakthrough result of [PW96, PW98], which established the applicability of perfect simulation, and made efficient simulation of challenging problems (e.g., Ising model) feasible. After this fundamental result, the literature of perfect simulation flourished and various exact sampling procedures have been developed, see e.g., [PW96, GM99, Wil00, HS00, FH00, Hub04]), and the references therein.
Learning and Ranking from Pairwise Comparisons. There is a vast amount of literature on ranking and parameter estimation from pairwise comparisons (see e.g., [FV86, Mar96, Cat12] and the references therein). A significant challenge in that field is to come up with models that accurately capture uncertainty in pairwise comparisons. The standard ones are the Bradley-Terry-Luce (BTL) model [BT52, Luc59] and the Thurstone model [Thu27]. Our work is closely related to previous work on the BTL model [RA14, SBGW16, DKNS01, NOS12, Hun04, JLYY11, APA18, LSR21] and our sample complexity rates depend on the Fiedler eigenvalue of a pairwise comparisons' matrix, as in [HOX14, SBB$^+$16, VY16, NOS17, VYZ20].
Truncated Statistics. Our work falls in the research agenda of efficient learning and exact sampling

from truncated samples [Bre96, Coh16]. Recent work has obtained efficient algorithms for efficient learning from truncated [DGTZ18, KTZ19, DGTZ19, IZD20, DRZ20, BDNP21, NP19, FKT20, DKTZ21] and censored [MMS21, FKKT21, Ple21] samples. Closer to our work, [FKT20] also deal with the question of exact sampling from truncated samples.

Correction Sampling. Sample correction is a field closely related to our work. This area of research deals with the case where input data are drawn from a noisy or imperfect source and the goal is to design *sampling correctors* [CGR18] that operate as filters between the noisy data and the end user. These algorithms exploit the structure of the distribution and make "on-the-fly" corrections to the drawn samples. Moreover, the problem of local correction of data has received much attention in the contexts of self-correcting programs, locally correctable codes, and local filters for graphs and functions (see e.g., [BLR93, SS10, JR13, ACCL04, BGJ+10, GLT20]).

Conditional Sampling. Conditional sampling deals with the *adaptive* learning analog of LSS, where the goal is again to learn an underlying discrete distribution from conditional/truncated samples, but the learner can change the truncation set on demand and is extensively studied [Can15, FJO+15, ACK15b, BC18, ACK15a, GTZ17, KT19, CCK+19].

2 An Exact Sampling Algorithm using CFTP

Consider a discrete target distribution $\mathcal{D}$. For pair distributions $\mathcal{Q}$, the Local Sampling Scheme $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ can be regarded as a graph $G=(V,E)$ with $V=\mathrm{supp}(\mathcal{D})$ and $\mathcal{E}=\mathrm{supp}(\mathcal{Q})$. We let $V=[n]$. Let $\boldsymbol{M}$ denote the canonical transition matrix of the Markov chain, associated with the oracle $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$. The entries of $\boldsymbol{M}=[\boldsymbol{M}_{xy}]_{x,y\in[n]}$ are defined as:

$$\boldsymbol{M}_{xy}=\mathcal{Q}(x,y)\frac{\mathcal{D}(y)}{\mathcal{D}(x)+\mathcal{D}(y)} \text{ for any } x\neq y \text{ and } \boldsymbol{M}_{xx}=1-\sum_{y\neq x}\boldsymbol{M}_{xy} \text{ for } x\in[n]\,. \tag{1}$$

Observe that the transition from x to y is performed when both the edge $\{x,y\}$ is drawn from $\mathcal{Q}$, with probability proportional to $\mathcal{Q}(x,y)$ and y is drawn from the conditional distribution $\mathcal{D}_{\{x,y\}}$. The Markov chain, whose transitions are described by $\boldsymbol{M}$, has some notable properties: it has $\mathcal{D}$ as stationary distribution and is ergodic. We can invoke the fundamental CFTP algorithm over the Markov chain of [PW96, PW98] to obtain a perfect sample from $\mathcal{D}$. We extensively discuss CFTP in Appendix A.2.3 for the interested reader. CFTP yields the following (unsatisfying) result, whose proof can be found at the Appendix B.

Theorem 2 (Direct Exact Sampling from LSS)**.** *Let $\mathcal{D}_{\min}$ be the minimum value of the target distribution $\mathcal{D}$. Under Assumption 1 and for any $\delta>0$, Algorithm 4 draws, with probability at least $1-\delta$, $N=O\left(\frac{n\log(1/\mathcal{D}_{\min})}{\Gamma(\boldsymbol{M})}\log(\frac{1}{\delta})\right)$ samples from a Local Sampling Scheme $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ over $[n]$, runs in time polynomial in N and outputs a sample distributed as in $\mathcal{D}$.*

Note that in the above result (Theorem 2), the confidence parameter δ corresponds to the number of samples required and not the quality of the output sample (the sample is perfect with probability 1). We comment that the proof of the above Theorem is nearly straightforward and follows from the analysis of the CFTP algorithm; the purpose of Theorem 2 is to indicate that a direct application of CFTP would potentially lead to an exponential number of samples due to the shape of the target distribution. Our main technical contribution begins in the next section: We give an exact sampler that surpasses the dependence on the target's structure using learning and rejection sampling.

3 Improving the Exact Sampling Algorithm using Learning

The main drawback of Theorem's 2 algorithm is that its sample complexity depends on both $\boldsymbol{Q}$ and $\boldsymbol{D}$ and consequently in many natural scenaria the sample complexity may be exponentially large in n. Next, we present Algorithm 1, which will allow us to remove the dependence on $\boldsymbol{D}$ for the Markov chain and reduce the sample complexity to polynomial in the domain size. The crucial differences compared to Algorithm 4 are the **orange** lines. Algorithm 1 is a parameterized extension of CFTP and its input is a vector $\boldsymbol{p}\in[0,1]^n$. We remark that when $\boldsymbol{p}$ is the all-ones vector $\mathbf{1}$, we obtain the previous Algorithm 4.

We perform Algorithm 1 using as input parameter $\boldsymbol{p}$ an estimate $\widetilde{\mathcal{D}}$ of the target $\mathcal{D}$ with (relative) error of order $\epsilon=1/\sqrt{n}$ (this estimate is obtained via Theorem 4 & Algorithm 5 using $O(n^2/\lambda(\boldsymbol{Q}))$

Algorithm 1 Parameterized Extension of Algorithm 4

1: **Input:** $\boldsymbol{p} \in [0,1]^n$ ▷ *For instance, this may be the output of Algorithm 5.*
2: **procedure** EXACTSAMPLER($\boldsymbol{p}$) ▷ *Sample access to the LSS oracle* $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$.
3: $t \leftarrow 0$, $F_0(x) \leftarrow x$, for any $x \in [n]$
4: **while** F_t has not coallesced **do** ▷ *While no coalescence has occured.*
5: $t \leftarrow t-1$
6: Draw sample (i,j,w) with $(i,j) \in \mathcal{E}$ and $w \in \{i,j\}$
7: **for** $x = 1 \dots n$ **do** ▷ *In order to update state* x.
8: **if** $x \notin \{i,j\}$ or $x = w$ **then** $F_t(x) \leftarrow F_{t+1}(x)$
9: **else** $F_t(x) \leftarrow \begin{cases} F_{t+1}(w), & \text{with probability } \min\{p(x)/p(w),1\} \\ F_{t+1}(x), & \text{otherwise} \end{cases}$
10: **end**
11: **end**
12: Draw $C \sim \mathrm{Be}(p(F_t(1)))$ ▷ *Remove bias using rejection sampling.*
13: **if** $C = 1$ **then** Output $F_t(1)$ **else** Output $\perp$ ▷ *Output the perfect sample or Fail.*
14: **end procedure**

samples from $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$, as we will see later). Under the perspective of LSSs as random walks on an irreducible Markov chain, Algorithm 1 proceeds by executing CFTP. Recall that in each draw from $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$, a transition can be realized by the matrix $\boldsymbol{M}$. The fact that this transition depends on the target $\mathcal{D}$ is pathological, as we observed previously. The algorithm, instead of performing this transition (as the non-adaptive CFTP algorithm of Theorem 2 does), performs a Bernoulli factory mechanism (downscaling), using the estimates $\widetilde{\mathcal{D}}$ in order to make each transition almost uniform (see Line 9 of Algorithm 1). This change introduces some (known) bias to the random walk and changes the stationary distribution from $\mathcal{D}$ to an almost uniform one. By performing the CFTP method, the algorithm iteratively reconstructs the past of the infinite simulation, until all the simulations have coalesced at time $t = 0$. When coalescence occurs, the algorithm has an exact sample from the biased (almost uniform) stationary distribution. Since the introduced bias is known (it corresponds to the inverse of the estimates $\widetilde{\mathcal{D}}$), the algorithm accepts the sample using rejection sampling, guided by $\widetilde{\mathcal{D}}$, so that the final sample is distributed according to the target distribution $\mathcal{D}$ (see Line 13 of Algorithm 1). Specifically, we show (for the proof, see Appendix C) that:

Theorem 3 (Exact Sampling from LSS using Learning)**.** *Under Assumption 1 and Assumption 2, for any $\delta > 0$, there exists an algorithm (Algorithm 2) that draws, with probability at least $1-\delta$, $N = O\left(n^2 \log^2(n)/\lambda(\boldsymbol{Q}) \cdot \log(1/\delta)\right)$ samples from a Local Sampling Scheme $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$, runs in time polynomial in N, and outputs a sample distributed as in $\mathcal{D}$.*

Algorithm 2 Exact Sampling from Local Sampling Schemes of Informal Theorem 2 & Theorem 3

1: **procedure** EXACTSAMPLERWITHLEARNING(δ) ▷ *The algorithm of Theorem 3.*
2: $\widetilde{\mathcal{D}} \leftarrow$ LEARN($\epsilon := 1/\sqrt{n}, \delta$) ▷ *Learn $\mathcal{D}$ in relative error with Algorithm 5 (Theorem 4).*
3: x $\leftarrow \perp$
4: **while** x $= \perp$ **repeat**
5: x $\leftarrow$ EXACTSAMPLER($\widetilde{\mathcal{D}}$) ▷ *Call Algorithm 1.*
6: Output x
7: **end procedure**

We proceed with the analysis of Algorithm 2, which can be decomposed into two parts: the Learning Step (Line 2) and several iterations of Algorithm 1 (Line 5).

Learning Step. For two distributions $\mathcal{D}, \widetilde{\mathcal{D}}$ with ground set $[n]$, we introduce the sequence/list (of length n) $1-\mathcal{D}/\widetilde{\mathcal{D}} := (1-\mathcal{D}(x)/\widetilde{\mathcal{D}}(x))_{x\in[n]}$. Observe that the pair of sequences $(1-\mathcal{D}/\widetilde{\mathcal{D}}, 1-\widetilde{\mathcal{D}}/\mathcal{D})$ captures the relative error between the two distributions. The sample complexity of the task of learning $\mathcal{D}$ in ϵ-relative error is summarized by the following:

Theorem 4 (Learning Phase)**.** *For any $\epsilon, \delta > 0$, there exists an algorithm (Algorithm 5) that draws $N = O\left(\frac{n}{\lambda(\boldsymbol{Q})\epsilon^2}\log(\frac{1}{\delta})\right)$ samples from $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ satisfying Assumptions 1 and 2, runs in time*

polynomial in N, and, with probability at least $1-\delta$, computes an estimate $\widetilde{\mathcal{D}}$ of the target distribution $\mathcal{D}$, that satisfies the relative error guarantee $\max\left\{\|1-\mathcal{D}/\widetilde{\mathcal{D}}\|_\infty, \|1-\widetilde{\mathcal{D}}/\mathcal{D}\|_\infty\right\} \leq \epsilon$.

We execute *once* the learning algorithm (Algorithm 5) that estimates $\mathcal{D}$ in relative error and we choose accuracy $\epsilon = 1/\sqrt{n}$. We would like to shortly comment on the choice of the accuracy value: An accuracy of constant order would still potentially yield a biased random walk. After the re-weighting step, our goal is to show that the target distribution becomes almost uniform and that the absolute spectral gap of the reformed matrix is of order $\lambda(\boldsymbol{Q})$ (see the proof of Lemma 15). In order to ensure this, any accuracy that vanishes with n is sufficient. We chose to set $\epsilon = 1/\sqrt{n}$ so that the complexity of learning $(\frac{n}{\lambda(\boldsymbol{Q})\epsilon^2})$ matches that of the exact sampling procedure $(\frac{n^2}{\lambda(\boldsymbol{Q})})$. This learning step costs $n^2/\lambda(\boldsymbol{Q})$ samples and the estimate $\widetilde{\mathcal{D}}$ is used as input to Algorithm 1. For the proof of Theorem 4, we refer to the Appendix E. We can now move to Line 5.

Executing Algorithm 1. We next focus on a single execution of the parameterized CFTP algorithm. Our goal is to transform the Markov chain of the Local Sampling Scheme defined in Equation (1) to a modified Markov chain with an almost uniform stationary distribution. We are going to provide some intuition. Applying Theorem 4, we can consider that, for any $x \in [n]$, there exists a coefficient $\widetilde{\mathcal{D}}(x) \approx \mathcal{D}(x)$. The idea is to use $\widetilde{\mathcal{D}}(x)$ and make the stationary distribution of the modified chain close to uniform. Intuitively, this transformation should speedup the convergence of the CFTP algorithm. The downscaling method of Bernoulli factories gives us a tool to do this modification.

We can implement the modified Markov chain via downscaling as follows: Consider an edge $\{x, y\}$ with transition probability pair (p_{xy}, p_{yx}). Without loss of generality, we assume that $\widetilde{\mathcal{D}}(y) > \widetilde{\mathcal{D}}(x)$ (which intuitively means that we should expect that $p_{xy} > p_{yx}$). Then, the downscaler leaves p_{yx} unchanged and reduces the mass of p_{xy} to make the two transitions almost balanced. Consider the LSS transition matrix $\boldsymbol{M}$ with $\boldsymbol{M}_{xy} = \mathcal{Q}(x,y)\frac{\mathcal{D}(y)}{\mathcal{D}(x)+\mathcal{D}(y)}$ and $\boldsymbol{M}_{xx} = 1 - \sum_{y\neq x}\boldsymbol{M}_{xy}$. Also, let $\widetilde{\mathcal{D}}$ be an estimate for the distribution $\mathcal{D}$ (Theorem 4). For the pair (x, y), we modify the transition probability $p_{xy} := \boldsymbol{M}_{xy}$, only if $p_{xy} > p_{yx}$, to be equal to the following: $\widetilde{p}_{xy} = \mathcal{Q}(x,y)\frac{\mathcal{D}(y)}{\mathcal{D}(x)+\mathcal{D}(y)}\frac{\widetilde{\mathcal{D}}(x)}{\widetilde{\mathcal{D}}(y)} \approx \mathcal{Q}(x,y)\frac{\mathcal{D}(x)}{\mathcal{D}(x)+\mathcal{D}(y)} = p_{yx}$, where we use that $\widetilde{\mathcal{D}}(x) \approx \mathcal{D}(x)$ and $\widetilde{\mathcal{D}}(y) \approx \mathcal{D}(y)$. The transition probability from x to y corresponds to a Bernoulli variable $\mathrm{Be}(p_{xy})$, which is downscaled by $\widetilde{\mathcal{D}}(x)/\widetilde{\mathcal{D}}(y) < 1$. The modified transition probability $\widetilde{p}_{xy}$ can be implemented by drawing a $\Lambda \sim \mathrm{Be}(\widetilde{\mathcal{D}}(x)/\widetilde{\mathcal{D}}(y))$ and then drawing a $P \sim \mathrm{Be}(p_{xy})$ (from $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$) and, finally, realizing the transition from x to y only if $\Lambda P = 1$. This implementation is valid since the two sources of randomness are independent. So, we perform a downscaled random walk based on the transition matrix $\widetilde{\boldsymbol{M}}$ with $\widetilde{\boldsymbol{M}}_{xy} = \widetilde{p}_{xy}$ (the "small" probability p_{yx} remained intact). We call $\widetilde{\boldsymbol{M}}$ the $\widetilde{\mathcal{D}}$-scaling of $\boldsymbol{M}$. Lemma 5 summarizes the key properties of $\widetilde{\boldsymbol{M}}$. The definition of the mixing time, used below, can be found at the notation Appendix A.1 and the full proof can be found at the Appendix D.

Lemma 5 (Properties of Rescaled Random Walk)**.** *Let $\mathcal{D}$ be a distribution on $[n]$ and consider an ϵ-relative approximation $\widetilde{\mathcal{D}}$ of $\mathcal{D}$, as in Theorem 4 with $\epsilon = 1/\sqrt{n}$. Consider the transition matrix $\boldsymbol{M}$ of the Local Sampling Scheme* $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ *(see Equation* (1)*), and let $\widetilde{\boldsymbol{M}}$ be the $\widetilde{\mathcal{D}}$-scaling of $\boldsymbol{M}$. Then, the transition matrix $\widetilde{\boldsymbol{M}}$ (i) has stationary distribution $\widetilde{\pi}_0$ with $\widetilde{\pi}_0(x) = \Theta(1/n)$ for all $x \in [n]$ and (ii) the mixing time of the transition matrix $\widetilde{\boldsymbol{M}}$ is $T_{\mathrm{mix}}(\widetilde{\boldsymbol{M}}) = O\left(\frac{\log(n)}{\lambda(\boldsymbol{Q})}\right)$.*

Proof Sketch. Set $J_{xy} = \mathcal{D}(x) + \mathcal{D}(y)$. For a pair $(x, y) \in \mathcal{E}$, before the downscaling, the pairwise (x, y)-comparison corresponds to the random coin $(\mathcal{D}(y)/J_{xy}, \mathcal{D}(x)/J_{xy})$ and let $\mathcal{D}(y) > \mathcal{D}(x)$. After the downscaling, this coin becomes (locally) almost fair, i.e., $(\mathcal{D}(x)/J_{xy} + \xi_{xy}, \mathcal{D}(x)/J_{xy})$. Hence, the modified transition matrix $\widetilde{\boldsymbol{M}}$ can be written as $\widetilde{\boldsymbol{M}} = \widetilde{\boldsymbol{D}} \otimes \boldsymbol{Q} + \boldsymbol{Q} \circ [\xi_{xy}]$, where $\widetilde{\boldsymbol{D}}$ is a symmetric matrix with $\widetilde{\boldsymbol{D}}_{xy} = \min\{\mathcal{D}(x), \mathcal{D}(y)\}/(\mathcal{D}(x) + \mathcal{D}(y)) = \widetilde{\boldsymbol{D}}_{yx}$, $\boldsymbol{Q}$ is the Laplacian matrix of the graph $G_{\mathcal{Q}}$ and $[\xi_{xy}]$ is a matrix with entries ξ_{xy} where ξ_{xy} is only non-zero in the modified transitions $x \to y$, i.e., if $p_{xy} > p_{yx}$ then $\widetilde{\boldsymbol{M}}_{xy} = \mathcal{Q}(x, y) \cdot (\mathcal{D}(x)/J_{xy} + \xi_{xy})$; for this pair, we set $\xi_{yx} = 0$. Finally, $\boldsymbol{Q} \circ [\xi_{xy}]$ denotes the modified Hadamard product[3] between $\boldsymbol{Q}$ and $[\xi_{xy}]$. Due to the accuracy ϵ of the learning algorithm, we have that $|\xi_{xy}| = O(1/\sqrt{n})$.

[3]We have that $\left(\boldsymbol{Q} \circ \boldsymbol{A}\right)_{xy} = \mathcal{Q}_{xy}A_{xy}$, $\left(\boldsymbol{Q} \circ \boldsymbol{A}\right)_{xx} = -\sum_{y\neq x}\mathcal{Q}_{xy}A_{xy}$.

We analyze (i) the stationary distribution and (ii) the mixing time. For **Part (i)**, the chain $\widetilde{\boldsymbol{M}}$ remains irreducible and the detailed balance equations of $\widetilde{\boldsymbol{M}}$ satisfy $\widetilde{p}_{xy}/\widetilde{p}_{yx} = \mathcal{D}(y) \cdot \widetilde{\mathcal{D}}(x)/(\widetilde{\mathcal{D}}(y) \cdot \mathcal{D}(x))$. Since $\widetilde{\mathcal{D}}$ is an ϵ-relative approximation of $\mathcal{D}$ with $\epsilon = O(1/\sqrt{n})$, it holds that for any $x \in [n]$: $\mathcal{D}(x)/\widetilde{\mathcal{D}}(x) \in [1-\epsilon, 1+\epsilon]$. Hence, the unique stationary distribution $\widetilde{\pi}_0$ satisfies: $\widetilde{\pi}_0(x)/\widetilde{\pi}_0(y) \in [1-\Theta(1/\sqrt{n}), 1+\Theta(1/\sqrt{n})]$ for any $x, y \in [n]$. So, it must be the case that $\widetilde{\pi}(x) = \Theta(1/n)$.

