Faster Spectral Sparsification of Laplacian and SDDM Matrix Polynomials*

Gorav Jindal[1] and Pavel Kolev[1]

1 **Max-Planck-Institut für Informatik, Saarbrücken, Germany**
`{gjindal,pkolev}@mpi-inf.mpg.de`

Abstract

We propose a simple, faster and unified algorithm that computes in nearly linear time a spectral sparsifier $D - \widehat{M}_N \approx_\varepsilon D - D\sum_{i=0}^{N}\gamma_i(D^{-1}M)^i$ of matrix-polynomials, where $B = D - M \in \mathbb{R}^{n\times n}$ is either Laplacian or SDDM matrix with m_B non-zero entries, the coefficients γ_i are induced by a mixture of discrete Binomial distributions (MDBD), and $N = 2^k$ for $k \in \mathbb{N}$.

Our sparsification algorithm runs in time $\widetilde{O}(\varepsilon^{-2}m_B\log^3 n\cdot\log^2 N + \varepsilon^{-4}\cdot nN\cdot\log^4 n\cdot\log^5 N)$[1] and exhibits a significant improvement over the previous best result of Cheng et al. [3] whose run time is $\widetilde{O}(\varepsilon^{-2}\cdot m_B N^2\cdot \mathrm{poly}(\log n))$. Although their algorithm handles general probability distributions, we demonstrate that MDBD approximate large class of continuous probability distributions. Moreover, we propose a nearly linear time algorithm that recovers exactly the coefficients of any convex combination of $N+1$ distinct Binomial distributions (MDBD), given as input the target probability distribution γ that is induced by a MDBD.

In addition, we show that a simple preprocessing step speeds up the run time of Spielman and Peng's [16] SDDM Solver from $\widetilde{O}(m_B\log^3 n\cdot\log^2\kappa_B)$ to $\widetilde{O}(m_B\log^2 n + n\log^4 n\cdot\log^5\kappa_B)$, where κ_B is the condition number of the input SDDM matrix B.

The heart of our algorithm is a simple and efficient sparsification method that outputs a sparsifier $D - \widehat{M}_p$ for any Laplacian or SDDM matrix polynomial $D - D\sum_{i=0}^{N}B_{N,i}(p)\cdot(D^{-1}M)^i$, whose coefficients follow Binomial distribution $B(N,p)$. Then, based on preprocessed sparsifiers, we can efficiently compute any positive linear combination of sparsifiers $D - \widehat{M}_{p_i}$, each of which corresponds to a different Binomial distribution $B(N,p_i)$, to obtain a sparsifier that is induced by a MDBD.

Our analysis of the representational power of MDBD relies on Hald's [10] result on mixture of Binomial distributions in the continuous setting, Cruz-Uribe and Neugebauer's [6, 7] sharp guarantees for approximating integrals using the Trapezoid method, and Doha et al.'s [8] closed formula for higher order derivatives of Bernstein basis.

When B is a dense Laplacian matrix, our result leads to the first nearly linear time algorithm for analyzing the long term behaviour of random walks induced by transition matrices of the form $\sum_{i=0}^{N}\gamma_i(D^{-1}M)^i$, where γ is induced by MDBD and the random walk length $N = \Theta(n)$. We hope that our sparsification algorithm for SDDM matrices can be useful in the context of Machine Learning.

1998 ACM Subject Classification G.2.2 [Discrete Mathematics]: Graph Theory—graph algorithms; G.3 [Probability and Statistics]: Probabilistic algorithms (including Monte Carlo).

Keywords and phrases spectral sparsification, mixture of Binomial distributions, Bernstein basis, graph algorithms, Markov chains, Laplacian polynomials, SDD and SDDM matrices.

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

* This work has been funded by the Cluster of Excellence "Multimodal Computing and Interaction" within the Excellence Initiative of the German Federal Government.

[1] We use the $\widetilde{O}(\cdot)$ notation to hide $(\log\log n)^3$ factors.

1 Introduction

In their seminal work Spielman and Teng [20] introduced the notion of spectral sparsifiers and proposed the first nearly linear time algorithm for spectral sparsification. In consecutive work, Spielman and Srivastava [19] proved that spectral sparsifiers with $O(\varepsilon^{-2}n\log n)$ edges exist and can be computed in $\widetilde{O}(m\log^c n\cdot\log(w_{max}/w_{min}))$ time for any undirected graph $G=(V,E,w)$. The computational bottleneck of their algorithm is to approximate the solutions of logarithmically many SDD[2] systems.

Recently, Koutis, Miller and Peng [13] developed an improved solver for SDD systems that works in $\widetilde{O}(m\log n\cdot\log(1/\varepsilon))$ time. In a survey result [11, Theorem 3] Kelner and Levin showed that in $\widetilde{O}(m\log^2 n)$ time all effective resistances can be approximated up to a constant factor. This yields a $(1\pm\varepsilon)$-spectral sparsifier with only a constant factor blow-up of non-zero edges $O(\varepsilon^{-2}n\log n)$. Although there are faster by a poly log-factor sparsification algorithms [12], they output spectral sparsifiers with poly log-factor more edges.

Spielman and Peng [16] introduced the notion of *sparse approximate inverse chain* of SDDM[3] matrices. They proposed the first parallel algorithm that finds such chains and runs in $\widetilde{O}(m\log^3 n\cdot\log^2\kappa)$ work and $O(\log^c n\cdot\log\kappa)$ depth, where κ is the condition number of the SDDM matrix with m non-zero entries and dimension n. Their main technical contributions shows that in $\widetilde{O}(\varepsilon^{-2}m\log^3 n)$ time a spectral sparsifier $\widetilde{D}-\widetilde{A}\approx_\varepsilon D-AD^{-1}A$ can be computed with $nnz(\widetilde{A})\leqslant O(\varepsilon^{-2}n\log n)$. In a follow up work, Cheng et al. [2] designed an algorithm that computes a normalized sparse approximate chain $\widetilde{C}$ such that $\widetilde{C}\widetilde{C}^{\mathrm{T}}\approx_\varepsilon M^p$ for any SDDM matrix M and $|p|\leqslant 1$. The construction of $\widetilde{C}$ involves an additional normalization step that outputs a sparsifier $D-\widetilde{M_{i+1}}\approx_\varepsilon D-\widetilde{M_i}D^{-1}\widetilde{M_i}$ that is expressed in terms of the original diagonal matrix D.

Most recently, Cheng et al. [3] initiated the study of random walk Laplacian matrix polynomials of the form $D-D\sum_{i=1}^N\xi_i(D^{-1}A)^i$, where ξ is a probability distribution over $[1:N]$, $D-A$ is a Laplacian matrix and $D^{-1}A$ is a random walk transition matrix. For the special case of even degree matrix monomials of the form $D-D(D^{-1}A)^N$, they proposed a sparsification algorithm that runs in nearly linear time. Although, their sparsification algorithm for Laplacian matrix polynomials works for any probability distribution ξ, the run time of their algorithm scales quadratically with the degree N and it takes $\widetilde{O}(\varepsilon^{-2}\cdot mN^2\cdot\log^{c_1}n\cdot\log^{c_2}N)$ time for some small constants c_1 and c_2.

2 Our Contribution

Our main contribution is to propose a unified spectral sparsification algorithm for both Laplacian and SDDM matrix polynomials that runs in $\widetilde{O}(\varepsilon^{-2}m_B\log^3 n\cdot\log^2 N+\varepsilon^{-4}\cdot nN\cdot\log^4 n\cdot\log^5 N)$ time and approximates large class of probability distributions. Before we formally state our results, we need some notations.

Since our analysis applies both for Laplacian and SDDM matrices and in addition all matrix operations that we perform preserve the matrix type, we adopt the following notation. Whenever we write B is $\mathcal{T}$-matrix we mean that it has a fixed type, i.e. it is either Laplacian or SDDM matrix. Moreover, to highlight that an operation preserves matrix type, we write that the input matrix B is $\mathcal{T}$-matrix and the output matrix B' is $\mathcal{T}$-matrix.

[2] SDD is the class of symmetric and diagonally dominant matrices.
[3] SDDM is the class of positive definite SDD matrices with non-positive off-diagonal entries.

We denote by $\mathcal{B}_{N,T}(\alpha, p) \triangleq \{(B(p_i, N), \alpha_i)\}_{i=1}^T$ a mixture of discrete Binomial distributions (MDBD) that satisfies the following two conditions:

1. (distinctness) $p_i \in (0,1)$ and $p_i \neq p_j$ for all $i \neq j \in [1:T]$;
2. (positive linear combination) $\sum_{i=1}^T \alpha_i \leqslant 1$ and $\alpha_i \in (0,1)$ for all $i \in [1:T]$.

Cheng et al. [3] studied Laplacian matrix polynomials of the form $D - D\sum_{i=1}^N \xi_i (D^{-1}A)^i$ for general probability distributions ξ. In contrast, we conduct a study that analyzes probability distributions that are induced by MDBD. We denote the Binomial probability mass function evaluated at $p \in (0,1)$ by $B_{N,i}(p) = \binom{N}{i} p^i (1-p)^{N-i}$. The ith coefficient of a MDBD $\mathcal{B}_{N,T}(\alpha, p)$ can be expressed as $\gamma_i = \sum_{j=1}^T \alpha_j \cdot B_{N,i}(p_j)$ for all $i \in [0:N]$. The Bernstein basis $B_{N,k}(p)$ is a well studied primitive in the literature for polynomial interpolation [7, 6]. We describe next the class of functions that are approximated by MDBD.

2.1 Main Theoretical Result

Our main theoretical contribution is to show, in Section 9, that the MDBD approximate large class of continuous probability distributions. Our approach is inspired by the following three influential studies. Hald [10] analyzed mixed Binomial distributions in the continuous case. Cruz-Uribe and Neugebauer [6, 7] established sharp guarantees for approximating integrals using the Trapezoid method. Doha et al. [8] gave a simple closed formula for higher order derivatives of Bernstein basis.