For **Part (ii)**, in order to show that the mixing time is of order $O(\log(n)/\lambda(\boldsymbol{Q}))$, it suffices to show (by Lemma 9) that (i) the minimum value of the stationary distribution is $O(1/n)$ and (ii) the absolute spectral gap $\Gamma(\widetilde{\boldsymbol{M}})$ of $\widetilde{\boldsymbol{M}}$ satisfies $\Gamma(\widetilde{\boldsymbol{M}}) = \Omega(\lambda(\boldsymbol{Q}))$, where $\lambda(\boldsymbol{Q})$ is the minimum non-zero eigenvalue of the Laplacian matrix $\boldsymbol{Q}$. The first property follows directly from the closeness of the stationary distribution of $\widetilde{\boldsymbol{M}}$ to the uniform one. We now focus on the latter. Our goal is to control the absolute spectral gap $\Gamma(\widetilde{\boldsymbol{M}})$. In fact, this matrix can be seen as a perturbed version of the matrix $\widetilde{\boldsymbol{D}} \otimes \boldsymbol{Q}$ as we discussed in the beginning of the proof sketch. The perturbation noise matrix $\boldsymbol{E} = \boldsymbol{Q} \circ [\xi_{xy}]$ is a matrix whose entries contain (among others) the approximation errors $[\xi_{xy}]$. Hence, we can use Weyl's inequality [Tao12] in order to control the absolute spectral gap of $\widetilde{\boldsymbol{M}}$. This inequality gives a crucial perturbation bound for the singular values for general matrices. Since the matrix $\widetilde{\boldsymbol{M}}$ can be seen as a perturbation (as we discussed previously), we can (informally speaking) use Weyl's inequality to control the singular values After some algebraic manipulation, we derive that $\Gamma(\widetilde{\boldsymbol{M}}) = \Omega(\lambda(\boldsymbol{Q}))$. □

So, a *single iteration* of the parameterized CFTP algorithm with transition matrix $\widetilde{\boldsymbol{M}}$ (see Line 5 of Algorithm 2) guarantees that with an expected number of $N = \widetilde{O}(nT_{\text{mix}}(\widetilde{\boldsymbol{M}})) = \widetilde{O}\left(n\log(n)/\lambda(\boldsymbol{Q})\right)$ samples, it reaches Line 12 with a sample $y = F_t(1)$ that is generated by the parameterized CFTP algorithm (the iterations in Line 4). We then have to study the *quality of the output sample* y. Crucially, by the utility of the standard CFTP algorithm, the law of y is the normalized measure that is proportional to $\mathcal{D}/\widetilde{\mathcal{D}}$, i.e., $\widetilde{\pi}_0(y) = (\mathcal{D}(y)/\widetilde{\mathcal{D}}(y))/\sum_{x\in[n]} \mathcal{D}(x)/\widetilde{\mathcal{D}}(x)$ (this is the stationary distribution of the modified random walk). Hence, we cannot simply output y. For this reason, we perform rejection sampling. Specifically, as we will see, a single execution of Algorithm 1 either produces a single sample distributed as in $\mathcal{D}$ with some known probability p or fails to output any sample with probability $1-p$ (this is because we do not sample from $\mathcal{D}$ but from an almost uniform distribution). If the rejection sampling process is successful, then the while loop (Line 4) of Algorithm 2 will break and the clean sample will be given as output. One can show that $\Theta(n)$ iterations of the parameterized CFTP algorithm suffice to generate a perfect sample. Hence, the total sample complexity is of order $O(n^2\log(n)\log(n/\delta)/\lambda(\boldsymbol{Q}))$ (at each step $\delta' = \Theta(\delta/n)$). We continue with a sketch behind the rejection sampling process of Line 13 of Algorithm 1.

Claim 1 (Sample Quality and Rejection Sampling)**.** *Algorithm 1, with input the output of the Learning Phase, outputs a state $x \in [n]$ with probability proportional to $\mathcal{D}(x)$ or outputs $\perp$. Moreover, Algorithm 2 outputs a perfect sample from $\mathcal{D}$ with probability 1.*

To see this, set $A = \sum_{y\in[n]} \mathcal{D}(y)/\widetilde{D}(y)$. At the end of Line 11, Algorithm 1 outputs $x \propto \mathcal{D}(x)/\widetilde{D}(x)$, since the unique stationary distribution is $\widetilde{\pi}(x) = (\mathcal{D}(x)/\widetilde{\mathcal{D}}(x))/A$ and the utility of the CFTP guarantees that the outputs state follows the stationary distribution. For the sample x, we perform a rejection sampling process, with acceptance probability $\widetilde{\mathcal{D}}(x)$. Algorithm 1 has $n+1$ potential outputs: It either prints $x \in [n]$ or $\perp$ indicating failure. The arbitrary sample $x \in [n]$ is observed with probability $\widetilde{\pi}(x) \cdot \widetilde{\mathcal{D}}(x) = \mathcal{D}(x)/A$. The remaining probability mass is assigned to $\perp$. We claim that the output of Line 6 of Algorithm 2 has law $\mathcal{D}$. Observe that the whole stochastic process of Algorithm 2 outputs $x \in [n]$ with probability $\sum_{i=0}^{\infty} \Pr[\text{Reject}]^i \Pr[x \text{ is Accepted}] = \mathcal{D}(x)/A \cdot \sum_{i=0}^{\infty} \Pr[\text{Reject}]^i$, where $\Pr[\text{Reject}] = \frac{1}{A}\sum_{y\in[n]} \frac{\mathcal{D}(y)}{\widetilde{\mathcal{D}}(y)}(1-\widetilde{\mathcal{D}}(y)) = 1 - \frac{1}{A} \in (0,1)$. As a result, we get that the probability that x is the output of the above stochastic process is $\mathcal{D}(x)$.

Conclusion. We close with an open question: Is the bound $\widetilde{O}(n^2/\lambda(\boldsymbol{Q}))$ tight? The main difficulty is that it is not clear how to obtain a lower bound against *any possible* exact sampling algorithm. We mention that this work is purely theoretical and has no negative societal impact.

References

[ACCL04] Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Property-preserving data reconstruction. In *International Symposium on Algorithms and Computation*, pages 16–27. Springer, 2004.

[ACK15a] Jayadev Acharya, Clément L. Canonne, and Gautam Kamath. Adaptive Estimation in Weighted Group Testing. In *Proceedings of the 2015 IEEE International Symposium on Information Theory*, ISIT '15, pages 2116–2120. IEEE Computer Society, 2015.

[ACK15b] Jayadev Acharya, Clément L. Canonne, and Gautam Kamath. A Chasm Between Identity and Equivalence Testing with Conditional Queries. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques.*, RANDOM '15, pages 449–466, 2015.

[AJ21] Konrad Anand and Mark Jerrum. Perfect sampling in infinite spin systems via strong spatial mixing. *arXiv preprint arXiv:2106.15992*, 2021.

[APA18] Arpit Agarwal, Prathamesh Patil, and Shivani Agarwal. Accelerated spectral ranking. In *International Conference on Machine Learning*, pages 70–79. PMLR, 2018. `URL: https://proceedings.mlr.press/v80/agarwal18b/agarwal18b.pdf`.

[BC18] Rishiraj Bhattacharyya and Sourav Chakraborty. Property Testing of Joint Distributions using Conditional Samples. *Transactions on Computation Theory*, 10(4):16:1–16:20, 2018.

[BC20] Siddharth Bhandari and Sayantan Chakraborty. Improved bounds for perfect sampling of k-colorings in graphs. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 631–642, 2020.

[BDNP21] Arnab Bhattacharyya, Rathin Desai, Sai Ganesh Nagarajan, and Ioannis Panageas. Efficient statistics for sparse graphical models from truncated samples. In *International Conference on Artificial Intelligence and Statistics*, pages 1450–1458. PMLR, 2021.

[BGJ+10] Arnab Bhattacharyya, Elena Grigorescu, Madhav Jha, Kyomin Jung, Sofya Raskhodnikova, and David P Woodruff. Lower bounds for local monotonicity reconstruction from transitive-closure spanners. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 448–461. Springer, 2010.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of computer and system sciences*, 47(3):549–595, 1993.

[Bre96] Richard Breen. *Regression models: Censored, sample selected, or truncated data*, volume 111. Sage, 1996.

[BT52] R.A. Bradley and M.E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39:324, 1952.

[Can15] Clément L. Canonne. Big Data on the Rise? - Testing Monotonicity of Distributions. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming*, ICALP '15, pages 294–305, 2015.

[Cat12] Manuela Cattelan. Models for paired comparison data: A review with emphasis on dependent data. *Statistical Science*, pages 412–433, 2012.

[CCK+19] Clément L Canonne, Xi Chen, Gautam Kamath, Amit Levi, and Erik Waingarten. Random Restrictions of High-Dimensional Distributions and Uniformity Testing with Subcube Conditioning. *CoRR*, abs/1911.07357, 2019.

[CGR18] Clément L Canonne, Themis Gouleakis, and Ronitt Rubinfeld. Sampling correctors. *SIAM Journal on Computing*, 47(4):1373–1423, 2018.

[Coh16] A. Clifford Cohen. *Truncated and censored samples: theory and applications*. CRC press, 2016.

[DGTZ18] Constantinos Daskalakis, Themis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Efficient Statistics, in High Dimensions, from Truncated Samples. In *59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 639–649. IEEE, 2018.

[DGTZ19] Constantinos Daskalakis, Themis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Computationally and Statistically Efficient Truncated Regression. In *Conference on Learning Theory (COLT)*, pages 955–960, 2019.

[DHKN17] Shaddin Dughmi, Jason D Hartline, Robert Kleinberg, and Rad Niazadeh. Bernoulli factories and black-box reductions in mechanism design. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 158–169, 2017. URL: https://arxiv.org/pdf/1703.04143.pdf.

[DKNS01] Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622, 2001.

[DKTZ21] Constantinos Daskalakis, Vasilis Kontonis, Christos Tzamos, and Emmanouil Zampetakis. A statistical taylor theorem and extrapolation of truncated densities. In *Conference on Learning Theory*, pages 1395–1398. PMLR, 2021.

[DRZ20] Constantinos Daskalakis, Dhruv Rohatgi, and Manolis Zampetakis. Truncated linear regression in high dimensions. *arXiv preprint arXiv:2007.14539*, 2020.

[FGY19] Weiming Feng, Heng Guo, and Yitong Yin. Perfect sampling from spatial mixing. *arXiv preprint arXiv:1907.06033*, 2019.

[FH00] James Allen Fill and Mark Huber. The randomness recycler: a new technique for perfect sampling. In *Proceedings 41st annual symposium on foundations of computer science*, pages 503–511. IEEE, 2000.

[FH10] Johannes Fürnkranz and Eyke Hüllermeier. Preference learning and ranking by pairwise comparison. In *Preference learning*, pages 65–82. Springer, 2010.

[FJO+15] Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, Venkatadheeraj Pichapati, and Ananda Theertha Suresh. Faster Algorithms for Testing under Conditional Sampling. In *Proceedings of the 28th Annual Conference on Learning Theory*, COLT '15, pages 607–636, 2015.

[FKKT21] Dimitris Fotakis, Alkis Kalavasis, Vasilis Kontonis, and Christos Tzamos. Efficient algorithms for learning from coarse labels. In *Conference on Learning Theory*, pages 2060–2079. PMLR, 2021.

[FKT20] Dimitris Fotakis, Alkis Kalavasis, and Christos Tzamos. Efficient parameter estimation of truncated boolean product distributions. In *Conference on Learning Theory*, pages 1586–1600. PMLR, 2020.

[FV86] Michael A Fligner and Joseph S Verducci. Distance Based Ranking Models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 359–369, 1986.

[Gal97] Francis Galton. An examination into the registered speeds of American trotting horses, with remarks on their value as hereditary data. *Proceedings of the Royal Society of London*, 62(379-387):310–315, 1897.

[GJL19] Heng Guo, Mark Jerrum, and Jingcheng Liu. Uniform sampling through the lovász local lemma. 66(3), 2019.

[GLR18a] Rong Ge, Holden Lee, and Andrej Risteski. Beyond log-concavity: provable guarantees for sampling multi-modal distributions using simulated tempering langevin monte carlo. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7858–7867, 2018.

[GLR18b] Rong Ge, Holden Lee, and Andrej Risteski. Simulated tempering langevin monte carlo ii: An improved proof using soft markov chain decomposition. *arXiv preprint arXiv:1812.00793*, 2018.

[GLT20] Evangelia Gergatsouli, Brendan Lucier, and Christos Tzamos. Black-box methods for restoring monotonicity. In *International Conference on Machine Learning*, pages 3463–3473. PMLR, 2020.

[GM99] Peter J Green and Duncan J Murdoch. Exact sampling for bayesian inference: towards general purpose algorithms. *Bayesian statistics*, 6:301–321, 1999.

[GTZ17] Themistoklis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Faster Sublinear Algorithms Using Conditional Sampling. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17, pages 1743–1757. SIAM, 2017.

[HJ12] Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[HOX14] Bruce Hajek, Sewoong Oh, and Jiaming Xu. Minimax-optimal inference from partial rankings. *Advances in Neural Information Processing Systems*, 27:1475–1483, 2014.

[HS00] Olle Häggström and Jeffrey E Steif. Propp–wilson algorithms and finitary codings for high noise markov random fields. *Combinatorics, Probability and Computing*, 9(5):425–439, 2000.

[HSW21] Kun He, Xiaoming Sun, and Kewen Wu. Perfect sampling for (atomic) lov\'asz local lemma. *arXiv preprint arXiv:2107.03932*, 2021.

[Hub98] Mark Huber. Exact sampling and approximate counting techniques. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 31–40, 1998.

[Hub04] Mark Huber. Perfect sampling using bounding chains. *The Annals of Applied Probability*, 14(2):734–753, 2004.

[Hub16] Mark L. Huber. *Perfect simulation*, volume 148. CRC Press, 2016.

[Hun04] David R. Hunter. Mm algorithms for generalized bradley-terry models. *The annals of statistics*, 32(1):384–406, 2004.

[HW71] David Lee Hanson and Farroll Tim Wright. A bound on tail probabilities for quadratic forms in independent random variables. *The Annals of Mathematical Statistics*, 42(3):1079–1083, 1971.

[IZD20] Andrew Ilyas, Emmanouil Zampetakis, and Constantinos Daskalakis. A theoretical and practical framework for regression and classification from truncated samples. In *International Conference on Artificial Intelligence and Statistics*, pages 4463–4473. PMLR, 2020.

[Jer03] Mark Jerrum. *Counting, sampling and integrating: algorithms and complexity*. Springer Science & Business Media, 2003.

[JLYY11] Xiaoye Jiang, Lek-Heng Lim, Yuan Yao, and Yinyu Ye. Statistical ranking and combinatorial hodge theory. *Mathematical Programming*, 127(1):203–244, 2011.

[JR13] Madhav Jha and Sofya Raskhodnikova. Testing and reconstruction of lipschitz functions with applications to data privacy. *SIAM Journal on Computing*, 42(2):700–731, 2013.

[JSS21] Vishesh Jain, Ashwin Sah, and Mehtaab Sawhney. Perfectly sampling $k \geq (8/3+o(1))\delta$-colorings in graphs. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1589–1600, 2021.

[JSV04] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM (JACM)*, 51(4):671–697, 2004.

[Kaz11] Gabriella Kazai. In search of quality in crowdsourcing for search engine evaluation. In *European conference on information retrieval*, pages 165–176. Springer, 2011.

[Ken05] Wilfrid S. Kendall. Notes on perfect simulation. *Markov chain Monte Carlo: innovations and applications*, 7, 2005.

[KT19] Gautam Kamath and Christos Tzamos. Anaconda: A Non-Adaptive Conditional Sampling Algorithm for Distribution Testing. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 679–693. SIAM, 2019.

[KTZ19] Vasilis Kontonis, Christos Tzamos, and Manolis Zampetakis. Efficient Truncated Statistics with Unknown Truncation. In *260th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1578–1595. IEEE, 2019.

[LOAF12] Miguel Angel Luengo-Oroz, Asier Arranz, and John Frean. Crowdsourcing malaria parasite quantification: an online game for analyzing images of infected thick blood smears. *Journal of medical Internet research*, 14(6):e2338, 2012.

[LP17] David A. Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017. `URL: https://www.stat.berkeley.edu/users/aldous/260-FMIE/Levin-Peres-Wilmer.pdf`.

[LSR21] Wanshan Li, Shamindra Shrotriya, and Alessandro Rinaldo. The performance of the mle in the bradley-terry-luce model in ℓ_∞-loss and under general graph topologies. *arXiv preprint arXiv:2110.10825*, 2021. `URL: https://arxiv.org/abs/2110.10825`.

[Luc59] R.D. Luce. *Individual Choice Behavior*. Wiley, 1959.

[Mar96] John I. Marden. *Analyzing and modeling rank data*. CRC Press, 1996.

[MMS21] Ankur Moitra, Elchanan Mossel, and Colin P Sandon. Learning to sample from censored markov random fields. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 3419–3451. PMLR, 15–19 Aug 2021.

[MP92] Enzo Marinari and Giorgio Parisi. Simulated tempering: a new monte carlo scheme. *EPL (Europhysics Letters)*, 19(6):451, 1992.

[MP05] Elchanan Mossel and Yuval Peres. New coins from old: computing with unknown bias. *Combinatorica*, 25(6):707–724, 2005.

[NOS12] Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pair-wise comparisons. In *Advances in neural information processing systems*, pages 2474–2482, 2012.

[NOS17] Sahand Negahban, Sewoong Oh, and Devavrat Shah. Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 65(1):266–287, 2017.

[NP05] Şerban Nacu and Yuval Peres. Fast simulation of new coins from old. *Annals of Applied Probability*, 15(1A):93–115, 2005.

[NP19] Sai Ganesh Nagarajan and Ioannis Panageas. On the Analysis of EM for truncated mixtures of two Gaussians. In *31st International Conference on Algorithmic Learning Theory (ALT)*, pages 955–960, 2019.

[Ple21] Orestis Plevrakis. Learning from censored and dependent data: The case of linear dynamics. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 3771–3787. PMLR, 15–19 Aug 2021.

[PW96] James Gary Propp and David Bruce Wilson. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures & Algorithms*, 9(1-2):223–252, 1996. `URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.3607&rep=rep1&type=pdf`.

[PW98] James Gary Propp and David Bruce Wilson. How to get a perfectly random sample from a generic markov chain and generate a random spanning tree of a directed graph. *Journal of Algorithms*, 27(2):170–217, 1998.

[RA14] Arun Rajkumar and Shivani Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *International Conference on Machine Learning*, pages 118–126, 2014.

[RV13] Mark Rudelson and Roman Vershynin. Hanson-wright inequality and sub-gaussian concentration. *Electronic Communications in Probability*, 18, 2013.

[SBB+16] Nihar B Shah, Sivaraman Balakrishnan, Joseph Bradley, Abhay Parekh, Kannan Ramchandran, and Martin J Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *The Journal of Machine Learning Research*, 17(1):2049–2095, 2016. URL: https://jmlr.org/papers/volume17/15-189/15-189.pdf.

[SBGW16] Nihar Shah, Sivaraman Balakrishnan, Aditya Guntuboyina, and Martin Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *International Conference on Machine Learning*, pages 11–20. PMLR, 2016.

[SS10] Michael Saks and Comandur Seshadhri. Local monotonicity reconstruction. *SIAM Journal on Computing*, 39(7):2897–2926, 2010.

[Tao12] Terence Tao. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012. URL: http://www.stat.ucla.edu/~ywu/research/documents/BOOKS/TaoRandomMatrix.pdf.

[Thu27] Louis L Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

[Tro15] Joel A Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015. URL: https://arxiv.org/pdf/1501.01571.pdf.

[VY16] Milan Vojnovic and Seyoung Yun. Parameter estimation for generalized thurstone choice models. In *International Conference on Machine Learning*, pages 498–506, 2016.

[VYZ20] Milan Vojnovic, Se-Young Yun, and Kaifang Zhou. Convergence rates of gradient descent and mm algorithms for bradley-terry models. In *International Conference on Artificial Intelligence and Statistics*, pages 1254–1264. PMLR, 2020. URL: https://arxiv.org/abs/1901.00150.

[Wil00] David Bruce Wilson. How to couple from the past using a read-once source of randomness. *Random Structures & Algorithms*, 16(1):85–113, 2000.

[WJJ13] Fabian Wauthier, Michael Jordan, and Nebojsa Jojic. Efficient ranking from pairwise comparisons. In *International Conference on Machine Learning*, pages 109–117. PMLR, 2013.

[ZCKM20] Tom Zahavy, Alon Cohen, Haim Kaplan, and Yishay Mansour. Unknown mixing times in apprenticeship and reinforcement learning. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 430–439. PMLR, 03–06 Aug 2020. URL: https://proceedings.mlr.press/v124/zahavy20a.html.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We have presented the assumptions of our models in Section 1. We have also explained future research directions that will further improve this work in the Conclusion paragraph at the end of the main body.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] We have discussed that in the Conclusion paragraph at the end of the main body.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] We have specified the theoretical models we are working on, see the Problem Formulation in Section 1.

(b) Did you include complete proofs of all theoretical results? [Yes] Due to space constraints we have included the full proofs of the results in the Appendix.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Additional Notation and Preliminaries

A.1 Notation

In this section, we provide the basic notation we are going to use in the technical part.

General Notation. We define $[n] := \{1, \dots, n\}$. We denote vectors with small bold letters $\boldsymbol{x}$ and matrices with bold letters $\boldsymbol{P} = [\boldsymbol{P}_{xy}]$. For a vector $\boldsymbol{x}$, we let $x_i = x(i)$ be its i-th entry. For $p \in \{1, 2, \infty\}$, we denote the L_p norm by $\|.\|_p$. For a matrix $\boldsymbol{A}$, let $\|\boldsymbol{A}\|_2$ denote its spectral norm (largest singular value). We consider graphs $G = (V, E)$, with $|V| = n$ vertices, whose associated Laplacian matrix is denoted $\boldsymbol{\Lambda}(G)$. The edge set of the graph G is the set $E(G)$. In Appendix A.2.2, we have included some basic definitions and facts about random walks and Markov chains. We define the weighted Laplacian matrix of $G_{\mathcal{Q}}$ for a distribution $\mathcal{Q}$ as

$$\boldsymbol{Q}_{xy} = -\mathcal{Q}(x, y) \text{ for any } x \neq y \text{ and } \boldsymbol{Q}\boldsymbol{1} = \boldsymbol{0}\,, \tag{2}$$

Distributions and Distances. The probability simplex is denoted by Δ^n. We let $\mathrm{supp}(\mathcal{D})$ denote the support of a distribution $\mathcal{D}$. The total variation distance between two discrete distributions $\mathcal{D}_1, \mathcal{D}_2$ over $[n]$ is equal to $\mathrm{TV}(\mathcal{D}_1, \mathcal{D}_2) = \max_{S \subseteq [n]}(\mathcal{D}_1(S) - D_2(S)) = \frac{1}{2}\|\mathcal{D}_1 - \mathcal{D}_2\|_1$. We let $\mathcal{D}(S) = \sum_{x \in S} \mathcal{D}(x)$ and $\mathcal{D}_S$ is the conditional distribution on the set S, i.e., $\mathcal{D}_S(x) = \frac{\mathcal{D}(x)}{\mathcal{D}(S)}\mathbf{1}\{x \in S\}$. The vector $\boldsymbol{z} \in \mathbb{R}^n$ with $z_i = \log(\mathcal{D}(i))$ is called the natural parameter vector of $\mathcal{D}$.