▶ Theorem 2.1. Let $N \in \mathbb{N}$ be a number, $\varepsilon > 0$ parameter, $I = [0,1]$ interval and $w(x)$ four times differentiable probability density function. Suppose there are reals $\mu \in (0,1)$ and $\kappa \geqslant 1$ such that:
1) $\max_{x\in I} |w''(x)| \leqslant 2\kappa \cdot N^2$, 2) $\max_{x\in I} |w'(x)| \leqslant \frac{1}{2}\kappa \cdot N$, 3) $\max_{x\in I} |w(x)| \leqslant \kappa$,
4) $\max_{x\in I} |b_1(x)| \leqslant \frac{1}{2}\mu \cdot N$, 5) $\max_{x\in I} |b_2(x)| \leqslant \frac{1}{2}\mu \cdot N^2$,
where the functions b_1, b_2 are defined by $b_1(x) = \frac{1}{w(x)}[-w(x) + (1-2x)w'(x) + \frac{1}{2}x(1-x)w''(x)]$ and $b_2(x) = \frac{1}{w(x)}[w(x) - 3(1-2x)w'(x) + (1-6x+6x^2)w''(x) + \frac{5}{6}x(1-x)(1-2x)w'''(x) + \frac{1}{8}x^2(1-x)^2 w^{\prime \mathrm{v}}(x)]$. Then for any $T \geqslant \Omega(N\sqrt{\kappa/\varepsilon})$ and all $i \in [3 : N-3]$ it holds that

$$\left|(1+\eta_i)\frac{w(i/N)}{N} - \frac{1}{T+1}\sum_{j=1}^T F_i\left(\frac{j}{T+1}\right)\right| \leqslant \frac{\varepsilon}{N}, \quad \text{where} \quad \eta_i \in [-\mu, \mu], \tag{1}$$

where the function $F_i(x) \triangleq w(x) \cdot B_{N,i}(x)$.

Using Theorem 2.1 and Theorem 2.3 we design an efficient algorithm (c.f. Appendix E) that constructs a spectral sparsifier of a matrix polynomial, the coefficients of which are induced by MDBD and approximate component-wise a desired *discretized* probability density function $w(i/N)/[\sum_{\ell=0}^N w(\ell/N)]$ for all $i \in [3 : N-3]$.

▶ Lemma 2.2. Suppose $w(x) = C \cdot f(x)$ is a probability density function such that
a) $1 \leqslant C \leqslant o(N)$, *b)* $0 \leqslant f(x) \leqslant 1$, *c)* $\frac{1}{2}[f(0) + f(1)] \geqslant \Omega(1)$, *d)* $\int_0^1 \left|f^{(2)}(x)\right| dx \leqslant o(N)$,
and in addition it satisfies the conditions in Theorem 2.1. Algorithm **Approx_DPDF** takes as input a $\mathcal{T}$-matrix $D - M$, function $w(x)$ as above, positive integer $N = 2^k$ for some $k \in \mathbb{N}$, and parameters $\varepsilon_I > 0$ and $\varepsilon_S > 0$. Then it outputs a spectral sparsifier $D - \widehat{M} \approx_{\varepsilon_S} D - D\sum_{i=0}^N \gamma_i (D^{-1}M)^i$ with probability vector γ such that for all $i \in [3 : N-3]$

$$\gamma_i \in \left[(1+2\eta_i)\frac{w\left(\frac{i}{N}\right)}{S} \pm \frac{2\varepsilon_I}{S}\right], \quad \text{where} \quad \eta_i \in [-\mu, \mu] \quad \text{and} \quad S = \sum_{\ell=0}^N w\left(\frac{\ell}{N}\right).$$

To illustrate our framework, we approximate in Appendix E two canonical *discretized* probability density functions - the Uniform distribution and the Exponential Families.

2.2 Main Algorithmic Results

Our main algorithmic contribution is to propose, in Section 7, a unified and efficient spectral sparsification algorithm for $\mathcal{T}$-matrix polynomials that are induced by MDBD.

▶ Theorem 2.3. Algorithm **SS_MDBD** takes as input $\mathcal{T}$-matrix $B = D - M$, numbers $N = 2^k \leqslant T$ for $k \in \mathbb{N}$, MDBD $\mathcal{B}_{N,T}(\alpha, p)$ with $\delta = 1 - \sum_{i=1}^{T} \alpha_i$ and parameter $\varepsilon \in (0,1)$. Then it outputs a spectral sparsifier $D - \widehat{M} \approx_\varepsilon D - D\sum_{i=0}^{N} \frac{1}{1-\delta}\gamma_i (D^{-1}M)^i$ that is $\mathcal{T}$-matrix with $O(\varepsilon^{-2} n \log n)$ non-zero entries, where $\gamma_i = \sum_{j=1}^{T} \alpha_j \cdot B_{N,i}(p_j)$. The algorithm's runtime is

$$\begin{cases} \widetilde{O}(m_B \log^2 n + \varepsilon^{-4} \cdot nT \cdot \log^4 n \cdot \log^5 N) & \text{, if } M \text{ is SPSD matrix;} \\ \widetilde{O}(\varepsilon^{-2} m_B \log^3 n \cdot \log^2 N + \varepsilon^{-4} \cdot nT \cdot \log^4 n \cdot \log^5 N) & \text{, otherwise.} \end{cases}$$

In Section 6, we develop Algorithm **LazySS** that efficiently sparsifies single Binomial $\mathcal{T}$-matrix polynomial of the form $D - D\sum_{i=0}^{N} B_{N,i}(p) \cdot (D^{-1}M)^i$ for any $p \in (0,1)$ and $N = 2^k$ where $k \in \mathbb{N}$. Theorem 2.3 builds upon Algorithm **LazySS** and speeds up the sparsification of T distinct Binomial $\mathcal{T}$-matrix polynomials and then sparsifies their positive linear combination (based on a vector α).