Random Walks. For a reversible transition matrix $\boldsymbol{P}$, let its (real) spectrum be $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -1$. We define the *absolute spectral gap* of $\boldsymbol{P}$ to be the difference $\Gamma(\boldsymbol{P}) = 1 - \max\{\lambda_2(\boldsymbol{P}), |\lambda_n(\boldsymbol{P})|\}$. If $\boldsymbol{P}$ is aperiodic and irreducible, then $\Gamma(\boldsymbol{P}) > 0$. One could also define the *spectral gap* $1 - \lambda_2(\boldsymbol{P})$. The two gaps are equal when the chain is lazy, i.e., when $\forall i \in [n]$, $\boldsymbol{P}_{ii} \geq 1/2$. We let $\lambda(\boldsymbol{L})$ be the second *smallest* eigenvalue of a Laplacian matrix $\boldsymbol{L}$, a.k.a., its Fiedler eigenvalue. The Laplacian matrix $\boldsymbol{L} \in \mathbb{R}^{n \times n}$ is positive semi-definite and induces a semi-norm on $\mathbb{R}^n$ with $\|\boldsymbol{v}\|_L := \sqrt{\boldsymbol{v}^T \boldsymbol{L} \boldsymbol{v}}$ for $\boldsymbol{v} \in \mathbb{R}^n$. Recall that a semi-norm differs from a norm in that the semi-norm of a non-zero element is allowed to be zero. In Algorithm 4, Algorithm 1 and Algorithm 7, we use the notation $F_t(x) \leftarrow F_{t+1}(w)$ for some $t < 0$. This means that we append the path of x at time t by adding the transition $x \to w$ and then the path of w from time $t + 1$ up to 0.

Matrix Operators. We denote by $\boldsymbol{A} \odot \boldsymbol{B} = (\boldsymbol{A}_{ij}\boldsymbol{B}_{ij})_{1 \leq i \leq j \leq n}$ the standard Hadamard matrix product and by $\boldsymbol{A} \circ \boldsymbol{B}$ a variation of the Hadamard matrix product, where the off-diagonal entries are equal to those of the standard Hadamard product $\boldsymbol{A} \odot \boldsymbol{B}$, but the diagonal entries correspond to the diagonal matrix with entries $(-\sum_{j \neq i} \boldsymbol{A}_{ij}\boldsymbol{B}_{ij})_{i \in [n]}$. Finally, we let $\boldsymbol{A} \otimes \boldsymbol{B} = \boldsymbol{I} - \boldsymbol{A} \circ \boldsymbol{B}$.

A.2 Preliminaries

In this section, we provide some preliminaries and some useful tools about (i) concentration of random matrices, (ii) random walks and (iii) CFTP. The reader may skip this section.

A.2.1 Random Matrices

We continue with some definitions for random matrices, needed for the proof of Lemma 17. The following can be found at [Tro15].

Definition 6. *Let $(\Omega, \mathcal{F}, \mu)$ be a probability space. A random matrix $\boldsymbol{X}$ is a measurable map $\boldsymbol{X} : \Omega \to \mathcal{M}^{n_1 \times n_2}$.*

The entries of $\boldsymbol{X}$ may be considered complex random variables that may or may not be correlated with each other. A finite sequence $\{\boldsymbol{X}_i\}$ of random matrices is independent when

$$\mu(\boldsymbol{X}_k \in F_k \text{ for each } k) = \prod_k \mu(\boldsymbol{X}_k \in F_k)\,,$$

for any collection $\{F_k\}$ of Borel subsets of $\mathcal{M}^{n_1 \times n_2}$.

Proposition 7 (Hermitian Matrix Chernoff Bounds (see [Tro15])). *Consider a finite sequence $\{\boldsymbol{X}_i\}$ of m independent, random, Hermitian square matrices with common dimension n. Assume that $0 \leq \lambda_{\min}(\boldsymbol{X}_i) \leq \lambda_{\max}(\boldsymbol{X}_i) \leq M$ for any $i \in [m]$. Let $\boldsymbol{Y} = \sum_i X_i$. Then, for $\epsilon \in [0, 1)$:*

$$\Pr\left[\lambda_{\min}(\boldsymbol{Y}) \leq (1-\epsilon)\lambda_{\min}(\mathbb{E}\boldsymbol{Y})\right] \leq n(e^{-\epsilon}/(1-\epsilon)^{1-\epsilon})^{\lambda_{\min}(\mathbb{E}\boldsymbol{Y})/M} \leq n\exp\left(-\epsilon^2\lambda_{\min}(\mathbb{E}\boldsymbol{Y})/(2M)\right).$$

A.2.2 Random Walks and Markov Chains

This section is mostly based on [LP17] and we refer the reader to [LP17] for a thorough exposition.

Markov Chains. Let Ω be a finite state space. A Markov chain is a sequence of random variables $X_0, X1, \ldots$ that satisfy the Markov property, i.e.,

$$\Pr[X_{t+1} = x | X_0 = x_0, \ldots, X_t = x_t] = \Pr[X_{t+1} = x | X_t = x_t].$$

A Markov chain is called time-homogeneous, if the RHS of the above equation does not depend of t. Such a chain is associated with a transition matrix $\boldsymbol{P} = \{\boldsymbol{P}(x, y)\}$, where $(x, y) \in \Omega \times \Omega$. It holds that

$$\boldsymbol{P}(x, y) = \Pr[X_{t+1} = y | X_t = x] \text{ for all } x, y \in \Omega, t \in \mathbb{N}.$$

A Markov chain is *ergodic* if there exists a time $t \in \mathbb{N}$ such that $\boldsymbol{P}^{(t)}(x, y) > 0$ for any $x, y \in \Omega$, i.e., there exists a finite time t so that the probability of going from any vertex to any other in t steps is positive. For finite state Markov chains, ergodicity is equivalent to irreducibility and aperiodicity. A Markov chain is *irreducible* if for any two states $x, y \in \Omega$, there exists a time step t such that $\boldsymbol{P}^{(t)}(x, y) > 0$ and is *aperiodic* if, for any state x, it holds that $\gcd\{t | \boldsymbol{P}^{(t)}(x, x) > 0\} = 1$. Observe that a Markov chain with self-loops is aperiodic.

Stationary Distribution. A stationary distribution $\boldsymbol{\pi} \in \Delta^n$, i.e., the $(n-1)$-dimensional simplex whose vertices are the n standard unit vectors, is defined as the fixed point of the transition matrix $\boldsymbol{P}$, that is $\boldsymbol{\pi}^T\boldsymbol{P} = \boldsymbol{\pi}^T$. An ergodic Markov chain has a *unique stationary distribution* $\boldsymbol{\pi}$ and, as t increases, it converges to it. An interesting property of Markov chains is the reversibility. A Markov chain is *reversible* if there exists a distribution $\boldsymbol{\pi}$ that satisfies the detailed balanced equations:

$$\boldsymbol{\pi}(x)\boldsymbol{P}(x, y) = \boldsymbol{\pi}(y)\boldsymbol{P}(y, x) \text{ for all } x, y \in \Omega. \tag{3}$$

In this case, we can verify that $\boldsymbol{\pi}$ is a stationary distribution. We can simply write π for the stationary distribution with associated probability vector $\boldsymbol{\pi}$.

Mixing Time. It is important to understand the convergence time of a Markov chain to its stationary distribution π. A crucial random variable for convergence time is the *mixing time*.

Definition 8 (Mixing Time). *Let $0 < \epsilon < 1/2$. Let M be an ergodic Markov chain on a finite state space Ω with stationary distribution π. Then, the mixing time with accuracy ϵ of M equals:*

$$T_{\text{mix}}(\boldsymbol{P}; \epsilon) = \inf\{t > 0 : \max_{x \in \Omega} \text{TV}(\boldsymbol{P}_x(X_t), \pi) \leq \epsilon\},$$

where $\boldsymbol{P}_x(X_t)$ is the distribution of the state X_t at time t for starting state $x \in \Omega$.

For a reversible transition matrix $\boldsymbol{P}$, let its spectrum be:

$$1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -1.$$

Note that $\lambda_2 < 1$ if and only if the chain is irreducible (exactly one connected component) and $\lambda_n > -1$ if and only if the chain is aperiodic (e.g., not a bipartite graph). We define the *absolute spectral gap* of $\boldsymbol{P}$ to be the difference: $\Gamma(\boldsymbol{P}) = 1 - \max\{|\lambda_2(\boldsymbol{P})|, |\lambda_n(\boldsymbol{P})|\}$. It holds that if $\boldsymbol{P}$ is aperiodic and irreducible, then $\Gamma(\boldsymbol{P})$ is strictly positive. One could also define the spectral gap $\lambda(\boldsymbol{P}) = 1 - \lambda_2(\boldsymbol{P})$. The two gaps are equal when the chain is lazy, i.e., when for any state $x \in \Omega$, it holds that $\boldsymbol{P}(x, x) \geq 1/2$.

Lemma 9 (Bounding T_{mix}, see [LP17]). *Let $0 < \epsilon < 1$. Assume that the transition matrix $\boldsymbol{P}$ is aperiodic, irreducible and reversible with respect to π. Then, it holds that:*

$$(t_{\text{rel}} - 1)\log\left(\frac{1}{2\epsilon}\right) \leq T_{\text{mix}}(\boldsymbol{P}; \epsilon) \leq \log\left(\frac{1}{\epsilon\pi_{\min}}\right)t_{\text{rel}},$$

where $t_{\text{rel}} = 1/\Gamma(\boldsymbol{P})$ is the relaxation time of the Markov chain, i.e., the inverse absolute spectral gap.

Coalescencing Random Walks. Let $\Omega = [n]$. In a coalescing random walk, a set of n particles perform independent discrete time random walks on an undirected connected graph $\mathcal{G} = (V, E)$ with $|V| = n$, with each particle initially placed at a single (distinct) vertex $x \in V$. In each time step, all particles move simultaneously. Whenever two or more particles meet at a vertex, they unite to form a single particle which then continues to make a random walk through the graph. The *coalescence time* T_{coal} is a random variable and is the first time when all particles coalesce. More formally, we can define *coalescence* as follows.

Definition 10 (Coalescence of Stochastic processes (see [Hub16]))**.** *Let $\mathcal{A}$ be a collection of stochastic processes, defined over a common index set $\mathcal{I}$ and common state space Ω. If there is an index $i \in \mathcal{I}$ and state $x \in \Omega$ such that, for all stochastic processes $X \in \mathcal{A}$, it holds that $X_i = x$, then we say that the stochastic processes have coalesced.*

Coalescence Time of Random Walks. Consider the random variable $X_t^{(i)}$, that indicates the position of the i-th particle at time t. The coalescence time is equal to:

$$T_{\text{coal}} = \inf_{t>0} \left\{ X_t^{(i)} = X_t^{(j)} \text{ for any } i \neq j, i, j \in [n] \right\}.$$

We conclude this section with a folklore result, that deals with the possible locations in the complex plane of the eigenvalues of a square matrix $\boldsymbol{A} \in \mathcal{M}_n$.

Lemma 11 (Geršgorin's Theorem (see [HJ12]))**.** *Let $\boldsymbol{A} = [\boldsymbol{A}_{ij}] \in \mathcal{M}_n$. For any $i \in [n]$, let*

$$R_i(\boldsymbol{A}) = \sum_{j \neq i} |\boldsymbol{A}_{ij}|,$$

and consider the n Geršgorin disks

$$B_i = \{z \in \mathbb{C} : |z - \boldsymbol{A}_{ii}| \leq R_i(\boldsymbol{A})\}.$$

Then, the eigenvalues of $\boldsymbol{A}$ are in the union of the Geršgorin disks, i.e.,

$$\text{spec}(\boldsymbol{A}) \subseteq \bigcup_{i=1}^{n} B_i.$$

A.2.3 Exact Sampling and CFTP Preliminaries

Markov chain Monte Carlo (MCMC) methods constitute a class of algorithms for sampling from probability measures and arise naturally in various fields of science such as theoretical computer science (e.g., approximation algorithms for #P-complete problems (see [Jer03, JSV04])), statistical physics (e.g., in order to understand phase transition phenomena for Ising models) and statistics. However, the theory of MCMC, in terms of exact sampling, can be seen as an asymptotic analysis (i.e., in the limit, the total variation distance vanishes). Perfect simulation is analogous to MCMC and deals with techniques for designing algorithms that return exact draws from the target distribution, instead of long-time approximations. Exact sampling comprises a well-studied field (see [Ken05, Hub16]) and has numerous applications in computer science (e.g., in approximate counting [Hub98]). The importance of exact simulation gave rise to various procedures in order to generate perfect samples. A small sample of this line of research ([PW96, GM99, Wil00, HS00, FH00, Hub04]).

Coupling from the past (CFTP) is a technique developed by Propp and Wilson ([PW96, PW98]), that provides an exact random sample from a Markov chain, that has law the (unique) stationary distribution. The algorithm assumes that a particle has been running on the Markov chain since time $-\infty$ (arbitrarily long in the past) and we are concerned with the location of the particle at (the fixed) time $t = 0$. The fact that the stopping time is deterministic and not a random variable (as in MCMC) is crucial. Since the particle performs a random walk in the Markov chain infinitely long, one would intuitively believe that the particle at time $t = 0$ is distributed according to the stationary distribution.

We define $F_{(t,0]}$ as the t-step evolution of the Markov chain from time $t \in \mathbb{Z}_{\leq 0}$ to the fixed time $T = 0$ (evolution *from the past*). Note that we can decompose the evolution of the chain into t independent applications of the random function f, which encodes the information of the random walk, i.e., $\Pr[f(x) = y] = \boldsymbol{P}_{xy}$ for any pair of states $x, y \in \Omega$. Hence, we have that

$$F_{(t,0]} = f_{-1} \circ f_{-2} \circ \ldots \circ f_t.$$

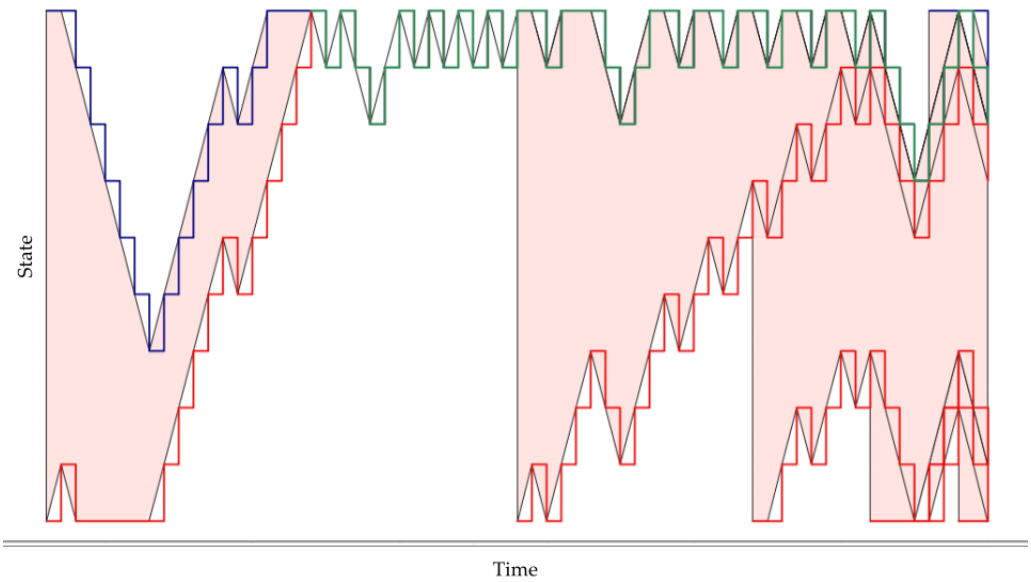

Figure 2: Execution of the CFTP Algorithm for a random walk on the path graph. Observe that it suffices to execute the random procedure only for the two extreme points, since their trajectories dominate any intermediate point of the path. After a series of executions which did not coalesce (see the right part of Figure 2), we observe that the two stochastic evolutions, the blue and the red one in the left part of Figure 2, coalesce and continue as a single trajectory until $t = 0$ (see the green trajectory). A sample execution code can be found at `https://warwick.ac.uk/fac/sci/statistics/staff/academic-research/kendall/personal/perfect_programs/`

Algorithm 3 Coupling From The Past Algorithm

procedure COUPLINGFROMTHEPAST() ▷ *Assuming (adaptive) access to the Markov oracle.*
  $t \leftarrow 0$
  $F_{(t,0]}(i) \leftarrow i,$ for any $i \in [n]$
  while $F_{(t,0]}$ has not coallesced **do**
    $t \leftarrow t - 1$
    $f_t \leftarrow \mathrm{Markov}()$
    $F_{(t,0]} \leftarrow F_{(t+1,0]} \circ f_t$
  end
  return $F_{(t,0]}(1)$
end procedure

We can think of the data structure $F_{(t,0]}$ as a list of n values, each one capturing the evolution of the Markov chain, initiated in a different state at time t and stopped at time 0. Similarly, the oracle call $\mathrm{Markov}()$ corresponds to n adaptive calls, each one giving a transition from a fixed state $x \in \Omega$, where the next state is distributed according to $\boldsymbol{P}(x, \cdot)$. Hence, f_t is an n-dimensional vector with values in Ω. We refer the reader to Figure 2 for an execution of the CFTP algorithm.

The following fact is the main result behind CFTP.

Fact 1 ([PW96])**.** *With probability* 1*, the Coupling from the Past algorithm returns a value, and this value is distributed according to the stationary distribution of the Markov chain.*

The following proposition provides an upper bound on the convergence of the CFTP algorithm.

Proposition 12 ([ZCKM20])**.** *Let* π *be the stationary distribution of an ergodic Markov chain* $\boldsymbol{P}$ *with* n *states. We run* n *simulations of the chain each starting at a different state. When two or more simulations coalesce, we merge them into a single simulation. With probability at least* $1 - \delta$*, all* n *chains have merged after at most* $O(nT_{\mathrm{mix}}(\boldsymbol{P}; 1/4)\log(1/\delta))$ *iterations.*

B The Proof of Theorem 2 (Direct Exact Sampling from LSS)

In this section, we provide the proof of Theorem 2. Let us begin with the algorithm. We remark that in the proof we will use the more technical notation $F_{(a,b]}$ to denote the evolution of the chain from a to b. In Algorithm 4, this is abbreviated as F_t to denote $F_{(t,0]}$.

Algorithm 4 Exact Sampling from Local Sampling Schemes of Informal Theorem 1 & Theorem 2

procedure EXACTSAMPLER() ▷ *Sample access to the LSS oracle* $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$.
  $t \leftarrow 0$
  $F_0(x) \leftarrow x$, for any $x \in [n]$
  while F_t has not coallesced **do** ▷ *While no coalescence has occured.*
    $t \leftarrow t-1$
    Draw sample (i,j,w) with $(i,j) \in \mathcal{E}$ and $w \in \{i,j\}$
    for $x = 1 \dots n$ **do** ▷ *In order to update state* x.
      if $x \notin \{i,j\}$ or $x = w$ **then** $F_t(x) \leftarrow F_{t+1}(x)$ ▷ *Append* F_t *in the past.*
      else $F_t(x) \leftarrow F_{t+1}(w)$ ▷ *New transition with probability* $\mathcal{Q}(x,w)\frac{\mathcal{D}(w)}{\mathcal{D}(x)+\mathcal{D}(w)}$.
    end
  end
  Output $F_t(1)$ ▷ *Output the perfect sample.*
end procedure

Proof. At each iteration, the algorithm draws a single sample from $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ Then, the algorithm performs the following procedure for each state $v \in [n]$: let the drawn sample be (i,j,w), where $(i,j) \in \mathcal{E}$ and $w \in \{0,1\}$ indicating the winning node between i and j, i.e., the node i stays in i (if $w = 1$) or node i moves to j (if $w = 0$). For any $v \in [n]$, the algorithm checks whether $v \in \{i,j\}$. If not, the state v remains at v; otherwise, assume that $i = v$. Then, the transition $v \to j$ is implemented only if $w = 1$, otherwise it remains at v. We claim that this process simulates the transitions of the matrix $\boldsymbol{M}$ and has unique stationary distribution $\mathcal{D}$. The probability that the transition $v \to j$ occurs is

$$\begin{aligned}\Pr[v \to j] &= \Pr[(v,j,w) \text{ is drawn and } w = 0] = \Pr[(v,j) \text{ is drawn}] \cdot \Pr[w = 0 | (v,j) \text{ is drawn}] \\ &= \mathcal{Q}(v,j) \cdot \frac{\mathcal{D}(j)}{\mathcal{D}(v)+\mathcal{D}(j)} = \boldsymbol{M}_{vj}\,.\end{aligned}$$

Hence, for $v \in [n]$ and by the above process, the transition probabilities are given by the matrix $\boldsymbol{M}$, that is a transition matrix with unique stationary distribution $\mathcal{D}$. We will see that this random process gives a perfect sample.

Consider the state space $\Omega = [n]$ and let T be a positive integer. Let $(X_t)_{t\in\mathbb{Z}}$ be a random walk on $[n]$ with transition probabilities as in $\boldsymbol{M}$ (see Equation (1)). Let $F_{(a,b]} : [n] \to [n]$ be such that $F_{(-t_0,T]}(x_0)$ denotes the state of the walk at time $t = T$, i.e., $X_T \in [n]$, when the random walk begun at time $t = -t_0$ (i.e., *in the past*) with initial state $X_{-t_0} = x_0$. Specifically, for each t_0, T, we have that $F_{(-t_0,T]} : [n] \to [n]$ is a random map and $F_{(-t_0,T]}(x_0)$ is a realization of the Markov chain, with initial state x_0, executed from time $-t_0$ to time T. We call $F_{(-t_0,T]}$ a *stochastic flow*.

The CFTP method is initialized at time $t = -t_0$ into the past and places n particles (i.e., simulates n concurrent random walks), one at each state $x \in [n]$. CFTP terminates when the stochastic flow $F_{(-t_0,0]}$ is constant (it has a single point random image, i.e., $F_{(-t_0,0]}(x) = F_{(-t_0,0]}(y)$, for all x, y). Namely, CFTP terminates when all the random walks that start at time $t = -t_0$ have coalesced at time $t = 0$ at a single $z \in [n]$. Then, the algorithm outputs the (common) state $z = F_{(-t_0,0]}(x)$, for any $x \in [n]$.

Since the transitions of each random walk are performed with respect to $\boldsymbol{M}$, the resulting sample z is distributed according to the stationary distribution $\mathcal{D}$. This can be shown as follows (see also [PW96, Hub16]): Let $\mathcal{L}(X)$ denote the distribution of the random variable X. Since the chain is aperiodic and irreducible, coalescence occurs almost surely. Specifically, assume that coalescence occurs when the simulation starts at a finite time $t = -t_0$. First, observe that if coalescence occurs at time $-t_0$, then if CFTP starts at any time $-t < -t_0$, we end up in the same state (because the randomness in $[-t_0, 0]$ remains the same), i.e., $F_{(-t,0]}(x) = F_{(-t_0,0]}(x)$ for any $-t < -t_0$ and any

state $x \in [n]$. Since, coalescence has occurred, we can work with a fixed starting state x_0 and let $F_{(-t,0]} = F_{(-t,0]}(x_0)$. Hence, it holds that $\mathcal{L}(F_{(-t,0]}) = \mathcal{L}(F_{(-t_0,0]})$ and so

$$\mathrm{TV}(\mathcal{L}(F_{(-t_0,0]}), \mathcal{D}) = \lim_{t\to\infty} \mathrm{TV}(\mathcal{L}(F_{(-t,0]}), \mathcal{D}) \,.$$

We note that the distribution of the time-backward coalescing random walk is equal to the distribution of the time-forward random walk, i.e., $\mathcal{L}(F_{(-t,0]}) = \mathcal{L}(F_{(0,t]})$. Thus, $\mathrm{TV}(\mathcal{L}(F_{(-t_0,0]}), \mathcal{D}) = \lim_{t\to\infty} \mathrm{TV}(\mathcal{L}(F_{(0,t]}), \mathcal{D}) = 0$. So, the output sample of the CFTP algorithm is distributed according to $\mathcal{D}$. Using Proposition 12, we get that, with high probability, coalescence occurs after $O(nT_{\mathrm{mix}}(\boldsymbol{M}; 1/4))$ iterations. From Lemma 9 with $\epsilon = 1/4$, we get that the mixing time of the Markov chain is $O(\log(1/\mathcal{D}_{\min})/\Gamma(\boldsymbol{M}))$. Finally, note that in order to perform a single iteration of the CFTP method, it requires to draw a single sample from the Local Sampling Scheme oracle. This concludes the analysis of the sample complexity. The time complexity is also polynomial in the number of samples [PW96]. □

Remark 1. *As we mentioned in the discussion after Informal Theorem 1, in order to obtain Theorem 2 only Assumption 1 is required. Under only this assumption, a quite large amount of Local Sampling Schemes is allowed and the resulting sample complexity is* $\widetilde{O}\left(\frac{n\log(1/\mathcal{D}_{\min})}{\Gamma(\boldsymbol{M})}\right)$. *However, if we additionally assume that Assumption 2 holds, as we did in the statement of Informal Theorem 1, we accept only LSSs whose target distribution satisfies* $\mathcal{D}_{\min} = \Omega(2^{-n})$. *This yields the (non-tight) bound* $\widetilde{O}(n^2/\Gamma(\boldsymbol{M}))$ *of Informal Theorem 1. We preferred to omit this detail in Informal Theorem 1 and emphasize only on the exponential dependence due to* $\Gamma(\boldsymbol{M})$ *for many natural instances.*

C The Proof of Theorem 3 (Exact Sampling from LSS using Learning)

The main result of this section is the convergence analysis of the Coupling From the Past algorithm when the algorithm performs the rescaling transformation internally in each transition.

Theorem 13 (Convergence of CFTP)**.** *Assume that Assumption 1 and Assumption 2 both hold. Algorithm 1, assuming access to a Local Sampling Scheme* $\mathrm{Samp}(\mathcal{Q}; \mathcal{D})$ *with weighted Laplacian matrix* $\boldsymbol{Q}$ *(see Equation* (2)*) associated with the pair distribution* $\mathcal{Q}$ *over* $[n] \times [n]$*, satisfies the following:*

- (i) *Algorithm 1 reaches Line 12 with probability* 1.
- (ii) *For any* $\delta \in (0,1)$*, a single execution of of Algorithm 1 uses* $\widetilde{O}\left(\frac{n\log(n)\log(1/\delta)}{\lambda(\boldsymbol{Q})}\right)$ *samples with probability* $1-\delta$ *and the running time is polynomial in the number of samples.*
- (iii) *Algorithm 1, with input the output of the Learning Phase, outputs a state* $x \in [n]$ *with probability proportional to* $\mathcal{D}(x)$ *or outputs* $\perp$*. Moreover, Algorithm 2 outputs a perfect sample from* $\mathcal{D}$*.*

Using the above properties, for any $\delta > 0$*, Algorithm 2 draws, with probability at least* $1-\delta$*,* $N = O\left(n^2\log^2(n)/\lambda(\boldsymbol{Q}) \cdot \log(1/\delta)\right)$ *samples from a Local Sampling Scheme* $\mathrm{Samp}(\mathcal{Q}; \mathcal{D})$*, runs in time polynomial in* N*, and outputs a sample distributed as in* $\mathcal{D}$*.*

of *Theorem 13.* Consider a target distribution $\mathcal{D}$, supported on $[n]$ and sample access to a Local Sampling Scheme $\mathrm{Samp}(\mathcal{Q}; \mathcal{D})$ for a pair distribution $\mathcal{Q}$. Let $\boldsymbol{M} = \boldsymbol{Q} \otimes \boldsymbol{D}$ be the transition matrix of the Markov chain associated with the Local Sampling Scheme graph. From Theorem 4 with accuracy $\epsilon > 0$, there exists an algorithm that, with high probability, computes the rescaling of the Markov chain that corresponds to $\boldsymbol{M}$, which we denote by $\widetilde{\boldsymbol{M}}$, the transition matrix of the rescaled (by $1/\widetilde{\mathcal{D}}$) Markov chain. The learning algorithm uses $\widetilde{O}(n^2/(\lambda(\boldsymbol{Q})))$ with probability $1-\delta$ and is executed only once. We now analyze a single execution of Algorithm 1.

Claim 2 (Termination of CFTP)**.** *Algorithm 1 reaches Line 12 with probability* 1.

Proof. Since the chain $\widetilde{\boldsymbol{M}}$ is ergodic, there is a T such that for any pair of states (x, y), the probability $\widetilde{\boldsymbol{M}}^T(x, y) > 0$. Similarly to the proof sketch of the naive exact sampling algorithm (see also

Theorem 2), let $(X_t)_{t\in\mathbb{Z}}$ be a random walk on $[n]$ and define the mapping $F_{(a,b]} : [n] \to [n]$, so that $F_{(-t_0,T]}(x_0)$ is the state of the random walk at time $t = T$, i.e., $X_T \in [n]$, where the random walk begun at time $t = -t_0$ (in the past) with starting state $X_{-t_0} = x_0$. Then, the n random walks have coalesced after T steps if and only if $|\mathrm{Image}(F_{(-T,0]})| = 1$. By the ergodicity of the chain, each one of the events $E_T = \{|\mathrm{Image}(F_{(-T,0]})| = 1\}$ for $T \in \mathbb{Z}_{\geq 0}$ has strictly positive probability to occur. Since the events are independent, with probability 1, there will be an event $E_{T^\star}$ that occurs. Then, for any $T > T^\star$, the desired property holds. □

Claim 3 (Sample Complexity of a Single Iteration)**.** *It holds that each execution of Algorithm 1 uses* $O\left(\frac{n\log(n)\log(1/\delta)}{\lambda(\boldsymbol{Q})}\right)$ *samples from the Local Sampling Scheme with probability* $1-\delta$.

Proof. The CFTP draws a single sample for each iteration and performs a step according to the matrix $\widetilde{\boldsymbol{M}}$ due to the rescaling. First, the unique stationary distribution is almost uniform (Lemma 15) and so $\log(1/\mathcal{D}_{\min}) = \Theta(\log(n))$. This implies, by Lemma 9 and Lemma 15, that the mixing time of $\widetilde{\boldsymbol{M}}$ for some constant accuracy $\epsilon_0 = 1/4$ is $T_{\mathrm{mix}}(\widetilde{\boldsymbol{M}}; 1/4) = O(\log(n)/\lambda(\boldsymbol{Q}))$. Hence, by Proposition 12, we get the desired result. □

Moreover, the update time of each execution of the CFTP algorithm is polynomial in n. Note that, since during the Learning Phase, the algorithm uses $O(n^2/\lambda(\boldsymbol{Q}))$ and is executed once, the sample complexity of Algorithm 2 follows, since Algorithm 1 will be executed $\Theta(n)$ times.

Claim 4 (Rejection Sampling)**.** *Algorithm 1, with input the output of the Learning Phase, outputs a state* $x \in [n]$ *with probability proportional to* $\mathcal{D}(x)$ *or outputs* $\perp$. *Moreover, Algorithm 2 outputs a perfect sample from* $\mathcal{D}$.

Proof. Let us set $A = \sum_{y\in[n]} \mathcal{D}(y)/\widetilde{D}(y)$. At the end of Line 11, the parameterized CFTP algorithm (Algorithm 1) outputs a sample $x \propto \mathcal{D}(x)/\widetilde{D}(x)$. This is because the unique stationary distribution is $\widetilde{\pi}(x) = (\mathcal{D}(x)/\widetilde{\mathcal{D}}(x))/A$ and the utility of the CFTP mechanism guarantees that when the Algorithm initiates a CFTP simulation from time $T = -T^\star$, the generated sample $x = F_{(-T^\star,0]} = F_{(-\infty,0]}$ will be distributed according to $\widetilde{\pi}$. This follows from the observation that the distribution of the state $F_{(-\infty,0]}$ is equal to the distribution of the state $F_{[0,+\infty)}$ obtained by running the simulation up to the limit $T \to +\infty$ forward in time. Hence, we have that x has law the unique stationary distribution $\widetilde{\pi}$. We now have to remove the bias induced by the learning step. For the sample x, we perform a rejection sampling process, with acceptance probability $\widetilde{\mathcal{D}}(x)$. The Algorithm 1 has $n+1$ potential outputs: It either prints $x \in [n]$ or $\perp$ indicating failure. The sample x is the output of the algorithm with probability $\frac{\mathcal{D}(x)/\widetilde{\mathcal{D}}(x)}{A} \cdot \widetilde{\mathcal{D}}(x) = \mathcal{D}(x)/A$. This holds for any $x \in [n]$. The remaining probability mass is assigned to $\perp$. Hence, a single execution of Algorithm 1 outputs a point $x \in [n]$ with probability $\mathcal{D}(x)/A$ and outputs "reject" with probability $1 - 1/A$. As we will see, the accepted sample is distributed according to $\mathcal{D}$. This is because, conditional on acceptance, its mass is exactly that assigned by $\mathcal{D}$. Specifically, we claim that the output of Line 6 of Algorithm 2 has law $\mathcal{D}$. Observe that the whole stochastic process of Algorithm 2 outputs $x \in [n]$ with probability

$$\begin{aligned}\sum_{i=0}^{\infty} \Pr[\text{Reject}]^i \Pr[x \text{ is Accepted}] &= \sum_{i=0}^{\infty} \Pr[\text{Reject}]^i \cdot \frac{(\mathcal{D}(x)/\widetilde{\mathcal{D}}(x)) \cdot \widetilde{\mathcal{D}}(x)}{\sum_{y\in[n]} \mathcal{D}(y)/\widetilde{\mathcal{D}}(y)} \\ &= \frac{\mathcal{D}(x)}{\sum_{y\in[n]} \mathcal{D}(y)/\widetilde{\mathcal{D}}(y)} \sum_{i=0}^{\infty} \Pr[\text{Reject}]^i \,.\end{aligned}$$

We have that $\Pr[\text{Reject}] = \frac{1}{A}\sum_{y\in[n]} \frac{\mathcal{D}(y)}{\widetilde{\mathcal{D}}(y)}(1 - \widetilde{\mathcal{D}}(y)) = 1 - \frac{1}{A} \in (0,1)$. Hence, we have that the whole stochastic process of Algorithm 2 outputs $x \in [n]$ with probability

$$\sum_{i=0}^{\infty} \Pr[\text{Reject}]^i \Pr[x \text{ is Accepted}] = \frac{\mathcal{D}(x)}{\sum_{y\in[n]} \mathcal{D}(y)/\widetilde{\mathcal{D}}(y)} \cdot A = \mathcal{D}(x)\,,$$

since $\sum_{i\geq 0} \lambda^i = \frac{1}{1-\lambda}$ for $\lambda \in (-1,1)$. As a result, we get that the probability that x is the output of the above stochastic process is $\mathcal{D}(x)$. □

These claims complete the proof. Specifically, the total sample complexity is derived as follows: Since we have to execute the CFTP iterations $\Theta(n)$ times, we should call each CFTP process with $\delta' = C\delta/n$ for some constant C. Proposition 12 guarantees that the CFTP process will terminate using $O(n\log(n)/\lambda(\boldsymbol{Q}) \cdot \log(n/\delta))$ samples with probability $1 - \Theta(\delta/n)$. Hence, applying the the union bound for the $\Theta(n)$ CFTP calls and the single call of the learning algorithm, gives that, with probability $1-\delta$, a number of $O(n^2\log^2(n)/\lambda(\boldsymbol{Q}) \cdot \log(1/\delta))$ samples suffices in order to get a perfect sample from $\mathcal{D}$. □

D The Proof of Lemma 5 (Properties of Rescaled Random Walk)

In the next section, we discuss some useful steps as a warm-up for the proof of Lemma 5. The proof can be found at the Appendix D.2.

D.1 Sketch of the Idea

By downscaling the transition probabilities (as we will see below), we can decouple the Markov chain from $\mathcal{D}$. Then, the transition matrix of the modified Markov chain only depends on the pair distribution $\mathcal{Q}$, and the convergence of the CFTP algorithm is determined by $\lambda(\boldsymbol{Q})$. Hence, we can sample efficiently even from target distributions $\mathcal{D}$ that may be multimodal or has many low probability points. E.g., in case of a multimodal stationary distribution $\mathcal{D}$, the spectral properties of the walk would remind these of a disconnected graph and the sample complexity of Algorithm 4 would be quite high. We use the estimate $\widetilde{\mathcal{D}}$ (which we obtained from our learning algorithm) of the target distribution $\mathcal{D}$ to transform the Markov chain of the Local Sampling Scheme defined in Equation (1) to a modified Markov chain with an almost uniform stationary distribution. This transformation can be viewed as a downscaling mechanism:

Definition 14 (Downscaling)**.** *Let $p \in (0,1)$ and let $X \sim \mathrm{Be}(p)$ be a Bernoulli random variable. Let $\lambda \in (0,1)$. The random variable $Y \sim \mathrm{Be}(\lambda p)$ is called a λ-downscaler of X.*

Applying Theorem 4, we can consider that, for any $x \in [n]$, there exists a coefficient $\widetilde{\mathcal{D}}(x) \approx \mathcal{D}(x)$. The idea is to use $\widetilde{\mathcal{D}}(x)$ and make the stationary distribution of the modified chain close to uniform. Intuitively, this transformation should speedup the convergence of the CFTP algorithm.

Step 1. *Implementation of the downscaling and the rescaled matrix $\widetilde{\boldsymbol{M}}$.*

We can implement the modified Markov chain via downscaling as follows: Consider an edge $\{x, y\}$ with transition probability pair (p_{xy}, p_{yx}). Without loss of generality, we assume that $\widetilde{\mathcal{D}}(y) > \widetilde{\mathcal{D}}(x)$ (which intuitively means that we should expect that $p_{xy} > p_{yx}$). Then, the downscaler leaves p_{yx} unchanged and reduces the mass of p_{xy} to make the two transitions almost balanced. Our exact sampling algorithm will perform this downscaling phase to the matrix $\boldsymbol{M}$ (see Equation (1)).

Consider the transition matrix $\boldsymbol{M}$ with $\boldsymbol{M}_{xy} = \mathcal{Q}(x,y)\frac{\mathcal{D}(y)}{\mathcal{D}(x)+\mathcal{D}(y)}$ and $\boldsymbol{M}_{xx} = 1 - \sum_{y \neq x} \boldsymbol{M}_{xy}$. Also, let $\widetilde{\mathcal{D}}$ be an estimate for the distribution $\mathcal{D}$ with some sufficiently small accuracy ϵ, to be chosen (see also Theorem 4). For the pair (x, y), we modify the transition probability $p_{xy} := \boldsymbol{M}_{xy}$, only if $p_{xy} > p_{yx}$, to be equal to the following

$$\widetilde{p}_{xy} = \mathcal{Q}(x,y)\frac{\mathcal{D}(y)}{\mathcal{D}(x)+\mathcal{D}(y)}\frac{\widetilde{\mathcal{D}}(x)}{\widetilde{\mathcal{D}}(y)} \approx \mathcal{Q}(x,y)\frac{\mathcal{D}(x)}{\mathcal{D}(x)+\mathcal{D}(y)} = p_{yx}\,,$$

where we use that $\widetilde{\mathcal{D}}(x) \approx \mathcal{D}(x)$ and $\widetilde{\mathcal{D}}(y) \approx \mathcal{D}(y)$. The transition probability from x to y corresponds to a Bernoulli variable $\mathrm{Be}(p_{xy})$, which is downscaled by $\widetilde{\mathcal{D}}(x)/\widetilde{\mathcal{D}}(y) < 1$. The modified transition probability $\widetilde{p}_{xy}$ can be implemented by drawing a $\Lambda \sim \mathrm{Be}(\widetilde{\mathcal{D}}(x)/\widetilde{\mathcal{D}}(y))$ and then drawing a $P \sim \mathrm{Be}(p_{xy})$ (from $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$) and, finally, realizing the transition from x to y only if $\Lambda P = 1$. This implementation is valid since the two sources of randomness are independent.

The modified transition matrix $\widetilde{\boldsymbol{M}}$ can be written as:

$$\widetilde{\boldsymbol{M}} = \widetilde{\boldsymbol{D}} \otimes \boldsymbol{Q} + \boldsymbol{Q} \circ [\epsilon_{xy}]\,, \tag{4}$$

where $\widetilde{\boldsymbol{D}}$ is a symmetric matrix with $\widetilde{\boldsymbol{D}}_{xy} = \min\{\mathcal{D}(x), \mathcal{D}(y)\}/(\mathcal{D}(x)+\mathcal{D}(y)) = \widetilde{\boldsymbol{D}}_{yx}$ and $\boldsymbol{Q}$ is the Laplacian matrix of the graph associated with the $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ and $\boldsymbol{Q} \circ [\epsilon_{xy}]$ denotes the modified Hadamard product[4] between the Laplacian $\boldsymbol{Q}$ and the matrix with the estimation error ϵ_{xy} is only non-zero in the modified transitions $x \to y$, i.e., if $p_{xy} > p_{yx}$ then $\widetilde{\boldsymbol{M}}_{xy} = \mathcal{Q}(x,y) \cdot (\frac{\mathcal{D}(x)}{\mathcal{D}(x)+\mathcal{D}(y)} + \epsilon_{xy})$. For this pair, we set $\epsilon_{yx} = 0$. We denote this error matrix with $[\epsilon_{xy}]$.

Specifically, for a pair (x, y), before the downscaling, the pairwise (x, y)-comparison corresponds to the random coin $\left(\frac{\mathcal{D}(y)}{\mathcal{D}(x)+\mathcal{D}(y)}, \frac{\mathcal{D}(x)}{\mathcal{D}(x)+\mathcal{D}(y)}\right)$ and let $\mathcal{D}(y) > \mathcal{D}(x)$. After the downscaling, this coin becomes (locally) almost fair, i.e., $(\frac{\mathcal{D}(x)}{\mathcal{D}(x)+\mathcal{D}(y)} + \epsilon_{xy}, \frac{\mathcal{D}(x)}{\mathcal{D}(x)+\mathcal{D}(y)})$.

Step 2. *Obtaining a simpler matrix $\widetilde{\boldsymbol{M}}$ with absolute spectral gap of the same order.*

Since the coins are locally fair, we can work with the following matrix that has also almost uniform stationary distribution and conductance of the same order (since making each transition (x, y) more lazy (i.e., increasing the probability of $x \to x$ and $y \to y$ by some *constant*) and still keeping the transitions $x \to y$ and $y \to x$ (almost) equal cannot significantly affect the conductance, i.e., the desired symmetry is preserved). Let $\widetilde{\boldsymbol{M}} = \boldsymbol{I} - c \cdot \boldsymbol{Q} + \boldsymbol{Q} \circ [\epsilon_{xy}]$ for some *constant* $0 \le c \le 1/2$. This matrix with $c = \min_{(x,y)\in\mathcal{E}} \mathcal{D}(x)/(\mathcal{D}(x) + \mathcal{D}(y))$ can be obtained by further performing downscaling at each transition (and c is constant due to Assumption 2); then each transition will be equal to the minimum global transition probability (potentially with some $O(1/\sqrt{n})$ noise term). For simplicity, we let $c = 1/2$. The next proof We get the following matrix

$$\widetilde{\boldsymbol{M}}_{xy} = \mathcal{Q}(x,y)\Big(\frac{1}{2} + \epsilon_{xy}\Big),\ \widetilde{\boldsymbol{M}}_{yx} = \mathcal{Q}(x,y)\Big(\frac{1}{2} + \epsilon_{yx}\Big)\,, \tag{5}$$

and

$$\widetilde{\boldsymbol{M}}_{xx} = 1 - \sum_{y\neq x} \widetilde{\boldsymbol{M}}_{xy} = 1 - \frac{1}{2}\sum_{y\neq x} \mathcal{Q}(x,y) - \sum_{y\neq x} \epsilon_{xy}\mathcal{Q}(x,y)\,.$$

Also, we observe that the generator of the Markov chain $\boldsymbol{I} - \widetilde{\boldsymbol{M}}$ is close up to scaling to the Laplacian $\boldsymbol{Q}$. This explains why the convergence time of the Markov chain with transition matrix $\widetilde{\boldsymbol{M}}$ spectrally depends only on the distribution $\mathcal{Q}$. Specifically, the Laplacian $\boldsymbol{Q}$ corresponds to the dominant component for the convergence of the algorithm and the Hadamard product is the low-order noise induced by the chain transformation. As we will see, the larger the smallest non-zero eigenvalue of the Laplacian matrix and, hence, the larger the spectral gap of the transition matrix of the transformed chain, the faster the Markov chain converges to its stationary distribution.