Theorem 2.1 shows how to choose the vectors α and p such that a MDBD $\mathcal{B}_{N,T}(\alpha, p)$ produces a vector γ that approximates a target probability distribution $w(x)$. Interestingly, in the case when the number of Binomial distributions is $T = N + 1$, we propose in Section 8 a nearly linear time algorithm that recovers exactly the vector α, given as input the vector p and the discrete probability distribution γ that is induced by a MDBD $\mathcal{B}_{N,T}(\alpha, p)$.

▶ Theorem 2.4. Let $p \in (0,1)^{N+1}$ be a vector that satisfies $0 < p_i \neq p_j < 1$ for every $i \neq j$. Suppose that $\gamma \in (0,1)^{N+1}$ is a vector generated by a mixture of discrete Binomial distributions $\mathcal{B}_{N,N+1}(\alpha, p)$ such that $\gamma_i = \sum_{j=1}^{N+1} \alpha_j \cdot B_{N,i}(p_j)$ for every $i \in [0:N]$. Then there is an algorithm that takes as input the vectors p and γ, and outputs in $O(N \log^2 N)$ time the vector $\alpha \in (0,1)^{N+1}$.

2.2.1 Application to Markov Chains

Using Theorem 2.3 we establish the following result on dense Laplacian matrices.

▶ Corollary 2.5. Suppose $B = D - M$ is dense Laplacian matrix with $m_B = \Omega(n^2)$, $\mathcal{B}_{N,T}(\alpha, p)$ is MDBD such that $\sum_{i=1}^{T} \alpha_i = 1$, the degree $N = 2^k$ for $k \in \mathbb{N}$ and the number of Binomials T are of order $\Theta(n)$, and $\varepsilon \in (0,1)$ is parameter. Then Algorithm **SS_MDBD** outputs in $\widetilde{O}(\varepsilon^{-4} m_B \log^9 n)$ time a spectral sparsifier $D - \widehat{M} \approx_\varepsilon D - D\sum_{i=0}^{N} \gamma_i (D^{-1}M)^i$ that is Laplacian matrix with $O(\varepsilon^{-2} n \log n)$ non-zero entries, where γ is a probability distribution induced by the MDBD $\mathcal{B}_{N,T}(\alpha, p)$.

Corollary 2.5 yields the first nearly linear time algorithm that analyzes Markov chains of length $\Theta(n)$ with corresponding transition matrices of the form $\sum_{i=0}^{\Theta(n)} \gamma_i (D^{-1}M)^i$, where γ is a probability distribution induced by MDBD. In comparison, the algorithm proposed by Cheng et al. [3, Theorem 2] runs in $\widetilde{O}(\varepsilon^{-2} n^4 \operatorname{poly}(\log n))$ time which makes it prohibitively expensive in practice. Thus, our result provides the first efficient and practical tool for analysing Markov chains of long length whose transition matrices are induced by MDBD.

2.3 Faster SDDM Solver

We show in Section 10 that the Spielman and Peng's [16] algorithm for finding a sparse approximate inverse chain of an SDDM matrix B can be speeded up by a simple preprocessing step. We improve the algorithm's run time from $\widetilde{O}(\varepsilon^{-2} m_B \log^3 n \cdot \log^2 \kappa_B)$ to $\widetilde{O}(m_B \log^2 n + \varepsilon^{-4} n \log^4 n \cdot \log^5 \kappa_B)$, which essentially shows that the algorithm's run time is upper bounded by the initial sparsification step.

▶ Theorem 2.6. There is an algorithm that takes as input SDDM matrix $B = D - M$ with condition number κ_B and parameter $\varepsilon \in (0, \frac{1}{2})$, then it outputs in time $\widetilde{O}(m_B \log^2 n + \varepsilon^{-4} n \log^4 n \cdot \log^5 \kappa_B)$ a matrix operator $\mathcal{M}$ with $nnz(\mathcal{M}) \leqslant O(\varepsilon^{-2} n \log n \cdot \log^3 \kappa_B)$ that spectrally approximates the inverse $\mathcal{M} \approx_\varepsilon B^{-1}$.