D.2 The Proof of Lemma 5

The following lemma summarizes the key properties of the downscaled random walk.

Lemma 15 (Properties of Rescaled Random Walk)**.** *Let $\mathcal{D}$ be a distribution on $[n]$ and consider an ϵ-relative approximation $\widetilde{\mathcal{D}}$ of $\mathcal{D}$, as in Theorem 4 with $\epsilon = O(1/\sqrt{n})$. Consider the transition matrix $\boldsymbol{M}$ of the Local Sampling Scheme $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ (see Equation (1)), and let $\widetilde{\boldsymbol{M}}$ be the $\widetilde{\mathcal{D}}$-scaling of $\boldsymbol{M}$ (see Equation (4)). Then, the following hold.*

- (i) *The transition matrix $\widetilde{\boldsymbol{M}}$ has stationary distribution $\widetilde{\pi}_0(x) = \Theta(1/n)$ for all $x \in [n]$.*
- (ii) *The absolute spectral gap $\Gamma(\widetilde{\boldsymbol{M}})$ of $\widetilde{\boldsymbol{M}}$ and the minimum non-zero eigenvalue $\lambda(\boldsymbol{Q})$ of the Laplacian matrix $\boldsymbol{Q}$ satisfy $\Gamma(\widetilde{\boldsymbol{M}}) = \Omega(\lambda(\boldsymbol{Q}))$.*
- (iii) *For any $\epsilon_0 \in (0, 1)$, the mixing time of the transition matrix $\widetilde{\boldsymbol{M}}$ is $T_{\mathrm{mix}}(\widetilde{\boldsymbol{M}};\epsilon_0) = O\left(\frac{\log(n/\epsilon_0)}{\lambda(\boldsymbol{Q})}\right)$.*

In the above, we can choose $\epsilon_0 = 1/4$. The proof of the Lemma 15 follows.

[4] We have that $\Big(\boldsymbol{Q} \circ \boldsymbol{A}\Big)_{xy} = \mathcal{Q}_{xy}A_{xy}$, $\Big(\boldsymbol{Q} \circ \boldsymbol{A}\Big)_{xx} = -\sum_{y\neq x} \mathcal{Q}_{xy}A_{xy}$.

of *Lemma 15*. We remark that it suffices to work with the matrix of Equation (5), since the results will only change by at most some constant. Observe that the spectral gap of the matrix of Equation (4) is at least as high as one of the matrix $\boldsymbol{I} - c \cdot \boldsymbol{Q} + \boldsymbol{Q} \circ [\epsilon_{xy}]$ with constant $c = \min_{(x,y)\in\mathcal{E}} \mathcal{D}(x)/(\mathcal{D}(x) + \mathcal{D}(y)) = \Theta(\phi)$ and this matrix has spectral gap of the same order as the matrix of Equation (5). We break the proof into three claims.

Claim 5. *The transition matrix $\widetilde{\boldsymbol{M}}$ has stationary distribution $\widetilde{\pi}_0$, that satisfies $\widetilde{\pi}_0(x) = \Theta(1/n)$ for any $x \in [n]$.*

Proof. The chain $\widetilde{\boldsymbol{M}}$ remains irreducible and the detailed balance equations of the matrix $\widetilde{\boldsymbol{M}}$ satisfy:

$$\frac{\widetilde{p}_{xy}}{\widetilde{p}_{yx}} = \frac{\mathcal{Q}(x,y)\mathcal{D}(y)/\widetilde{\mathcal{D}}(y)}{\mathcal{Q}(y,x)\mathcal{D}(x)/\widetilde{\mathcal{D}}(x)} .$$

Since $\widetilde{\mathcal{D}}$ is an ϵ-relative approximation of $\mathcal{D}$ with $\epsilon = O(1/\sqrt{n})$, it holds that for any $x \in [n]$:

$$\mathcal{D}(x)/\widetilde{\mathcal{D}}(x) \in [1-\epsilon, 1+\epsilon] ,$$

and hence the unique stationary distribution $\widetilde{\pi}_0$ satisfies: $\widetilde{\pi}_0(x)/\widetilde{\pi}_0(y) \in [1-\epsilon, 1+\epsilon]$ for any $x, y \in [n]$. So, this holds for $x = \operatorname{argmax} \widetilde{\pi}_0$ and $y = \operatorname{argmin} \widetilde{\pi}_0$ and, since it should hold that $\sum_x \widetilde{\pi}_0(x) = 1$, it must be the case that $\widetilde{\pi}(x) = \Theta(1/n)$. □

Claim 6. *For the absolute spectral gap $\Gamma(\widetilde{\boldsymbol{M}})$ of $\widetilde{\boldsymbol{M}}$ and the minimum non-zero eigenvalue $\lambda(\boldsymbol{Q})$ of the Laplacian matrix $\boldsymbol{Q}$, it holds that: $\Gamma(\widetilde{\boldsymbol{M}}) = \Omega(\lambda(\boldsymbol{Q}))$.*

Proof. For $x \neq y$, there exist error estimates ϵ_{xy} and ϵ_{yx}; one of them is zero and the other's absolute value is of order $O(1/\sqrt{n})$. Our goal is to control the absolute spectral gap $\Gamma(\widetilde{\boldsymbol{M}})$. Recall that the transition matrix can be written as follows

$$\widetilde{\boldsymbol{M}} = \boldsymbol{I} - \frac{1}{2}\boldsymbol{Q} + \boldsymbol{Q} \circ [\epsilon_{xy}] ,$$

where $\epsilon_{xy} \in [-\epsilon, \epsilon]$ for some $\epsilon = O(1/\sqrt{n})$. Note that $[\epsilon_{xy}]$ denotes the $n \times n$ error matrix and the operator $\circ$ denotes the standard Hadamard product. Note that the matrix $\boldsymbol{I} - \frac{1}{2}\boldsymbol{Q}$ is symmetric, while $\boldsymbol{Q} \circ [\epsilon_{xy}]$ needs not to be.

Let $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n$ be the spectrum of the transition matrix $\widetilde{\boldsymbol{M}}$. Also, the matrix is right stochastic with $\widetilde{\boldsymbol{M}}\mathbf{1} = \mathbf{1}$ and $\lambda_1 = 1$. Let $\Gamma(\widetilde{\boldsymbol{M}})$ be the absolute spectral gap, that is strictly positive by aperiodicity and irreducibility. The Weyl's Inequality for general matrices ([HJ12, Tao12]) describes a multiplicative majorization between the ordered absolute eigenvalues and singular values of $\boldsymbol{A}$ and gives a useful perturbation bound. Recall that $\sigma_i^2(\boldsymbol{A}) = \lambda_i(\boldsymbol{A}\boldsymbol{A}^*) = \lambda_i(\boldsymbol{A}^*\boldsymbol{A})$ for an arbitrary matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$.

Fact 2 (Weyl's Inequality (see [HJ12]))**.** *Consider the matrices $\boldsymbol{A}, \boldsymbol{E} \in \mathbb{C}^{n\times n}$. Define the singular value of $\boldsymbol{A}$ in decreasing order (counting multiplicity) $\sigma_1(\boldsymbol{A}) \geq \ldots \geq \sigma_n(\boldsymbol{A}) \geq 0$. Then, for $k = 1, \ldots, n$, the following hold*

- $\prod_{i\in[k]} |\lambda_i(\boldsymbol{A})| \leq \prod_{i\in[k]} \sigma_i(\boldsymbol{A})$ *with* $|\lambda_1(\boldsymbol{A})| \geq |\lambda_2(\boldsymbol{A})| \geq \ldots |\lambda_n(\boldsymbol{A})|$.
- $|\sigma_k(\boldsymbol{A}+\boldsymbol{E}) - \sigma_k(\boldsymbol{A})| \leq \|\boldsymbol{E}\|_2 .$

For the matrix $\widetilde{\boldsymbol{M}}$, we have that $|\lambda_1| = 1$ and the second largest absolute eigenvalue is $\max\{|\lambda_2|, |\lambda_n|\}$. We can apply the first property for the singular values from the Fact 2 and get that

$$\Gamma(\widetilde{\boldsymbol{M}}) = \lambda_1 - \max\{|\lambda_2|, |\lambda_n|\} \geq 1 - \sigma_1(\widetilde{\boldsymbol{M}})\sigma_2(\widetilde{\boldsymbol{M}}) ,$$

where σ_1, σ_2 correspond to the two largest singular values. Observe that the largest singular value is $\sigma_1(\widetilde{\boldsymbol{M}}) = \max_{\|\boldsymbol{v}\|_2=1} \|\widetilde{\boldsymbol{M}}\boldsymbol{v}\| = 1$ and, using the second property of Fact 2, we can control the perturbation of the second largest singular value

$$\sigma_2\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{Q}\right) - \|\boldsymbol{Q} \circ [\epsilon_{xy}]\|_2 \leq \sigma_2(\widetilde{\boldsymbol{M}}) \leq \sigma_2\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{Q}\right) + \|\boldsymbol{Q} \circ [\epsilon_{xy}]\|_2 .$$

In order to lower bound the absolute spectral gap, it suffices to upper bound the second largest singular value. But, for the second largest singular eigenvalue of the real symmetric matrix $\boldsymbol{I} - \boldsymbol{Q}/2$, it holds that

$$\sigma_2\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{Q}\right) = \max\left\{\left|\lambda_2\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{Q}\right)\right|, \left|\lambda_n\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{Q}\right)\right|\right\}.$$

Since the matrix $\boldsymbol{I} - \boldsymbol{Q}/2$ is symmetric, we have that

$$\sigma_2\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{Q}\right) = \max\left\{\left|\max_{\boldsymbol{v}\perp\boldsymbol{1}, \|\boldsymbol{v}\|_2=1} \boldsymbol{v}^\top\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{Q}\right)\boldsymbol{v}\right|, \left|\min_{\|\boldsymbol{v}\|_2=1} \boldsymbol{v}^\top\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{Q}\right)\boldsymbol{v}\right|\right\},$$

and, hence, if we let $0 = \mu_1 \leq \mu_2 \leq \ldots \leq \mu_n$ be the spectrum of the Laplacian matrix $\boldsymbol{Q}$, we have that

$$\sigma_2\left(\boldsymbol{I} - \frac{1}{2}\boldsymbol{Q}\right) = \max\left\{\left|1 - \frac{1}{2}\mu_2(\boldsymbol{Q})\right|, \left|1 - \frac{1}{2}\mu_n(\boldsymbol{Q})\right|\right\}.$$

For the Laplacian matrix $\boldsymbol{Q}$, observe that $|\boldsymbol{Q}_{xx}| = |\sum_{y\sim x} \mathcal{Q}(x,y)| \leq 1$. Hence, from Gershgorin's Theorem (see Lemma 11), we get that all the eigenvalues of the Laplacian lie on the real axis (since the matrix is symmetric) and all the disks $B(x,r)$ are centered at points $|x| \leq 1$ and the radii r are upper bounded by 1. Hence, it holds that $0 \leq \mu_2(\boldsymbol{Q}) \leq \mu_n(\boldsymbol{Q}) \leq 2$. So, we can take that $\sigma_2(\boldsymbol{I} - \frac{1}{2}\boldsymbol{Q}) = 1 - \frac{1}{2}\mu_2(\boldsymbol{Q}) \stackrel{\text{def}}{=} 1 - \frac{1}{2}\lambda(\boldsymbol{Q}) \geq 0$, where $\lambda(\boldsymbol{Q})$ denotes the smallest non-zero eigenvalue of the Laplacian matrix $\boldsymbol{Q}$.

Now, we have that $\|\boldsymbol{Q} \circ [\epsilon_{xy}]\|_2 = \max_{\|\boldsymbol{v}\|_2=1} \|(\boldsymbol{Q} \circ [\epsilon_{xy}])\boldsymbol{v}\|_2 = O(1/\sqrt{n}) \cdot \mu_n(\boldsymbol{Q}) = O(1/\sqrt{n})$ and so we get $\sigma_2(\widetilde{\boldsymbol{M}}) \leq (1 - \Theta(\lambda(\boldsymbol{Q}))) + O(1/\sqrt{n})$. This implies that

$$\sigma_2(\widetilde{\boldsymbol{M}}) \lesssim \max\{1 - \Theta(\lambda(\boldsymbol{Q})), O(1/\sqrt{n})\},$$

and so

$$\Gamma(\widetilde{\boldsymbol{M}}) \gtrsim 1 - \max\{1 - \Theta(\lambda(\boldsymbol{Q})), O(1/\sqrt{n})\}$$

This implies that $\Gamma(\widetilde{\boldsymbol{M}}) = \Omega(\lambda(\boldsymbol{Q}))$ for sufficiently large n. The above proof will be similar for any matrix $\widetilde{\boldsymbol{M}} = I - c \cdot \boldsymbol{Q} + \boldsymbol{Q} \cdot [\epsilon_{xy}$ for $0 \leq c \leq 1/2$ which is obtained by the discussion of **Step 2**. Such a matrix has an absolute spectral gap that is, on the one side, lower bounded by $\lambda(\boldsymbol{Q})$ (as we showed above) and, on the other side, upper bounded by the absolute spectral gap of the original matrix of Equation (4). This completes the proof. □

We continue with the mixing time result of the transformed Markov chain. Since we have shown that the absolute spectral gap is $\Omega(\lambda(\boldsymbol{Q}))$, we can directly get the desired result, whose proof relies on the analysis of Lemma 9 and can be found, e.g., in [LP17].

Claim 7. *The ϵ_0-mixing time of the transition matrix $\widetilde{\boldsymbol{M}}$ is equal to*

$$T_{\text{mix}}(\widetilde{\boldsymbol{M}}; \epsilon_0) = O(\log(n/\epsilon_0)/\lambda(\boldsymbol{Q})).$$

In the above claim, we choose $\epsilon_0 = 1/4$.

Combining these claims, the proof is completed. □

E The Proof of Theorem 4 (Main Result of Learning Phase)

The first step of Algorithm 2 is to learn the target distribution $\mathcal{D}$ in ϵ-relative error for some $\epsilon > 0$ and pass it as input to Algorithm 1. In this section, we will provide the learning results for abstract ϵ; however, our algorithm applies these results with $\epsilon = 1/\sqrt{n}$. For two distributions $\mathcal{D}, \widetilde{\mathcal{D}}$ with ground set $[n]$, we introduce the sequence/list (of length n) $1 - \mathcal{D}/\widetilde{\mathcal{D}} := (1 - \mathcal{D}(x)/\widetilde{\mathcal{D}}(x))_{x\in[n]}$. Observe that the pair of sequences $(1 - \mathcal{D}/\widetilde{\mathcal{D}}, 1 - \widetilde{\mathcal{D}}/\mathcal{D})$ captures the relative error between the two distributions. The sample complexity of the task of learning $\mathcal{D}$ in ϵ-relative error is summarized by the following (restatement of Theorem 4):

Theorem. *For any $\epsilon, \delta > 0$, there exists an algorithm (Algorithm 5) that draws $N = O\left(\frac{n}{\lambda(\boldsymbol{Q})\epsilon^2}\log(\frac{1}{\delta})\right)$ samples from a Local Sampling Scheme $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$ satisfying Assumptions 1 and 2, runs in time polynomial in N, and, with probability at least $1-\delta$, computes an estimate $\widetilde{\mathcal{D}}$ of the target distribution $\mathcal{D}$, that satisfies the following relative error guarantee $\max\left\{\|1-\mathcal{D}/\widetilde{\mathcal{D}}\|_\infty, \|1-\widetilde{\mathcal{D}}/\mathcal{D}\|_\infty\right\} \leq \epsilon$.*

Algorithm 5 Learn using Shifting from (Pairwise) Local Sampling Schemes

procedure LEARN-SHIFT(ϵ, δ) ▷ *Sample access to oracle* $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$.
  Set $N = \Theta\left(\frac{n}{\lambda(\boldsymbol{Q})\epsilon^2}\log(\frac{1}{\delta})\right)$.
  Draw N samples of the form $((x_i, y_i), q_i) \in \mathcal{E} \times \{0,1\}$.
  Obtain $\widehat{\boldsymbol{z}}$ using Algorithm 6 (the estimation vector satisfies $\langle \mathbf{1}, \widehat{\boldsymbol{z}}\rangle = 0$ and must be shifted).
  Compute $C = \log\left(\sum_{x\in[n]}\exp(\widehat{z}_x)\right)$
  Set $z'_x = \widehat{z}_x - C$ for any $x \in [n]$ ▷ *See Appendix F.2.*
  Output $\boldsymbol{z}'$ satisfying $\|\boldsymbol{z} - \boldsymbol{z}'\|_\infty \leq \epsilon$.
end procedure

Let us sketch the proof of Theorem 4. For an arbitrary weight vector $\boldsymbol{w} \in \mathbb{R}^n_{>0}$, we use the re-parameterization $z_i = \log(w_i)$. When in addition $\boldsymbol{w} \in \Delta^n$, i.e., $\boldsymbol{w}$ is a probability distribution over $[n]$ and is usually denoted by $\mathcal{D}$, we call $\boldsymbol{z}$ the natural parameter vector of $\mathcal{D}$. Recall that $\boldsymbol{Q}$ is a Laplacian matrix where $\boldsymbol{Q}_{xy} = -\mathcal{Q}(x,y)$, i.e., it is the Laplacian matrix of the graph $G_{\mathcal{Q}}$ weighted by the mass assigned by $\mathcal{Q}$. The proof goes as follows: we draw i.i.d. samples from the LSS. We first apply the following result which is a modification of the results of [SBB+16].

Theorem 16 (Variant of [SBB+16])**.** *Let $G_{\mathcal{Q}}$ be the graph of the support $\mathcal{E}$ of $\mathcal{Q}$ satisfying Assumption 1 with $|V| = n$ vertices and associated weighted Laplacian matrix $\boldsymbol{Q}$ (see Equation (2)). Let $\boldsymbol{L}$ be the associated empirical Laplacian matrix with $\mathbb{E}[\boldsymbol{L}] = \boldsymbol{Q}$. Consider a vector $\boldsymbol{z} \in \mathbb{R}^n$ with $\langle \mathbf{1}, \boldsymbol{z}\rangle = 0$ satisfying the constraint $\max_{x,y\in\mathcal{E}}|z_x - z_y| \leq \log(\phi)$ for some constant $\phi > 1$. Let $\mathrm{Ex}(\mathcal{Q};\boldsymbol{z})$ be the oracle that generates the example $\{(x,y), q\}$ as follows: the edge (x,y) is chosen with probability $\mathcal{Q}(x,y)$ and the bit q is set to 1 with probability $\exp(z_x)/(\exp(z_x)+\exp(z_y))$; otherwise it is set to 0. For any $\epsilon, \delta > 0$, there is a maximum likelihood estimator of $\boldsymbol{z}$ (Algorithm 6) which draws $N = O\left(\frac{n}{\lambda(\boldsymbol{L})\epsilon^2}\log(\frac{1}{\delta})\right)$ samples from $\mathrm{Ex}(\mathcal{Q};\boldsymbol{z})$ and computes, in time polynomial in the number of samples N, an estimate $\widehat{\boldsymbol{z}}$ such that $\|\boldsymbol{z} - \widehat{\boldsymbol{z}}\|_2 \leq \epsilon$, with probability at least $1-\delta$.*

The proof can be found at Appendix F.1. Some comments are in order:

1. In the work of [SBB+16], the target is the (re-parameterized) weights vector $\boldsymbol{z}^\star \in \mathbb{R}^n$ satisfying the conditions $\langle \mathbf{1}, \boldsymbol{z}^\star\rangle = 0$ and $\|\boldsymbol{z}^\star\|_\infty \leq B$ for some constant B. The provided algorithm minimizes the empirical log-likelihood and, using $O(n/(\lambda(\boldsymbol{Q})\cdot\epsilon^2)))$ samples, computes an estimate $\widehat{\boldsymbol{z}}$ so that $\|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_2 \leq \epsilon$.
2. Our provided Algorithm 6 has the same guarantees as the algorithm of [SBB+16] but the target weight vector satisfies the conditions $\langle \mathbf{1}, \boldsymbol{z}^\star\rangle = 0$ and $|z^\star_x - z^\star_y| \leq B$ for some constant B. The algorithm draws sufficiently many samples and then minimizes the empirical negative log-likelihood objective over an appropriately selected constrained set (see Appendix F.1). Observe that for a Local Sampling Scheme with distribution $\mathcal{D}$, the natural parameter vector $\boldsymbol{z}^\star$ does not satisfy $\langle \mathbf{1}, \boldsymbol{z}^\star\rangle = 0$ and so Theorem 16 and Algorithm 6 cannot be directly applied.
3. We will discuss (Appendix F.2) how to apply Algorithm 6 to distributions, i.e., target weight vectors with $\langle \mathbf{1}, \boldsymbol{z}^\star\rangle \leq 0$. This results in learning the target vector $\boldsymbol{z}^\star$ in L_∞ norm (which is sufficient for Algorithm 2).

The algorithm of Theorem 16 follows:

To complete the proof of Theorem 4, we combine the upcoming Lemma 17 and Lemma 18 with the main result of [SBB+16].

Lemma 17 states that, after drawing sufficiently many samples from $\mathrm{Samp}(\mathcal{Q};\mathcal{D})$, we can construct an empirical Laplacian matrix $\boldsymbol{L}$, whose Fiedler eigenvalue is of the same order as the one of the

Algorithm 6 Learning from (Pairwise) Local Sampling Schemes

1: **procedure** LEARN(ϵ, δ) ▷ *Sample access to oracle* $\mathrm{Ex}(\mathcal{Q}; \boldsymbol{z})$.
2: Set $N = \Theta\left(\frac{n}{\lambda(\boldsymbol{Q})\epsilon^2}\log(\frac{1}{\delta})\right)$.
3: Draw N samples of the form $((x_i, y_i), q_i) \in \mathcal{E} \times \{0, 1\}$.
4: Compute the empirical negative log-likelihood objective

$$L_N(\boldsymbol{z}; \{((x_i, y_i), q_i)\}_{i\in[N]}) = -\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{q_i = 1\}z_{x_i} + \mathbf{1}\{q_i = 0\}z_{y_i} - \log(\exp(z_{x_i}) + \exp(z_{y_i}))\,.$$

5: Minimize L_N using gradient descent in the subspace Ω_ϕ. ▷ *See Appendix F.1.*
6: Output the guess vector $\widehat{\boldsymbol{z}}$.
7: **end procedure**

unknown matrix $\boldsymbol{Q}$ of the Local Sampling Scheme. Moreover, notice that Lemma 17 implies that, with high probability, the graph induced by the matrix $\boldsymbol{L}$ is connected, i.e., $\lambda(\boldsymbol{L}) > 0$.

Lemma 17 (Concentration of Empirical Fiedler Eigenvalue)**.** *Let $\epsilon > 0$ and $V = \{\boldsymbol{v} \in \{\pm 1, 0\}^n : \boldsymbol{v} = \boldsymbol{e}_i - \boldsymbol{e}_j$ for any $i < j\}$ and let $\mathcal{Q}$ be a distribution over V. Let $\boldsymbol{Q}$ be the Laplacian matrix of $\mathcal{Q}$ (see Equation* (2)*). There exists an algorithm that uses $O(\log(n/\delta)/(\lambda(\boldsymbol{Q})\epsilon^2))$ samples from $\mathcal{Q}$ and computes a matrix $\boldsymbol{L}$ that satisfies $|\lambda(\boldsymbol{L}) - \lambda(\boldsymbol{Q})| \le \epsilon\lambda(\boldsymbol{Q})$, with probability at least $1 - \delta$, where $\lambda(\cdot)$ denotes the second smallest eigenvalue of a Laplacian matrix.*

Proof. Let $\{\boldsymbol{X}_i = \boldsymbol{v}_i\boldsymbol{v}_i^T\}$ be a finite sequence of m independent symmetric square matrices (of common dimension n) with $\boldsymbol{v}_i \sim \mathcal{Q}$, let $\boldsymbol{L} = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{X}_i$, and let $M = \lambda_{\max}(\boldsymbol{X}_i) = \Theta(1)$. To deal with the second smallest eigenvalue of the Laplacian matrix, we can project the Laplacian matrix $\boldsymbol{L}$ to the orthogonal complement of the vector $\mathbf{1}$. Under this transformation, we can obtain a matrix $\boldsymbol{L}'$ that has two crucial properties. First, it still is a sum of independent positive semidefinite terms and its minimum eigenvalue coincides with the Fiedler eigenvalue of $\boldsymbol{L}$. To achieve this, as in [Tro15], we introduce the transformation $\boldsymbol{R} \in \mathbb{R}^{(n-1)\times n}$, that satisfies $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}_{n-1}$ and $\boldsymbol{R}\mathbf{1} = \mathbf{0}$. Hence, it holds that $\boldsymbol{L}' = \boldsymbol{R}\boldsymbol{L}\boldsymbol{R}^T$ and $\mathbb{E}[\boldsymbol{L}'] = \boldsymbol{R}\boldsymbol{Q}\boldsymbol{R}^T$. Then,

$$\Pr_{\boldsymbol{v}_{1..m}\sim\mathcal{Q}^{\otimes m}}\left[\lambda(\boldsymbol{L}) \le (1-\epsilon)\lambda(\boldsymbol{Q})\right] = \Pr_{\boldsymbol{v}_{1..m}\sim\mathcal{Q}^{\otimes m}}\left[\lambda_{\min}(\boldsymbol{L}') \le (1-\epsilon)\lambda_{\min}(\boldsymbol{R}\boldsymbol{Q}\boldsymbol{R}^T)\right].$$

Using Proposition 7, we get that:

$$\Pr_{\boldsymbol{v}_{1..m}\sim\mathcal{Q}^{\otimes m}}\left[\lambda(\boldsymbol{L}) \le (1-\epsilon)\lambda(\boldsymbol{Q})\right] \le n\left(e^{-\epsilon}/(1-\epsilon)^{1-\epsilon}\right)^{\lambda_{\min}(\boldsymbol{R}\boldsymbol{Q}\boldsymbol{R}^T)/M} \le n\exp\left(-\epsilon^2 m\frac{\lambda(\boldsymbol{Q})}{2M}\right).$$

To upper bound this probability by δ, it suffices to draw $O(\log(n/\delta)/(\epsilon^2\lambda(\boldsymbol{Q})))$ samples. So, we get that, with probability at least $1 - \delta$, the Fiedler eigenvalue of the empirical matrix $\boldsymbol{L}$ satisfies $\lambda(\boldsymbol{L}) \in (1 \pm \epsilon)\lambda(\boldsymbol{Q})$. Moreover, the second smallest eigenvalue of the empirical Laplacian matrix $\boldsymbol{L}$ is strictly positive and thus, the induced graph is connected (recall that the number of connected components in the graph is the dimension of the nullspace of the Laplacian matrix). □

The next lemma guarantees that learning the natural parameters of any distribution in L_∞ norm is sufficient for learning the distribution with small relative error over its support.

Lemma 18 (Stability of Relative Error)**.** *Let $\mathcal{D}_1, \mathcal{D}_2$ be discrete distributions supported on the ground set $[n]$ and let $\boldsymbol{z}_1, \boldsymbol{z}_2$ be the corresponding natural parameter vectors. For any sufficiently small accuracy parameter $\epsilon > 0$, if it holds that $\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|_\infty \le \epsilon$, then the distributions $\mathcal{D}_1, \mathcal{D}_2$ are close in relative error, i.e.,* $\max\left\{\|1 - \mathcal{D}_1/\mathcal{D}_2\|_\infty, \|1 - \mathcal{D}_2/\mathcal{D}_1\|_\infty\right\} \le \epsilon$.

of *Lemma 18.* Let $\epsilon > 0$ sufficiently small and assume that $\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|_\infty \le \epsilon$, i.e.,

$$\max_{x\in[n]}\left|\log\left(\frac{\mathcal{D}_1(x)}{\mathcal{D}_2(x)}\right)\right| \le \epsilon\,.$$

For $x > 0$, it holds that $1 - \frac{1}{x} \leq \log_e(x)$[5]. Hence, for any $x \in [n]$, we get that

$$1 - \frac{\mathcal{D}_2(x)}{\mathcal{D}_1(x)} \leq \log\left(\frac{\mathcal{D}_1(x)}{\mathcal{D}_2(x)}\right) \leq \epsilon\,,$$

and

$$1 - \frac{\mathcal{D}_1(x)}{\mathcal{D}_2(x)} \leq \log\left(\frac{\mathcal{D}_2(x)}{\mathcal{D}_1(x)}\right) \leq \epsilon\,.$$

This gives that

$$\frac{\mathcal{D}_1(x)}{\mathcal{D}_2(x)} \geq 1 - \epsilon \;\text{ and }\; \frac{\mathcal{D}_2(x)}{\mathcal{D}_1(x)} \geq 1 - \epsilon\,.$$

This implies that both ratios are upper bounded by $1 + \epsilon$. For contradiction, assume that

$$\frac{\mathcal{D}_1(x)}{\mathcal{D}_2(x)} > 1 + \epsilon \iff \frac{\mathcal{D}_2(x)}{\mathcal{D}_1(x)} < \frac{1}{1+\epsilon}\,.$$

But, the Taylor expansion of the function $x \mapsto \frac{1}{1+x}$ is equal to $1 - x + O(x^2)$ for $|x| < 1$. Hence, for sufficiently small ϵ, the desired bound follows. □

To wrap up, the proof of Theorem 4 goes as follows:

of *Theorem 4.* Assume that $\boldsymbol{z}^\star \in \mathbb{R}^n$ is the true natural parameter vector for the target distribution $\mathcal{D}$. By Lemma 18, in order to obtain the desired relative error, it suffices to control $\boldsymbol{z}^\star$ in L_∞. Also, let $\boldsymbol{z} \in \mathbb{R}^n$ be a shifted variant of $\boldsymbol{z}^\star$ so that $\langle \mathbf{1}, \boldsymbol{z} \rangle = 0$, i.e., $\boldsymbol{z}^\star = \boldsymbol{z} + C\mathbf{1}$ for some C. First, the algorithm draws $N = \Theta\left(\frac{n}{\lambda(\boldsymbol{Q})\epsilon^2}\log(\frac{1}{\delta})\right)$ i.i.d. samples from the LSS with Laplacian matrix $\boldsymbol{Q}$. Using the concentration of the Fiedler eigenvalue (Lemma 17), we have that this number of samples is sufficient to apply Theorem 16. This will yield a vector $\widehat{\boldsymbol{z}}$ so that $\|\boldsymbol{z} - \widehat{\boldsymbol{z}}\|_2 \leq \epsilon$ with probability $1 - \delta$. We then apply the transformation described in Appendix F.2 (which corresponds to Line 5-6 of Algorithm 5). This yields the desired L_∞ bound and completes the proof. □

F The Proof of Theorem 16 (Variant of [SBB$^+$16])

In this section, we prove a slight variant of the algorithm of [SBB$^+$16] where the target weight vector satisfies the conditions $\langle \mathbf{1}, \boldsymbol{z}^\star \rangle = 0$ and $|z^\star_x - z^\star_y| \leq B$ for some constant B.

F.1 Description of the Learning Algorithm for Abstract Weight Vectors

In this section, we provide a complete description of the learning algorithm of [SBB$^+$16] (with a slight modification). We remark that the technical steps required in order to establish our result follow the analysis of [SBB$^+$16]. It suffices to present an efficient learning algorithm that estimates the target weights vector $\boldsymbol{z}^\star = (z^\star_1, \ldots, z^\star_n)$ in L_∞ norm. *We underline that for this section, the target vector $\boldsymbol{z}^\star$ is an abstract weight vector and not the natural parameter vector induced by a distribution.* We assume that $\boldsymbol{z}^\star$ lies in the subspace Ω_ϕ, defined in Equation (6). Hence, it suffices that the learning algorithm computes an estimate $\widehat{\boldsymbol{z}}$ such that $\|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_2 \leq \epsilon$, with high probability, for some accuracy parameter $\epsilon > 0$.

of *Theorem 16.* The algorithm minimizes the empirical negative log-likelihood over a subspace $\Omega_\phi \subseteq \mathbb{R}^n$ of the natural parameter vectors. The empirical negative log-likelihood objective with N draws from the Local Sampling Scheme corresponds to the function

$$L_N(\boldsymbol{z}; \{((x_i, y_i), q_i)\}_{i\in[N]}) = -\frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{q_i = 1\} z_{x_i} + \mathbf{1}\{q_i = 0\} z_{y_i} - \log(\exp(z_{x_i}) + \exp(z_{y_i}))\,,$$

where $x_i, y_i \in [n]$ and $q_i \in \{0, 1\}$ for any $i \in [N]$. We optimize this objective using gradient descent in the subspace Ω_ϕ, where

$$\Omega_\phi = \{\boldsymbol{z} \in \mathbb{R}^n : \langle \mathbf{1}, \boldsymbol{z} \rangle = 0, \max_{(x,y)\in\mathcal{E}} |z_x - z_y| \leq \log(\phi)\}\,. \tag{6}$$

[5] This inequality holds when the base of the logarithm is e, the base that we have assumed that we work with.

The first constraint $\langle \mathbf{1}, \boldsymbol{z} \rangle = 0$ is imposed in order to work in the subspace where Laplacian matrices are positive definite (since any Laplacian matrix has its first eigenvalue equal to 0 with corresponding eigenvector $\mathbf{1}$). The second constraint[6] is equivalent to Assumption 2. Since any valid target distribution satisfies $\frac{1}{\phi} \leq \frac{\mathcal{D}(x)}{\mathcal{D}(y)} \leq \phi$ for $(x, y) \in \mathcal{E}$, we have that

$$-\log(\phi) \leq z_x - z_y \leq \log(\phi) \text{ for any } x, y \in \mathcal{E} .$$

Let $\boldsymbol{L}$ be the empirical Laplacian matrix whose Fiedler eigenvalue $\lambda_2(\boldsymbol{L})$ is close to the true $\lambda_2(\boldsymbol{Q})$ (see Lemma 17). We introduce the following notation for the quadratic form $\|\boldsymbol{v}\|_L^2 := \boldsymbol{v}^T \boldsymbol{L} \boldsymbol{v}$ for any vector $\boldsymbol{v} \in \mathbb{R}^n$. The key idea of the algorithm of [SBB+16] is to compute an estimate $\widehat{\boldsymbol{z}}$ in the $\| \cdot \|_L$ semi-norm. Afterwards, by the min-max principle for Hermitian matrices, we have that

$$\|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_L^2 \geq \lambda_2(\boldsymbol{L}) \|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_2^2 . \tag{7}$$

Hence, the estimation of the true natural parameter vector in L_2 norm is directly implied. The analysis for the estimation in the $\boldsymbol{L}$ semi-norm is based on the following fact about M-estimators. A proof of this result can be found in [SBB+16].

Fact 3 (see [SBB+16])**.** *Let $\Omega_1 = \{\boldsymbol{z} \in \mathbb{R}^n : \langle \mathbf{1}, \boldsymbol{z} \rangle = 0\}$ and let $\Omega \subseteq \Omega_1$. Consider the M-estimator*

$$\widehat{\boldsymbol{z}} \in \operatorname*{argmin}_{\boldsymbol{z} \in \Omega} \ell(\boldsymbol{z}) ,$$

and let ℓ be a differentiable objective that is κ-strongly convex[7] at $\boldsymbol{z}^\star \in \Omega$ with respect to the $\boldsymbol{L}$ semi-norm. Then, it holds that

$$\|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_L^2 \leq \frac{1}{\kappa} \|\nabla \ell(\boldsymbol{z}^\star)\|_{L^\dagger} ,$$

where $\boldsymbol{L}^\dagger$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{L}$.

Using the above result, it suffices to verify that the empirical negative log-likelihood objective is strongly convex in the true parameter vector $\boldsymbol{z}^\star \in \Omega_\phi$ and to upper bound the dual norm $\|\nabla L_N(\boldsymbol{z}^\star)\|_{L^\dagger} = \nabla L_N(\boldsymbol{z}^\star)^T \boldsymbol{L}^\dagger \nabla L_N(\boldsymbol{z}^\star)$. Recall that $\boldsymbol{L}$ is the empirical estimate of the true Laplacian matrix $\boldsymbol{Q}$. We continue with two claims, that are sufficient in order to control the $\boldsymbol{L}$ semi-norm of the vector $\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}$.

Claim 8 (Strong Convexity at $\boldsymbol{z}^\star$)**.** *The empirical negative log-likelihood $L_N(\boldsymbol{z})$ is κ-strongly convex at $\boldsymbol{z}^\star \in \Omega_\phi$ with respect to the $\boldsymbol{L}$ semi-norm with $\kappa = \mathrm{poly}(1/\phi)$.*

Proof. It suffices to lower bound the quadratic form $\boldsymbol{w}^T \nabla L_N(\boldsymbol{z}) \boldsymbol{w}$ for all vectors $\boldsymbol{w} \in \mathbb{R}^n$ and $\boldsymbol{z} \in \Omega_\phi$. As in Lemma 17, we introduce the measurement vector notation $\boldsymbol{v}_i \in \{-1, 0, 1\}^n$ in order to represent each drawn sample $((x_i, y_i), q_i)$ (recall that the edges in this training set lie in $\mathcal{E}$ and $q_i \in \{0, 1\}$). We set $\boldsymbol{v}_i(x_i) = 2q_i - 1, \boldsymbol{v}_i(y_i) = -\boldsymbol{v}_i(x_i)$ and the other coordinates are set to 0. These measurement vectors are the building blocks of the empirical matrix (see Lemma 17), since we have that

$$\boldsymbol{L} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{v}_i \boldsymbol{v}_i^T := \frac{1}{N} \boldsymbol{X}^T \boldsymbol{X} .$$

[6] As we mentioned, our learning algorithm almost exactly follows the algorithm of [SBB+16]. The only difference appears in the *second* constraint of the set Ω_ϕ, where [SBB+16] optimize over vectors with upper bounded L_∞ norm.

[7] A function $f : \mathcal{X} \to \mathbb{R}$ is κ-strongly convex with respect to a norm $\| \cdot \|$ if, for all x, y in the relative interior of the domain of f and $\lambda \in (0, 1)$, we have

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y) - \frac{1}{2} \kappa \lambda (1 - \lambda) \|x - y\|^2 .$$

If the function f is differentiable, then a second definition for κ-strong convexity with respect to a norm $\| \cdot \|$ is that for all points x, y, we have that $f(y) - f(x) - \langle \nabla f(x), y - x \rangle \geq \kappa \|x - y\|^2$. Recall that the relative interior of a set S is defined as its interior within the affine hull of S, i.e., $\mathrm{relint}(S) = \{x \in S : \exists \epsilon > 0, \mathcal{N}_\epsilon(x) \cap \mathrm{aff}(S) \subseteq S\}$.

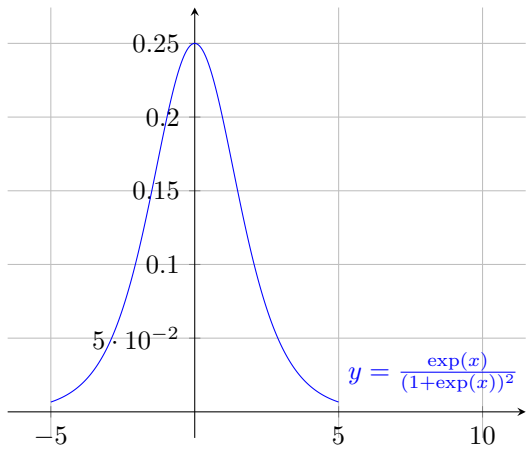

Figure 3: The second derivative of the negative logarithm of the Bradley-Terry function F.

Note that the matrix $\boldsymbol{X} \in \{-1, 0, 1\}^{N \times n}$ has the i-th measurement vector $\boldsymbol{v}_i^T$ as its i-th row. Using this notation and following the computations of [SBB+16] for the Hessian of L_N, we get that

$$\nabla^2 L_N(\boldsymbol{z}) = \frac{1}{N} \sum_{i=1}^{N} \Big\{ \mathbf{1}\{v_i = 1\} A_1 + \mathbf{1}\{v_i = 0\} A_0 \Big\} \boldsymbol{v}_i \boldsymbol{v}_i^T \,,$$

where

$$A_1 = (\log F)''(\langle \boldsymbol{z}, \boldsymbol{v}_i \rangle), A_0 = (\log(1-F))''(\langle \boldsymbol{z}, \boldsymbol{v}_i \rangle) \text{ and } F(x) = 1/(1+\exp(-x)) \,.$$

The function $F(x) = 1/(1+\exp(-x))$ is the Bradley-Terry function. The function F is strongly-log concave in the interval $[-\log(\phi), \log(\phi)]$. Specifically, we have that

$$\frac{d^2}{dx^2}\Big(-\log F(x) \Big) = \frac{e^x}{(1+e^x)^2} \geq \frac{\phi}{(1+\phi)^2} =: \kappa(\phi) \,,$$

since the mapping $x \mapsto \exp(x)/(1+\exp(x))^2$ is symmetric and, so, focusing on the interval $[0, \log(\phi)]$, its minimum is attained at $\log(\phi)$.

The physical interpretation of these properties is that for the pair i, j, it does not matter whether $i \succ j$ or $j \succ i$ (symmetry) but if the comparison gap (i.e., $|z_i - z_j|$) attains very large values, then the strong convexity will be very low. Intuitively, note also that, since $\exp(x)/(1+\exp(x))^2 \leq 1/4$, the desired objective is also smooth and it is known that strong convexity and smoothness of a function over a domain of the form $\{\boldsymbol{z} \in \mathbb{R}^n : \langle \mathbf{1}, \boldsymbol{z} \rangle = 0\} \cap \mathcal{Z}$, where $\mathcal{Z}$ is convex, imply that the function satisfies the PL inequality over $\mathcal{Z}$. Hence, for this function, gradient-based methods will converge with fast rates (see e.g., [VYZ20]).