3 Background and Notations

We say that matrix X is spectral sparsifier of matrix Y if it satisfies $(1-\varepsilon)Y \preceq X \preceq (1+\varepsilon)Y$, for short $X \approx_\varepsilon Y$, where the partial relation $X \succeq 0$ stands for X is symmetric positive semi-definite (SPSD) matrix. We write $X \approx_{\varepsilon_1 \oplus \varepsilon_2} Y$ to indicate $(1-\varepsilon_1)(1-\varepsilon_2)Y \preceq X \preceq (1+\varepsilon_1)(1+\varepsilon_2)Y$. We use in our analysis the following five basic facts (c.f. [20, 1]).

▶ Fact 3.1. For positive semi-definite (PSD) matrices X, Y, W and Z it holds

a. if $Y \approx_\varepsilon Z$ then $X + Y \approx_\varepsilon X + Z$;
b. if $X \approx_\varepsilon Y$ and $W \approx_\varepsilon Z$ then $X + W \approx_\varepsilon Y + Z$;
c. if $X \approx_{\varepsilon_1} Y$ and $Y \approx_{\varepsilon_2} Z$ then $X \approx_{\varepsilon_1 \oplus \varepsilon_2} Z$;
d. if X and Y are PD matrices such that $X \approx_\varepsilon Y$ then $X^{-1} \approx_{2\varepsilon} Y^{-1}$, $\forall \varepsilon \in (0, \frac{1}{2})$;
e. for any matrix V if $X \approx_\varepsilon Y$ then $V^{\mathrm{T}} X V \approx_\varepsilon V^{\mathrm{T}} Y V$.

We denote by $nnz(A)$ or m_A the number of non-zero entries of matrix A. When we write "$B = D - M$ is $\mathcal{T}$-matrix" we assume that D is positive diagonal matrix and $B \in \mathbb{R}^{n \times n}$. We note that all algorithms presented in this paper output a spectral sparsifier *with high probability* and their run time is measured in *expectation*.

3.1 Spectral Sparsification Algorithms

We use the following algorithms. The first combines Kelner and Levin's [11, Theorem 3] and Cohen et al.'s [5, Lemma 4], whereas the second is proposed by Peng et al. [16, Corollary 6.4].

▶ Theorem 3.2. [11] Algorithm **ImpSS** takes as input parameter $\varepsilon \in (0,1)$, matrices D and A such that D is positive diagonal and A is symmetric non-negative with $A_{ii} = 0$ for all i such that $L = D - A$ is Laplacian matrix. Then in $\widetilde{O}(m_L \log^2 n)$ time outputs a positive diagonal matrix $\widetilde{D}$ and symmetric non-negative matrix $\widetilde{A}$ such that $nnz(\widetilde{A}) \leqslant O(\varepsilon^{-2} n \log n)$, $\widetilde{A}_{ii} = 0$ for all i, and $\widetilde{D} - \widetilde{A} \approx_\varepsilon D - A$. Moreover, $\widetilde{D} - \widetilde{A}$ is Laplacian matrix.

▶ Theorem 3.3. [16] Algorithm **PS14** that takes as input SDDM matrix $B = D - M$ and parameter $\varepsilon \in (0,1)$. Then in $O(\varepsilon^{-2} m_B \log^2 n)$ time outputs a positive diagonal matrix $\widetilde{D}$ and symmetric non-negative matrix $\widetilde{M}$ with $nnz(\widetilde{M}) \leqslant O(\varepsilon^{-2} m_B \log n)$ and $\widetilde{M}_{ii} = 0$ for all i, such that $\widetilde{D} - \widetilde{M} \approx_\varepsilon D - M D^{-1} M$ and $\widetilde{D} \approx_\varepsilon D$. Moreover, $\widetilde{D} - \widetilde{M}$ is SDDM matrix.

4 Spectral Sparsification of $\mathcal{T}$-Matrices

We begin by showing that the output sparsifier of Algorithm **ImpSS** can be amended such that it is expressed in terms of the original diagonal matrix D. Then we show that both Algorithms **ImpSS** and **PS14** can sparsify $\mathcal{T}$-matrices and the returned sparsifiers can be expressed in terms of the diagonal matrix D. Our analysis builds upon several results proposed by Peng et al. [17, 16, 2, 3].

▶ Lemma 4.1. There is an Algorithm **mImpSS** that takes as input a positive diagonal matrix D, symmetric non-negative matrix A (possibly $A_{ii} \neq 0$) such that $B = D - A$ is Laplacian matrix and parameter $\varepsilon \in (0,1)$. Then it outputs in $\widetilde{O}(m_B \log^2 n)$ time a spectral sparsifier $D - \widehat{A} \approx_\varepsilon D - A$ that is Laplacian matrix and satisfies $\widehat{A}$ is symmetric non-negative matrix with $nnz(\widehat{A}) \leqslant O(\varepsilon^{-2} n \log n)$.

The next result implicitly appears in [3]. For completeness we prove it in Appendex A.

▶ Lemma 4.2. Suppose $D - A$ is Laplacian matrix (possibly $A_{ii} \neq 0$) and $\widetilde{D} - \widetilde{A}$ a sparsifier with $\widetilde{A}_{ii} = 0$ for every i such that $(1-\varepsilon)(D-A) \preceq \widetilde{D} - \widetilde{A} \preceq (1+\varepsilon)(D-A)$. Then the symmetric non-negative matrix $\widehat{A} = (D - \frac{1}{1+\varepsilon}\widetilde{D}) + \frac{1}{1+\varepsilon}\widetilde{A}$ satisfies $(1-2\varepsilon)(D-A) \preceq D - \widehat{A} \preceq (1+2\varepsilon)(D-A)$.

We present now the proof of Lemma 4.1.

Proof of Lemma 4.1. Notice that $D - A = D' - A'$, where D' is positive diagonal matrix and A' is symmetric non-negative matrix such that $A'_{ii} = 0$ for all i. By Theorem 3.2 we obtain a sparsifier $\widetilde{D}' - \widetilde{A}' \approx_{\varepsilon/2} D' - A'$. Then by Lemma 4.2 we have $D' - \widehat{A}' \approx_\varepsilon D' - A'$, where $\widehat{A}' = (D' - \frac{1}{1+\varepsilon}\widetilde{D}') + \frac{1}{1+\varepsilon}\widetilde{A}'$ is symmetric non-negative matrix. We define by $D_A = D - D'$ a non-negative diagonal matrix. Set $\widehat{A} = D_A + \widehat{A}'$ and observe that it is symmetric and non-negative matrix. Now the statement follows since $D - \widehat{A} = D' - \widehat{A}'$. ∎

4.1 $\mathcal{T}$-Matrices

We show that if $D - M$ is $\mathcal{T}$-matrix then $D - D(D^{-1}M)^N$ is $\mathcal{T}$-matrix for any $N \in \mathbb{N}_+$. Then we present two algorithms that sparsifying matrices $D - D(D^{-1}M)^N$ for $N \in \{1,2\}$ such that the resulting sparsifiers are expressed in terms of the diagonal matrix D.

We give next a sparsification algorithm for $\mathcal{T}$-matrices of the form $D - M$.

▶ Lemma 4.3. There is an Algorithm **mKLP14** that takes as input $\mathcal{T}$-matrix $B = D - M$ and parameter $\varepsilon \in (0,1)$, then it outputs in $\widetilde{O}(m_B \log^2 n)$ time a spectral sparsifier $D - \widehat{M} \approx_\varepsilon D - M$ that is $\mathcal{T}$-matrix and satisfies $\widehat{M}$ is symmetric non-negative matrix with $nnz(\widehat{M}) \leqslant O(\varepsilon^{-2} n \log n)$.

Proof. By definition $B = D_1 + L$ where D_1 is non-negative diagonal matrix and $L = D_2 - M$ is Laplacian matrix. We obtain by Lemma 4.1 a sparsifier $D_2 - \widehat{M} \approx_\varepsilon D_2 - M$ that is Laplacian matrix. Now we consider two cases. If $D_1 = 0$ then we are done. Otherwise D_1 is PSD matrix and by Fact 3.1.a we have $D - \widehat{M} \approx_\varepsilon D - M$. Since $D - M$ is SDDM matrix and the operator $\approx_\varepsilon$ preserves the kernel space, it follows that $D - \widehat{M}$ is SDDM matrix. ∎

▶ Remark 4.4. Spielman and Peng [16, Proposition 5.6] showed that if $D - M$ is SDDM matrix, then $D - MD^{-1}M$ is SDDM matrix. We prove in Appendix A that $D - D(D^{-1}M)^N$ is SDDM matrix for every $N \in \mathbb{N}_+$. Also Cheng et al. [3, Proposition 25] showed that if $D - M$ is Laplacian matrix then $D - D(D^{-1}M)^N$ is Laplacian matrix for every $N \in \mathbb{N}_+$.

Based on Lemma 4.3 and Remark 4.4 we establish the following result.

▶ Corollary 4.5. Suppose $D - M$ is $\mathcal{T}$-matrix and $D - \widehat{M} \approx_\varepsilon D - M$ is a spectral sparsifier. Then $D - D(D^{-1}\widehat{M})^N$ is $\mathcal{T}$-matrix for every $N \in \mathbb{N}_+$.

We proceed by stating an interesting structural result that implicitly appears in [16] (c.f. Section "Efficient Parallel Construction"). For completeness we prove it in Appendix A.1.

▶ Lemma 4.6. Suppose $B = D - M$ is $\mathcal{T}$-matrix. Let $\eta_i = M_{i,:}^{\mathrm{T}} - M_{i,i} \cdot \mathbf{1}_i$ be a column vector, $d_i = \langle M_{i,:}, \mathbf{1} \rangle$ and $s_i = d_i - M_{ii}$ numbers, and $\mathbf{D}_{N_i} = (s_i/d_i) \cdot \mathrm{diag}(N_i)$ positive diagonal matrix for all i, where $N_i = \{M_{ij} \,|\, M_{ij} \neq 0\}$. Let $\mathbf{B}_{ij} = (M_{ii}/d_i + M_{jj}/d_j) \cdot M_{ij}$ be the (i,j)th entry of a matrix with same dimensions as matrix M and $\mathbf{D}_B = \mathrm{diag}(\mathbf{B} \cdot \mathbf{1})$ be a diagonal matrix.