Since any $\boldsymbol{z} \in \Omega_\phi$, we conclude that for any $\boldsymbol{w}$

$$\boldsymbol{w}^T \nabla L_N(\boldsymbol{z}) \boldsymbol{w} \geq \frac{\kappa(\phi)}{N} \|\boldsymbol{X}\boldsymbol{w}\|_2^2 \,,$$

where $\boldsymbol{X}$ the above measurement matrix and $\boldsymbol{z} \in \Omega_\phi$. Hence, if we define $\boldsymbol{\Delta} = \widehat{\boldsymbol{z}} - \boldsymbol{z}^\star$, we get (in order to form the definition of strong convexity) that

$$L_N(\boldsymbol{z}^\star + \boldsymbol{\Delta}) - L_N(\boldsymbol{z}^\star) - \langle \nabla L_N(\boldsymbol{z}^\star), \boldsymbol{\Delta} \rangle \geq \frac{\kappa(\phi)}{N} \|\boldsymbol{X}\boldsymbol{\Delta}\|_2^2 = \kappa(\phi) \|\boldsymbol{\Delta}\|_L^2 \,,$$

since $\|\boldsymbol{\Delta}\|_L^2 = \boldsymbol{\Delta}^T \boldsymbol{L} \boldsymbol{\Delta} = \frac{1}{N} \boldsymbol{\Delta}^T \boldsymbol{X}^T \boldsymbol{X} \boldsymbol{\Delta} = \frac{1}{N} \|\boldsymbol{X}\boldsymbol{\Delta}\|_2^2$. Hence, the empirical negative log-likelihood objective is $\kappa(\phi)$-strongly convex at the true natural parameter vector $\boldsymbol{z}^\star$ with respect to the $\boldsymbol{L}$ semi-norm. □

From the strong-convexity guarantee and, using Fact 3, we get that

$$\|\boldsymbol{\Delta}\|_L^2 \leq \frac{1}{\kappa(\phi)} \|\nabla L_N(\boldsymbol{z}^\star)\|_{L^\dagger} \,. \tag{8}$$

Claim 9 (Dual norm)**.** *Let $\boldsymbol{L}$ be the empirical Laplacian estimate matrix of $\boldsymbol{Q}$ and let $\boldsymbol{L}^\dagger$ the Moore-Penrose pseudo-inverse of $\boldsymbol{L}$. There exists a random vector $\boldsymbol{Y} \in \mathbb{R}^n$ such that*

$$\|\nabla L_N(\boldsymbol{z}^\star)\|_{L^\dagger} = \frac{1}{N^2}(\boldsymbol{X}^T\boldsymbol{Y})^T\boldsymbol{L}^\dagger(\boldsymbol{X}^T\boldsymbol{Y})\,,$$

where $\boldsymbol{X}$ is the $N \times n$ measurements matrix.

Proof. We first provide a compact form the the gradient $\nabla L_N(\boldsymbol{z}^\star)$. Again, following the analysis of [SBB+16], we define a random vector $\boldsymbol{Y} \in \mathbb{R}^n$ with independent coordinates $(Y_1, \ldots, Y_n)$ where we set $Y_i = F'(\langle \boldsymbol{z}^\star, \boldsymbol{v}_i\rangle)/F(\langle \boldsymbol{z}^\star, \boldsymbol{v}_i\rangle)$ with probability $F(\langle \boldsymbol{z}^\star, \boldsymbol{v}_i\rangle)$ and we set $Y_i = -F'(\langle \boldsymbol{z}^\star, \boldsymbol{v}_i\rangle)/(1 - F(\langle \boldsymbol{z}^\star, \boldsymbol{v}_i\rangle))$ otherwise. Recall that F, in our setting, is the Bradley-Terry function $F(x) = 1/(1 + \exp(-x))$ and $\boldsymbol{v}_i \in \{-1, 0, 1\}^n$ is the i-th comparison vector indicating the drawn edge. Hence, after computing the gradient at the true parameter vector $\boldsymbol{z}^\star$, one gets

$$\nabla L_N(\boldsymbol{z}^\star) = -\frac{1}{N}\boldsymbol{X}^T\boldsymbol{Y}\,.$$

Claim 9 follows. □

We now introduce the matrix $\boldsymbol{J} := \frac{1}{\kappa(\phi)N^2}\boldsymbol{X}\boldsymbol{L}^\dagger\boldsymbol{X}^T$. Combining Equation (8) with Claim 9, we get that

$$\|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_L^2 \le \boldsymbol{Y}^T\boldsymbol{J}\boldsymbol{Y}\,.$$

Hence, in order to bound the $\boldsymbol{L}$ semi-norm estimation error, it suffices to control the quadratic form of the right hand side. Similarly to [SBB+16], our goal is to apply the Hanson-Wright inequality (see Lemma 19) in order to control the quadratic form.

Lemma 19 ([HW71, RV13])**.** *Let $\boldsymbol{Y} \in \mathbb{R}^n$ be a random vector with independent zero-mean components, which are sub-Gaussian with parameter K and let $\boldsymbol{J} \in \mathbb{R}^{n\times n}$ be an arbitrary matrix. Then there is a universal constant $c > 0$ such that*

$$\Pr\left[\left|\boldsymbol{Y}^T\boldsymbol{J}\boldsymbol{Y} - \mathbb{E}[\boldsymbol{Y}^T\boldsymbol{J}\boldsymbol{Y}]\right| > t\right] \le 2\exp\left(-c\min\left\{\frac{t^2}{K^4\|\boldsymbol{J}\|_F^2}, \frac{t}{K^2\|\boldsymbol{J}\|_2}\right\}\right)\,,$$

for any $t > 0$.

Observe that $\mathbb{E}[\boldsymbol{Y}] = 0$ and, for any $i \in [n]$, we have that

$$|Y_i| \le \sup_{\boldsymbol{z}\in\Omega_\phi}\sup_{\boldsymbol{x}\in\mathbb{X}}\max\left\{\frac{F'(\langle \boldsymbol{z}, \boldsymbol{x}\rangle)}{F(\langle \boldsymbol{z}, \boldsymbol{x}\rangle)}, \frac{F'(\langle \boldsymbol{z}, \boldsymbol{x}\rangle)}{1 - F(\langle \boldsymbol{z}, \boldsymbol{x}\rangle)}\right\}\,,$$

where $\mathbb{X} \subseteq \{-1, 0, 1\}^n$ is the space of all valid measurement vectors (they should induce edges in $\mathcal{E}$). Hence, we get that

$$|Y_i| \le \max_{x\in[-\log(\phi),\log(\phi)]}\max\left\{\frac{\exp(-x)/(1+\exp(-x))^2}{1/(1+\exp(-x))}, \frac{\exp(-x)/(1+\exp(-x))^2}{\exp(-x)/(1+\exp(-x))}\right\} =: u(\phi)\,.$$

Note that $u(\phi) \le 1$, since, for $0 < b < 1$, it holds that $\max\left\{\frac{1}{b}, \frac{1}{1-b}\right\} \le \frac{1}{b(1-b)}$. Since each $|Y_i|$ is bounded, we get that the random variables are $u(\phi)$-sub-Gaussian. Also, we have that

$$\mathbb{E}[\boldsymbol{Y}^T\boldsymbol{J}\boldsymbol{Y}] \le \mathbb{E}\Big[\|\boldsymbol{Y}\|_\infty^2\mathrm{tr}(\boldsymbol{J})\Big] \le \frac{u(\phi)^2}{\mathrm{poly}(\kappa(\phi))}\frac{n}{N}\,.$$

It remains to compute the two matrix norms. As in [SBB+16], one gets that

$$\|\boldsymbol{J}\|_F^2 = \sum_{i,j}|\boldsymbol{J}_{ij}|^2 = \frac{n-1}{\mathrm{poly}(\kappa(\phi))N^2} = \Theta(n/N^2)\,,$$

and

$$\|\boldsymbol{J}\|_2 = \max_{\boldsymbol{x}\neq\boldsymbol{0}}\|\boldsymbol{J}\boldsymbol{x}\|_2/\|\boldsymbol{x}\|_2 = \frac{1}{\mathrm{poly}(\kappa(\phi))N} = \Theta(1/N)\,.$$

Applying the Hanson-Wright concentration inequality, we conclude that

$$\Pr\left[\|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_L^2 > c\cdot t\frac{u(\phi)^2}{\mathrm{poly}(\kappa(\phi))}\frac{n}{N}\right] \le e^{-t}\,,$$

for all $t \ge 1$ and some universal constant $c > 0$. Recall that ϕ is the constant of Assumption 2 and hence the term $\frac{u(\phi)^2}{\mathrm{poly}(\kappa(\phi))} \in \Theta(1)$. Finally, note that integrating the above tail bound, one gets the desired on expectation bound. □

F.2 Application of Algorithm 6 to Distributions

Given a target distribution $\mathcal{D}$, the associated natural parameters vector $\boldsymbol{z} = \log(\mathcal{D})$ satisfies $\langle \mathbf{1}, \boldsymbol{z} \rangle \leq 0$. Hence, we have to shift it appropriately in order to apply Algorithm 6. Recall that the condition $\langle \mathbf{1}, \boldsymbol{z} \rangle = 0$ is only useful for identifiability purposes of the BTL model.

Given a target distribution $\mathcal{D} \in \Delta^n$, we convert it to the following weights vector: let $A = \sum_{x \in [n]} \log(\mathcal{D}(x))$ and set $\mathcal{D}'(x) = \exp(-A/n)\mathcal{D}(x)$. So, we have to shift $\boldsymbol{z}$ in the direction $(-A/n) \cdot \mathbf{1}$ and obtain the weight vector $\boldsymbol{z}'$. For the weights $(\mathcal{D}'(x))_{x \in [n]}$ with weight vector $\boldsymbol{z}'$, it holds that

1. $\langle \mathbf{1}, \boldsymbol{z}' \rangle = \sum_{x \in [n]} (-A/n + \log(\mathcal{D}(x))) = 0$ and
2. for any $x, y \in [n]$, it holds that $\mathcal{D}'(x)/\mathcal{D}'(y) = \mathcal{D}(x)/\mathcal{D}(y)$.

Given an estimate for $\boldsymbol{z}'$, one can extract a good approximation for the natural parameter vector $\boldsymbol{z}$ of $\mathcal{D}$. In particular, the following properties hold: first, given two distributions $\boldsymbol{p} \neq \boldsymbol{q}$, it holds that $\boldsymbol{z}'(\boldsymbol{p}) \neq \boldsymbol{z}'(\boldsymbol{q})$, i.e., any $\boldsymbol{z}'$ is uniquely identified by its distribution since one of the following will hold: if $\sum_{i \in [n]} \log(p_i) = A = \sum_{i \in [n]} \log(q_i)$, then since there exists $j \in [n]$ so that $p_j \neq q_j$, we have that $z(\boldsymbol{p})'_j = \log(p_j) - A/n \neq \log(q_j) - A/n = z(\boldsymbol{q})'_j$. Otherwise, if, without loss of generality, $A_p = \sum_i \log(p_i) > \sum_i \log(q_i) = A_q$, we have that there exists $j \in [n]$ so that $p_j < q_j$ and so $z(\boldsymbol{p})'_j = \log(p_j) - A_p/n < \log(q_j) - A_q/n = z(\boldsymbol{q})'_j$. Hence, any distribution $\mathcal{D}$ uniquely induces a vector $\boldsymbol{z}'$. Moreover, the actual natural parameter vector $\boldsymbol{z}$ can be estimated having an estimation for $\boldsymbol{z}'$. Assume that we have an estimate $\widehat{\boldsymbol{z}'}$ that satisfies $\|\boldsymbol{z}' - \widehat{\boldsymbol{z}'}\|_2 \leq \epsilon$. We can estimate $\boldsymbol{z}$ using a vector $\widehat{\boldsymbol{z}}$ (that can be extracted from our estimate $\widehat{\boldsymbol{z}'}$) as follows: Let $\boldsymbol{z}' = \boldsymbol{z} + C\mathbf{1}$, where $C = -A/n$. In order to initiate our exact sampling algorithm, an L_∞ estimate for $\boldsymbol{z}$ is only required. We have that

$$\|\boldsymbol{z} - \widehat{\boldsymbol{z}}\|_\infty = \|(\boldsymbol{z}' - C\mathbf{1}) - (\widehat{\boldsymbol{z}'} - C'\mathbf{1})\|_\infty \leq \|\boldsymbol{z}' - \widehat{\boldsymbol{z}'}\|_\infty + |C - C'|\,,$$

where the estimate $\widehat{\boldsymbol{z}'}$ satisfies $\|\boldsymbol{z}' - \widehat{\boldsymbol{z}'}\|_\infty \leq \|\boldsymbol{z}' - \widehat{\boldsymbol{z}'}\|_2 \leq \epsilon$ via Algorithm 6 and the shift constant can be estimated since we know that $\mathcal{D}$ is a distribution:

$$\sum_{x \in [n]} \exp(\widehat{z}_x) = 1 \iff \sum_{x \in [n]} e^{\widehat{z'}_x - C'} = 1 \iff e^{C'} = \sum_{x \in [n]} e^{\widehat{z'}_x}\,.$$

Note that the right hand side of the above equation contains only our estimates. We have that $\sum_{x \in [n]} \exp(z_x) = 1$ and so we set $e^C := \sum_{x \in [n]} e^{z'_x}$. Since it holds that $|z'_x - \widehat{z'}_x| \leq \epsilon$ for any $x \in [n]$, we have that $|C - C'| \leq O(\epsilon)$. This gives that for the actual natural parameters it holds that $\|\boldsymbol{z} - \widehat{\boldsymbol{z}}\|_\infty \leq O(\epsilon)$.

Hence, given a distribution $\mathcal{D}$, we can learn the weights $\boldsymbol{z}'$ in L_2 norm, satisfying the conditions of Algorithm 6. From this estimation, we can get the estimation for the natural parameters $\boldsymbol{z}$ by removing the introduced shift. We remark that the above shifting methodology can be applied to any score vector $\boldsymbol{z}$ satisfying $\langle \mathbf{1}, \boldsymbol{z} \rangle = B \neq 0$.

G Local Sampling Schemes and Hypergraphs

In this section, we discuss how to extend our analysis to sets of size larger that 2. The hypergraph structure of Local Sampling Schemes can be settled as follows:

Definition 20 (Hypergraph Structure of LSS)**.** *Let $\mathcal{Z}$ be a finite discrete domain and let $\mathcal{Q}$ be a distribution supported on subsets of $\mathcal{Z}$. Then, the hypergraph $G = (V, E)$ with vertex set $V = \mathcal{Z}$ and hyperedge set $E = \mathrm{supp}(\mathcal{Q})$ is called a Local Sampling Scheme hypergraph. If $\mathcal{Q}$ is a pair distribution supported on $\mathcal{Z} \times \mathcal{Z}$, G corresponds to a graph.*

The general Markov Chain for distributions $\mathcal{Q}$, supported on sets $S \subseteq 2^{[n]}$ has the following transition probabilities:

$$p_{xy} = \sum_{S \supseteq \{x,y\}} \mathcal{Q}(S) \frac{\mathcal{D}(y)}{\mathcal{D}(S)} = \mathcal{D}(y) \sum_{S \supseteq \{x,y\}} \frac{\mathcal{Q}(S)}{\mathcal{D}(S)} \,. \tag{9}$$

For the transition $x \to y$, we can think of a flow f_{xy} with mass p_{xy} and, hence, induce a (simple) graph structure on a flow graph $F = (f_{xy})_{xy}$. Observe that any pair (x, y) shares the same collection of hyperedges and, hence, we have that:

$$\frac{p_{xy}}{p_{yx}} = \frac{\mathcal{D}(y)}{\mathcal{D}(x)} \frac{\sum_{S \supseteq \{x,y\}} \mathcal{Q}(S)/\mathcal{D}(S)}{\sum_{S \supseteq \{x,y\}} \mathcal{Q}(S)/\mathcal{D}(S)} = \frac{\mathcal{D}(y)}{\mathcal{D}(x)} \,.$$

The ergodicity of the flow graph implies that the stationary distribution corresponds to $\mathcal{D}$ and is unique. For what follows, we consider the case where any hyperedge has size k, i.e., k-uniform hypergraphs. The Laplacian of the distribution $\mathcal{Q}$ can be generalized over hyperedges and have that

$$\boldsymbol{Q}_{xy} = -\sum_{S \supseteq \{x,y\}} \mathcal{Q}(S) \;\text{ for any } x, y \in [n] \;\text{ and } \boldsymbol{Q}_{xx} = (k-1) \sum_{S \ni x} \mathcal{Q}(S) \;\text{ for any } x \in [n] \,.$$

In the following section, we describe some necessary notation.

G.1 Notation

We denote the set $\binom{[n]}{k}$ the family of size k subsets of the ground set $[n]$. Let $\mathcal{Q}$ be a distribution on $\binom{[n]}{k}$. Such a distribution will be called a k-set distribution. Recall that a pair distribution is simply a 2-set distribution. A sample from the Local Sampling Scheme $\mathrm{Samp}_k(\mathcal{Q}; \mathcal{D})$, where $\mathcal{Q}$ is a k-set distribution, will be denoted

$$(S, \boldsymbol{v}) \sim \mathrm{Samp}_k(\mathcal{Q}; \mathcal{D}) \,,$$

where $S \in \mathrm{supp}(\mathcal{Q})$ and, in the case where $i \in S$ won between the elements of S, we have that $\boldsymbol{v} = \boldsymbol{e}_i \in \mathbb{R}^n$ ($\boldsymbol{v}$ is the indicator vector of the winning node) and we have that $\Pr[\boldsymbol{v} = \boldsymbol{e}_i] = \mathcal{D}(i)/\mathcal{D}(S)\mathbf{1}\{i \in S\}$. For the hypergraph case, the canonical transition matrix induced by the Local Sampling Scheme will be denoted by $\boldsymbol{P}$. In the case $k = 2$, this matrix was denoted by $\boldsymbol{M}$ (recall Equation (1)). Let $\boldsymbol{P}$ denote the transition matrix of the Markov chain, associated with the Local Sampling Scheme $\mathrm{Samp}_k(\mathcal{Q}; \mathcal{D})$, where $\mathcal{Q}$ corresponds to a k-set distribution. The entries of $\boldsymbol{P} = [\boldsymbol{P}_{xy}]_{x,y \in [n]}$ are defined as:

$$\boldsymbol{P}_{xy} = \sum_{S \supseteq \{x,y\}} \mathcal{Q}(S) \frac{\mathcal{D}(y)}{\mathcal{D}(S)} \;\text{ for } x \neq y \;\text{ and } \boldsymbol{P}_{xx} = 1 - \sum_{y \neq x} \boldsymbol{P}_{xy} \;\text{ for } x \in [n] \,. \tag{10}$$

Observe that the transition from x to y is performed when

1. a hyperedge $S \in \mathrm{supp}(\mathcal{Q})$ is chosen (with probability $\mathcal{Q}(S)$) and both x and y lie in S,
2. and the vertex $y \in [n]$ is the 'winning' node among the nodes of S, i.e., y is drawn from the conditional distribution $\mathcal{D}_S$.

There is a natural reduction to the graph case. Consider the marginals of $\mathcal{Q}$ to 2-sets (i.e., edges). Then, one can define a Markov chain over a graph with $[n]$ nodes, whose transition probabilities are described by $\boldsymbol{P}$. The random walk has the following properties: Since any pair of vertices shares the same collection of hyperedges, we get that

$$\frac{\boldsymbol{P}_{xy}}{\boldsymbol{P}_{yx}} = \frac{\mathcal{D}(y)}{\mathcal{D}(x)} \frac{\sum_{S \supseteq \{x,y\}} \mathcal{Q}(S)/\mathcal{D}(S)}{\sum_{S \supseteq \{x,y\}} \mathcal{Q}(S)/\mathcal{D}(S)} = \frac{\mathcal{D}(y)}{\mathcal{D}(x)} \,,$$

for any pair of vertices. Moreover, the Markov chain is ergodic, since it is irreducible, since by the structure of $\mathcal{Q}$ and since $\mathcal{D}$ is supported on $[n]$, one eventually can get from every state to every other state with positive probability; and aperiodic, since it contains self-loops. Hence, it has a unique stationary distribution which coincides with $\mathcal{D}$, using the detailed balance equations.

Modifications of the LSS Conditions. We can modify the information-theoretic connectivity condition (Assumption 1) for k-uniform hypergraphs and let $\mathcal{E}$ the support of $\mathcal{Q}$. Also, we have to consider the variation of Assumption 2 over the support $\mathcal{E}$ with

$$1/\phi \leq \max_{(x,y)\in S\in\mathcal{E}} \mathcal{D}(x)/\mathcal{D}(y) \leq \phi\,,$$

for some constant ϕ. We remark that the learning results of [SBB+16] hold for $k = O(1)$. Similar, our learning tools which are modifications of the work of [SBB+16] hold for the same regime.

G.2 Learning Phase for Hypergraphs

As in the learning phase of the 2-set case, the learning algorithm for the k-set problem is essentially a variation of the analysis of [SBB+16], but the steps are similar. Consider N i.i.d. samples $(S_i, \boldsymbol{v}_i)$ drawn from the sampling oracle $\mathrm{Samp}_k(\mathcal{Q};\mathcal{D})$ with natural parameter vector $\boldsymbol{z}^\star \in \mathbb{R}^n$. The analysis that follows assumes that any k-set sample in the empirical likelihood lies in $\mathcal{E}$, i.e., the support of $\mathcal{Q}$. Our goal is to estimate the true parameter vector. We consider the negative empirical log-likelihood objective

$$L_N(\boldsymbol{z};\{S_i,\boldsymbol{v}_i\}_{i\in[N]}) = -\frac{1}{N}\sum_{i=1}^{N}\langle \boldsymbol{z},\boldsymbol{v}_i\rangle - \log\sum_{j\in S_i}\exp(z_j)\,,$$

and we optimize it over the parameter space

$$\Omega_\phi = \{\boldsymbol{z}\in\mathbb{R}^n : \langle \mathbf{1},\boldsymbol{z}\rangle = 0,\ \max_{(x,y)\in S\in\mathcal{E}} |z_x - z_y| \leq \log(\phi)\}\,.$$

Likelihood Objective and PL model. We now observe that the above likelihood objective is directly connected to the Plackett-Luce model, which captures the process of choosing a single alternative from a given set, i.e., given a set S of m alternatives with values $w_1,\ldots,w_m$, the likelihood of choosing the i-th item is

$$F(w_i, w_1, ..., w_{i-1}, w_{i+1}, ..., w_m) = \exp(w_i)/\sum_{j=1}^{m}\exp(w_j)\,.$$

Note that the negative log-likelihood of the Plackett-Luce model is exactly the same as the single sample version of our objective function. Observe that this function is shift-invariant and its value is independent of the ordering of the last $(m-1)$ elements. Shift invariance is crucial: if one does not work in the subspace $\{\boldsymbol{z} : \langle \mathbf{1},\boldsymbol{z}\rangle = 0\}$, then neither our problem nor the problem of determining the values of the alternatives in the Plackett-Luce model are identifiable, since any solution of the form $\boldsymbol{z}^\star + c\mathbf{1}$ is valid. Technically, shift invariance implies that $\mathbf{1}$ lies in the nullspace of the Hessian of the negative log-likelihood (and this is where the first constraint of the set Ω_ϕ arises).