Then it holds that $D - MD^{-1}M = \mathbf{D}_1 + \mathbf{L}_B + \sum_{i=1}^{n} \mathbf{L}_{N_i}$ where $\mathbf{D}_1 = \mathrm{diag}([D - MD^{-1}M]\mathbf{1})$ is non-negative diagonal matrix, $\mathbf{L}_B = \mathbf{D}_B - \mathbf{B}$ is Laplacian matrix with at most m_B non-zero entries and every $\mathbf{L}_{N_i} = (s_i/d_i)\mathbf{D}_{N_i} - \eta_i \eta_i^{\mathrm{T}}/d_i$ is Laplacian matrix corresponding to a clique with positively weighted edges that is induced by the neighbour set N_i.

Spielman and Peng [16] proposed Algorithm **PS14** (c.f. Theorem 3.3) for sparsifying matrices of the form $D - MD^{-1}M$, where $D - M$ is SDDM-matrix. We extend their result to $\mathcal{T}$-matrices and output a sparsifier that is expressed in terms of the diagonal matrix D.

▶ Lemma 4.7. There is an Algorithm **mPS14** that takes as input $\mathcal{T}$-matrix $B = D - M$ and parameter $\varepsilon \in (0,1)$. Then in $\widetilde{O}(\varepsilon^{-2} m_B \log^3 n)$ time outputs a spectral sparsifier $D - \widehat{M} \approx_\varepsilon D - MD^{-1}M$ that is $\mathcal{T}$-matrix and satisfies $\widehat{M}$ is symmetric non-negative matrix with $nnz(\widehat{M}) \leqslant O(\varepsilon^{-2} n \log n)$.

Proof. By Lemma 4.1 we have $D - MD^{-1}M = \mathbf{D}_1 + \mathbf{L}$, where $\mathbf{D}_1$ is non-negative diagonal matrix and $\mathbf{L}$ is sum of Laplacian matrices. Using similar arguments as in "Section 6 Efficient Parallel Construction" [16] we find a sparsifier $\widetilde{D} - \widetilde{M} \approx_{\varepsilon/2} \mathbf{L}$. Moreover, we can compute the positive diagonal matrix $D' = \mathrm{diag}(\mathbf{L})$ in $O(m_B)$ time (c.f. Appendix A.1), and then by Lemma 4.2 we obtain a sparsifier $D' - \widehat{M} \approx_\varepsilon \mathbf{L}$. Since $\mathbf{D}_1$ is PSD matrix the statement follows by Fact 3.1.a. ∎

5 Core Iterative Algorithm

Cheng et al. [3] proposed an efficient sparsification algorithm for Laplacian matrices of the form $D - D(D^{-1}A)^N$, where $D - A$ is Laplacian matrix and $N \in \mathbb{N}$ is even. We establish in this section a generalized sparsification algorithm that handles $\mathcal{T}$-matrices. Moreover, we show that in the case when M is SPSD matrix our algorithm has faster runtime by $(\varepsilon^{-2} \log n)$-factor. Interestingly, in this case we remove the ε^{-2} dependence on the non-zero entries m_B that appears in [3].

▶ Theorem 5.1. There is an Algorithm **PwrSS** that takes as input $\mathcal{T}$-matrix $B = D - M$, $N = 2^k$ for $k \in \mathbb{N}$ and $\varepsilon \in (0,1)$, then it outputs a spectral sparsifier $D - \widehat{M_N} \approx_\varepsilon D - D(D^{-1}M)^N$ that is $\mathcal{T}$-matrix with $nnz(\widehat{M_N}) \leqslant O(\varepsilon^{-2} n \log n)$. The Algorithm **PwrSS** runs in time

$$\begin{cases} \widetilde{O}(m_B \log^2 n + \varepsilon^{-4} \cdot n \log^4 n \cdot \log^5 N) & \text{, if } M \text{ is SPSD matrix;} \\ \widetilde{O}(\varepsilon^{-2} m_B \log^3 n \cdot \log^2 N + \varepsilon^{-4} \cdot n \log^4 n \cdot \log^5 N) & \text{, otherwise.} \end{cases}$$

The rest of this section is devoted to proving Theorem 5.1. Our proof follows the approach proposed in [4]. We begin by extending [4, Lemma 4.3 and 4.4]. For completeness, we provide a prove in Appendix F where in addition we generalize [4, Fact 4.2].

▶ Lemma 5.2. Suppose $B = D - M$ is $\mathcal{T}$-matrix and $D - \widehat{M} \approx_\varepsilon D - M$ is spectral sparsifier. If M is SPSD matrix then it holds that $D - \widehat{M}D^{-1}\widehat{M} \approx_\varepsilon D - MD^{-1}M$.

Based on Lemma 5.2 we propose a faster sparsification algorithm for $\mathcal{T}$-matrices of the form $D - MD^{-1}M$, in the case when M is SPSD matrix. Our improvement is due to the fact that it suffices to sparsify matrix $D - \widehat{M}D^{-1}\widehat{M}$ instead of matrix $D - MD^{-1}M$.