Hessian matrix. Let us introduce the vector $\boldsymbol{e}^z(S) = (\exp(z_i))_{i\in S} \in \mathbb{R}^k$ for an arbitrary set $S \subseteq \binom{[n]}{k}$. After standard computations (see also [SBB+16]), we get that

$$\nabla^2_{\boldsymbol{z}}\Big(-\langle \boldsymbol{z},\boldsymbol{v}\rangle + \log\sum_{j\in S}\exp(z_j)\Big) = \frac{\Big(\sum_{j\in S}\exp(z_j)\Big)\mathrm{diag}(\boldsymbol{e}^z(S)) - \boldsymbol{e}^z(S)\boldsymbol{e}^z(S)^T}{\Big(\sum_{j\in S}\exp(z_j)\Big)^2}\,,$$

and hence

$$\nabla^2_{\boldsymbol{z}} L_N(\boldsymbol{z};\{S_i,\boldsymbol{v}_i\}_{i\in[N]}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\langle \boldsymbol{e}^z(S),\mathbf{1}\rangle\mathrm{diag}(\boldsymbol{e}^z(S_i)) - \boldsymbol{e}^z(S_i)\boldsymbol{e}^z(S_i)^T}{\Big(\langle \boldsymbol{e}^z(S_i),\mathbf{1}\rangle\Big)^2}\,.$$

Quantitative Strong Convexity of PL model. Consider an arbitrary direction $\boldsymbol{v} \in \mathbb{R}^k$. Without loss of generality, assume that $S = [k]$. Then, it holds that

$$\boldsymbol{v}^T \left(\boldsymbol{e}^z(S) \boldsymbol{e}^z(S)^T \right) \boldsymbol{v} = \sum_{i,j=1}^{k} v_i v_j \exp(z_i + z_j) \leq \sum_{i=1}^{k} \exp(z_i) \sum_{j=1}^{k} v_j^2 \exp(z_j) = \langle \boldsymbol{e}^z(S), \mathbf{1} \rangle \mathrm{diag}(\boldsymbol{e}^z(S)) \,,$$

using the Cauchy-Schwarz inequality, where equality holds if and only if $\boldsymbol{v} \in \mathrm{span}(\mathbf{1})$. Hence, for the Plackett-Luce function $F : \mathbb{R}^k \to \mathbb{R}$, we have that

$$\lambda_2(-\log(F(\boldsymbol{w}))) > 0, \quad \text{for any} \quad \boldsymbol{w} \in \Omega_\phi \,.$$

In order to quantify the strong convexity parameter, we will make use of the second constraint. For the Plackett-Luce function F, we have that

$$\nabla^2(-\log(F(\boldsymbol{w})) \succeq \boldsymbol{H} \,,$$

for some $k \times k$ symmetric matrix $\boldsymbol{H}$ with $\lambda_2(\boldsymbol{H}) > 0$. We can observe that the matrix

$$\boldsymbol{H} = \beta(\phi)(\boldsymbol{I} - \mathbf{1}\mathbf{1}^T) \,,$$

where $\beta(\phi) = \min_{\boldsymbol{z} \in \Omega_\phi} \lambda_2 \left(\frac{\langle \boldsymbol{e}^z(S), \mathbf{1} \rangle \mathrm{diag}(\boldsymbol{e}^z(S)) - \boldsymbol{e}^z(S) \boldsymbol{e}^z(S)^T}{\left(\langle \boldsymbol{e}^z(S), \mathbf{1} \rangle \right)^2} \right)$, satisfies the strong log-concavity condition for the Plackett-Luce model, i.e., the function $F : \mathbb{R}^k \to \mathbb{R}$.

Underlying Laplacian matrix and estimation. Similarly to the case $k = 2$, our goal is to establish strong convexity for the empirical negative log-likelihood around the true parameters $\boldsymbol{z}^\star \in \Omega_\phi$ with respect to the $\boldsymbol{L}$ semi-norm. Hence, we have to first introduce the appropriate Laplacian matrix $\boldsymbol{L}$. In the k-set setting, we have that

$$\boldsymbol{Q}_{x,y} = -\sum_{S \ni x,y} \mathcal{Q}(S) \text{ for any } x, y \in [n], \; \boldsymbol{Q}_{x,x} = (k-1) \sum_{S \ni x} \mathcal{Q}(S) \text{ for any } x \in [n] \,.$$

Observe that $\boldsymbol{Q}\mathbf{1} = 0$ (since each set $S \subseteq \binom{[n]}{k}$ that contains x, also contains other $k-1$ elements). Recall that for the case $k = 2$, the estimate for the Laplacian matrix $\boldsymbol{Q}$ is given by the empirical matrix $\boldsymbol{L} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{v}_i \boldsymbol{v}_i^T = \frac{1}{N} \boldsymbol{X}^T \boldsymbol{X}$ (see Appendix F.1). In the case $k > 2$ case, the Laplacian estimate for the matrix $\boldsymbol{Q}$ is given by the $n \times n$ matrix

$$\boldsymbol{L} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{E}_i (k\boldsymbol{I} - \mathbf{1}\mathbf{1}^T) \boldsymbol{E}_i^T \,,$$

where the vectors $\boldsymbol{v}_i$ are substituted by the $(n \times k)$ matrices $\boldsymbol{E}_i$, where each one of the k columns of $\boldsymbol{E}_i$ is a unit vector. The k non-zero elements of $\boldsymbol{E}_i$ correspond to the k elements that lie in the i-th drawn set S_i, i.e., $\boldsymbol{E}_i = \left[\boldsymbol{e}_{i_1} \middle| \boldsymbol{e}_{i_2} \middle| \ldots \middle| \boldsymbol{e}_{i_k} \right]$ for $S_i = \{i_1, i_2, \ldots, i_k\}$.

In the estimation part[8], our goal is to estimate the unknown matrix $\boldsymbol{Q}$. Using concentration results of the Fiedler eigenvalue of sums of Hermitian matrices [Tro15], using $O(\log(n)/\lambda(\boldsymbol{Q}))$ samples, one can compute an estimate $\boldsymbol{L}$ (that corresponds to the above empirical estimate) that approximates the true $\lambda(\boldsymbol{Q})$ with $(1 + \epsilon)$ multiplicative error.

We have that

$$\boldsymbol{L} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{X}_i \,,$$

[8]Recall that the learning algorithm does not require the knowledge of the Laplacian matrix $\boldsymbol{Q}$. The concentration result for the empirical estimation of the matrix $\boldsymbol{Q}$ using the matrix $\boldsymbol{L}$ enables us to express our sample complexity results using the second smallest eigenvalue of the true (population) matrix $\boldsymbol{Q}$ and not its empirical estimate.

where[9] $\boldsymbol{X}_i = \boldsymbol{E}_i(k\boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T)\boldsymbol{E}_i^T$, i.e., the estimate $\boldsymbol{L}$ is a sum of independent Hermitian matrices with $\lambda_{max}(\boldsymbol{X}_i) = \Theta(k)$. Using Proposition 7 in a similar manner as in Lemma 17, we get that

$$\Pr_{\boldsymbol{X}_{1..m} \sim \mathcal{Q}^{\otimes m}} \left[\lambda(\boldsymbol{L}) \leq (1-\epsilon)\lambda(\boldsymbol{Q})\right] \leq n \exp\left(-\epsilon^2 m \frac{\lambda(\boldsymbol{Q})}{2k}\right) .$$

Hence, it suffices to draw at least

$$m = O\left(\frac{k}{\epsilon^2 \cdot \lambda(\boldsymbol{Q})} \log(n/\delta)\right)$$

in order to guarantee the desired concentration bound with confidence at least $1 - \delta$.

Strong Convexity of Empirical NLL w.r.t. $\|\cdot\|_L$. Having introduced the empirical Laplacian matrix, we are able to get the desired strong convexity result for the empirical negative log-likelihood with respect to the $\boldsymbol{L}$ semi-norm. Following the analysis of [SBB+16], we get that, for any vector $\boldsymbol{w} \in \mathbb{R}^d$ and parameter $\boldsymbol{z} \in \Omega_\phi$,

$$\boldsymbol{w}^T \nabla^2 L_N(\boldsymbol{z}) \boldsymbol{w} \geq \frac{\lambda_2(\boldsymbol{H})}{k} \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} \boldsymbol{1}\{\boldsymbol{v}_i = \boldsymbol{e}_j\} \boldsymbol{w}^T \boldsymbol{E}_i (k\boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T) \boldsymbol{E}_i^T \boldsymbol{w} ,$$

where $\boldsymbol{H} = \beta(\phi)(\boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T)$. Hence, we get that

$$\boldsymbol{w}^T \nabla^2 L_N(\boldsymbol{z}) \boldsymbol{w} \geq \frac{\lambda_2(\beta(\phi)(\boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T))}{k} \|\boldsymbol{w}\|_L^2 .$$

Note that $\lambda_2(\boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T) = 1$. [10] Consequently, the empirical negative log-likelihood is $\beta(\phi)/k$-strongly convex around the true parameters $\boldsymbol{z}^\star \in \Omega_\phi$. Using Fact 3, we get that

$$\|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_L^2 \leq \frac{k^2}{\beta(\phi)^2} \|\nabla L_N(\boldsymbol{z}^\star)\|_{L^\dagger}^2 .$$

Bounding the (expected) dual norm $\|\cdot\|_{L^\dagger}$. Following the exact analysis of [SBB+16] for the expected value of the dual norm, we get that for our setting

$$\mathbb{E}\Big[\langle \nabla L_N(\boldsymbol{z}^\star), \boldsymbol{L}^\dagger \nabla L_N(\boldsymbol{z}^\star)\rangle\Big] \leq \frac{n}{N} \sup_{\boldsymbol{z} \in \mathbb{R}^k : |z_x - z_y| \leq \log(\phi)} \|\nabla \log F(\boldsymbol{z})\|_2^2 ,$$

where F is the Plackett-Luce function and assuming that $S = [k]$ (and S satisfies the modified mass condition, i.e., S lies in the support of $\mathcal{Q}$). We have that

$$\nabla_{\boldsymbol{z}} \log F(\boldsymbol{z}) = \boldsymbol{e}_i - \frac{(\exp(z_1), ..., \exp(z_k))^T}{\sum_{j \in [k]} \exp(\boldsymbol{z}_j)} ,$$

for some $i \in [k]$. Hence, we get that

$$\|\nabla_{\boldsymbol{z}} \log F(\boldsymbol{z})\|_2^2 = \left(1 - \frac{e^{z_i}}{\sum_{j \in [k]} e^{z_i}}\right)^2 + \frac{\sum_{j \in [k] \setminus \{i\}} e^{2z_j}}{(\sum_{j \in [k]} e^{z_j})^2} = \frac{(\sum_{j \in [k] \setminus \{i\}} \exp(z_j))^2 + \sum_{j \in [k] \setminus \{i\}} \exp(2z_j)}{(\sum_{j \in [k]} \exp(z_j))^2}$$

Over the optimization space Ω_ϕ and since the natural parameters $\boldsymbol{z}_x = \log(\mathcal{D}(x)) \leq 0$, we have that

$$\frac{1}{k\phi} \leq \frac{\mathcal{D}(x)}{k \max_{j \in S} \mathcal{D}(j)} \leq \frac{\mathcal{D}(x)}{\sum_{j \in S = [k]} \mathcal{D}(j)} \leq \frac{\mathcal{D}(x)}{k \min_{j \in S} \mathcal{D}(j)} \leq \phi/k$$

[9] For instance, let $n = 5, k = 3$ and $S = \{1, 2, 3\}$.

We have that $\boldsymbol{E} = [\boldsymbol{e}_1 | \boldsymbol{e}_2 | \boldsymbol{e}_3] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and $\boldsymbol{X} = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

[10] The characteristic polynomial of $\boldsymbol{A} = \boldsymbol{I} - \boldsymbol{1}\boldsymbol{1}^T$ is $\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = (\lambda - 1)^{n-1}(\lambda + n - 1)$.

Hence, we have that

$$\left(\frac{\sum_{j\in[k]\setminus\{i\}}\mathcal{D}(j)}{\sum_{j\in[k]}\mathcal{D}(j)}\right)^2 \leq ((1-1/k)\phi)^2\,,$$

and

$$\frac{\sum_{j\in[k]\setminus\{i\}}\mathcal{D}(j)^2}{(\sum_{j\in[k]}\mathcal{D}(j))^2} = \sum_{j\in[k]\setminus\{i\}}\frac{\mathcal{D}(j)}{\sum_{j\in[k]}\mathcal{D}(j)}\frac{\mathcal{D}(j)}{\sum_{j\in[k]}\mathcal{D}(j)} \leq (k-1)\frac{\phi^2}{k^2}\,.$$

Finally, we get that

$$\|\nabla_{\boldsymbol{z}}\log F(\boldsymbol{z})\|_2^2 \leq \phi^2\frac{k-1}{k}\,.$$

Conclusion of the Learning Phase in k-sets. We get that we can estimate the true parameters in $\boldsymbol{L}$ semi-norm, i.e.,

$$\|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_L^2 \leq \frac{k^2}{\beta(\phi)^2}\frac{n}{N}\phi^2\frac{k-1}{k}.$$

Hence, we can transfer this result to an L_2 bound and get

$$\|\boldsymbol{z}^\star - \widehat{\boldsymbol{z}}\|_2^2 \leq \frac{k^2}{\beta(\phi)^2}\frac{n}{N\lambda_2(\boldsymbol{L})}\phi^2\frac{k-1}{k}.$$

Using $N = O\left(\frac{nk^2}{\lambda_2(\boldsymbol{L})\beta(\phi)^2}\cdot\frac{1}{\epsilon^2}\right) =_\epsilon O\left(\frac{nk^2}{\lambda(\boldsymbol{Q})\beta(\phi)^2}\cdot\frac{1}{\epsilon^2}\right)$ samples from the oracle $\mathrm{Samp}_k(\mathcal{Q};\mathcal{D})$ on the support set $\mathcal{E}$, we can learn the parameter vector in L_2. At this point, the proof of the learning phase is similar to the $k=2$ case: We can apply the trick of Appendix F.2 and get an L_∞ bound. Hence, using Lemma 18, we can estimate the target distribution with relative error.

G.3 Downscaling Phase for Hypergraphs

Let $\widehat{\boldsymbol{z}}$ be the estimate of the natural parameters vector and let $\widetilde{\mathcal{D}}$ the correspond distribution. We use the estimate $\widetilde{\mathcal{D}}$ of the target distribution $\mathcal{D}$ to transform the Markov chain of the k-set Local Sampling Scheme into another almost uniform stationary distribution, as in the case $k=2$. Using the Bernoulli downscaling mechanism, for the pair (x,y), we have that

$$\widetilde{p}_{xy} = \mathcal{D}(y)\frac{\widetilde{\mathcal{D}}(x)}{\widetilde{\mathcal{D}}(y)}\sum_{S\ni x,y}\frac{\mathcal{Q}(S)}{\mathcal{D}(S)} \approx \mathcal{D}(x)\sum_{S\ni x,y}\frac{\mathcal{Q}(S)}{\mathcal{D}(S)} = p_{yx}\,.$$

We have that

$$\frac{\widetilde{p}_{xy}}{\widetilde{p}_{yx}} = \frac{\mathcal{D}(y)\frac{\widetilde{\mathcal{D}}(x)}{\widetilde{\mathcal{D}}(y)}\sum_{S\ni x,y}\frac{\mathcal{Q}(S)}{\mathcal{D}(S)}}{\mathcal{D}(x)\sum_{S\ni x,y}\frac{\mathcal{Q}(S)}{\mathcal{D}(S)}} = \frac{\mathcal{D}(y)/\widetilde{\mathcal{D}}(y)}{\mathcal{D}(x)/\widetilde{\mathcal{D}}(x)}\,.$$

The modified transition matrix (as in the $k=2$ case) has an almost uniform stationary distribution. Also, it has an absolute spectral gap of same order as the matrix $\widetilde{\boldsymbol{P}}$ that can be expressed as

$$\widetilde{\boldsymbol{P}}_{xy} = \left(\frac{1}{2}+\epsilon_{xy}\right)\sum_{S\ni x,y}\mathcal{Q}(S)\,,\;\; \widetilde{\boldsymbol{P}}_{yx} = \left(\frac{1}{2}+\epsilon_{yx}\right)\sum_{S\ni x,y}\mathcal{Q}(S)\,,$$

and $\widetilde{\boldsymbol{P}}_{xx} = 1-\sum_{y\neq x}\widetilde{\boldsymbol{P}}_{xy}$. Hence, we get that

$$\widetilde{\boldsymbol{P}} = \boldsymbol{I} - \frac{1}{2}\boldsymbol{Q} + \boldsymbol{Q}\circ[\epsilon_{xy}]\,.$$

The expression of the modified transition matrix $\widetilde{\boldsymbol{P}}$ is exactly similar to the $k=2$ case (see Equation (5)). Spectral analysis as in Lemma 15 of this matrix will result to the following properties:

- (i) The transition matrix $\widetilde{\boldsymbol{P}}$ has stationary distribution $\widetilde{\pi}_0(x) = \Theta(1/n)$ for all $x\in[n]$, i.e., an *almost* uniform probability measure.
- (ii) The absolute spectral gap $\Gamma(\widetilde{\boldsymbol{P}})$ of $\widetilde{\boldsymbol{P}}$ and the minimum non-zero eigenvalue $\lambda(\boldsymbol{Q})$ of the Laplacian matrix $\boldsymbol{Q}$ satisfy $\Gamma(\widetilde{\boldsymbol{P}}) = \Omega(\lambda(\boldsymbol{Q}))$.
- (iii) The mixing time of the transition matrix $\widetilde{\boldsymbol{P}}$ is $T_{\mathrm{mix}}(\widetilde{\boldsymbol{P}};1/4) = O(\log(n)/\lambda(\boldsymbol{Q}))$, where $\lambda(\boldsymbol{Q})$ is the minimum non-zero eigenvalue of the Laplacian matrix $\boldsymbol{Q}$.

G.4 CFTP Phase for Hypergraphs

The structure of the k-set algorithm remains similar to the $k = 2$ case, with the exception that the drawn samples are now hyperedges. We continue with the modified Algorithm 7. Recall that, for each drawn sample $(S, \boldsymbol{v}) \sim \mathrm{Samp}_k(\mathcal{Q}; \mathcal{D})$, the vector $\boldsymbol{v} \in \{\boldsymbol{e}_1, ..., \boldsymbol{e}_n\}$ is the indicator vector of the winning node of the set $S \subseteq \binom{[n]}{k}$.

Algorithm 7 Exact Sampling from k-set Local Sampling Schemes

procedure EXACTSAMPLER-k-SET($\boldsymbol{p}$) ▷ *Sample access to the LSS oracle* $\mathrm{Samp}_k(\mathcal{Q}; \mathcal{D})$.
  $t \leftarrow 0$
  $F_0(x) \leftarrow x$, for any $x \in [n]$
  while F_t has not coallesced **do** ▷ *While no coalescence has occured.*
    $t \leftarrow t - 1$
    Draw $(S, \boldsymbol{v})$ where $S \in \mathcal{E}$ and $\boldsymbol{v} = \boldsymbol{e}_i$ for some $i \in [n]$
    $y \leftarrow$ position where $\boldsymbol{v}[y] = 1$ ▷ *Find the winning node* $y \in S$.
    for $x = 1 \dots n$ **do** ▷ *In order to update state* x.
      if $x \notin S$ or $x = y$ **then** $F_t(x) \leftarrow F_{t+1}(x)$
      else

$$F_t(x) \leftarrow \begin{cases} F_{t+1}(w), & \text{with probability } \min\{p(x)/p(w), 1\} \\ F_{t+1}(x), & \text{otherwise} \end{cases}$$

    end
  end
  Draw $C \sim \mathrm{Be}(p(F_t(1)))$ ▷ *Remove bias using rejection sampling.*
  if $C = 1$ **then** Output $F_t(1)$ **else** Output $\perp$ ▷ *Output the perfect sample or reject.*
end procedure

procedure EXACTSAMPLER-k-SET-WITHLEARNING(δ) ▷ *The algorithm of Corollary 21.*
  $\widetilde{\mathcal{D}} \leftarrow$ LEARN($\epsilon := 1/\sqrt{n}, \delta$) ▷ *Learn* $\mathcal{D}$ *in relative error as in Appendix G.2.*
  x $\leftarrow \perp$
  while x $= \perp$ **repeat**
    x $\leftarrow$ EXACTSAMPLER-k-SET($\widetilde{\mathcal{D}}$)
  Output x
end procedure

G.5 Conclusion

Let N_{Learn} and N_{CFTP} be the expected number of samples required for a single execution of the learning algorithm for k-sets and of the parameterized CFTP algorithm respectively (see Algorithm 7). Under Assumption 1 (for the distribution $\mathcal{Q}$ over hyperedges) and the modified version of Assumption 2, the expected sample complexity of the generalized algorithm for the k-set case is

$$N_{\mathrm{Learn}}(\epsilon := 1/\sqrt{n}) + \Theta(n) \cdot N_{\mathrm{CFTP}} = O\left(\frac{k \log(n)}{\lambda(\boldsymbol{Q})} + \frac{n^2 k^2}{\lambda(\boldsymbol{Q})\beta(\phi)^2}\right) + \widetilde{O}\left(\frac{n^2}{\lambda(\boldsymbol{Q})}\right).$$

Note that $\beta(\phi)$ is constant under Assumption 2. As explained in [SBB+16], it is natural to consider the regime where $k = O(1)$. In this regime, the learning bounds match the CFTP dependence on n under $\widetilde{O}$ notation.

Corollary 21 (Exact Sampling from k-Set LSS)**.** *For any positive constant integer $k > 2$, under Assumption 1 (for the distribution $\mathcal{Q}$ over k-sets) and Assumption 2, there exists an algorithm (Algorithm 7) that draws an expected number of $\widetilde{O}\left(\frac{n^2 k^2}{\lambda(\boldsymbol{Q})}\right)$ samples from a k-Set Local Sampling Scheme $\mathrm{Samp}_k(\mathcal{Q}; \mathcal{D})$, and generates a sample distributed as in $\mathcal{D}$.*