▶ **Theorem 5.3.** There is an Algorithm **InitSS** that takes as input $\mathcal{T}$-matrix $B = D - M$ such that M is SPSD matrix, and parameter $\varepsilon \in (0,1)$. Then it outputs in $\widetilde{O}(m_B \log^2 n + \varepsilon^{-4} n \log^4 n)$ time a spectral sparsifier $D - \widehat{M_2} \approx_\varepsilon D - MD^{-1}M$ that is $\mathcal{T}$-matrix with $nnz(\widehat{M_2}) \leqslant O(\varepsilon^{-2} n \log n)$.

Proof of Theorem 5.3. We apply Lemma 4.3 to obtain a sparsifier $D - \widehat{M} \approx_{\varepsilon/4} D - M$ in $\widetilde{O}(m_B \log^2 n)$ time with $nnz(\widehat{M}) \leqslant O(\varepsilon^{-2} n \log n)$ such that $D - \widehat{M}$ is $\mathcal{T}$-matrix. Then by Lemma 5.2 we know that $D - \widehat{M} D^{-1} \widehat{M} \approx_{\varepsilon/4} D - MD^{-1}M$. Now, by Corollary 4.5 $D - \widehat{M} D^{-1} \widehat{M}$ is $\mathcal{T}$-matrix. Then we apply Lemma 4.7 to obtain in $\widetilde{O}(\varepsilon^{-4} n \log^4 n)$ time a sparsifier $D - \widehat{M_2} \approx_{\varepsilon/4} D - \widehat{M} D^{-1} \widehat{M}$ with $nnz(\widehat{M}) \leqslant O(\varepsilon^{-2} n \log n)$ such that $D - \widehat{M_2}$ is $\mathcal{T}$-matrix. The claims follows by Fact 3.1.c. ∎

To prove Theorem 5.1 we need to bound the approximation error incurred after $O(\log N)$ consecutive square sparsification operations. We follow the approach proposed in [4, Lemma 4.1], but in addition we exhibit an interesting algebraic structure that all matrices of the form $D(D^{-1}M)^{2^k}$ have in common. The speed up in our algorithm is based on this observation.

▶ **Lemma 5.4.** If M is symmetric matrix, then $D(D^{-1}M)^{2^k}$ is SPSD matrix for every $k \in \mathbb{N}_+$.

Proof. Let $Y \triangleq D^{-1/2} M D^{-1/2}$. Notice that $D(D^{-1}M)^{2^k} = D^{1/2} Y^{2^k} D^{1/2} = X^{\mathrm{T}} X$, where $X = Y^{2^{k-1}} D^{1/2}$. The statement follows since $X^{\mathrm{T}} X$ is SPSD matrix. ∎

We argue in a similar manner as in [4, Lemma 4.1] to analyze the incurred approximation error after $O(\log N)$ square sparsifications. For completeness we proof it in Appendix B.

▶ **Lemma 5.5.** Let $D - M$ and $D - \widehat{M_{2^k}}$ are $\mathcal{T}$-matrices such that $D - \widehat{M_{2^k}} \approx_{\varepsilon'} D - D(D^{-1}M)^{2^k}$. There is an Algorithm **SqrSS** that takes as input the $\mathcal{T}$-matrix $D - \widehat{M_{2^k}}$ and parameter $\varepsilon \in (0,1)$, then it outputs in $\widetilde{O}(\varepsilon^{-2} nnz(\widehat{M_{2^k}}) \log^3 n)$ time a symmetric non-negative matrix $\widehat{M_{2^{k+1}}}$ with $nnz(\widehat{M_{2^{k+1}}}) \leqslant O(\varepsilon^{-2} n \log n)$ such that $D - \widehat{M_{2^{k+1}}} \approx_\varepsilon D - \widehat{M_{2^k}} D^{-1} \widehat{M_{2^k}} \approx_{\varepsilon'} D - D(D^{-1}M)^{2^{k+1}}$ is $\mathcal{T}$-matrix.

Lemma 5.5 gives a simple iterative method (Algorithm **PwrSS**) for sparsifying $\mathcal{T}$-matrices of the form $D - D(D^{-1}M)^N$. We are now ready to prove Theorem 5.1.

Proof of Theorem 5.1. Let $\varepsilon' = \varepsilon / 2 \log N$. We apply Theorem 5.3 if M is SPSD matrix otherwise we use Lemma 4.7 to find a sparsifier $D - \widehat{M_2} \approx_{\varepsilon'} D - MD^{-1}M$ with $nnz(\widehat{M_2}) \leqslant O(\varepsilon^{-2} n \log n \cdot \log^2 N)$. Then we call $O(\log N)$ times Lemma 5.5. Each call finds a sparsifier $D - \widehat{M_{2^k}} \approx_{\oplus^k \varepsilon'} D - D(D^{-1}M)^{2^k}$ with $nnz(\widehat{M_{2^k}}) \leqslant O(\varepsilon^{-2} n \log n \cdot \log^2 N)$ in time $\widetilde{O}(\varepsilon^{-4} n \log^4 n \cdot \log^4 N)$. The approximation error follows by Fact 3.1.c and the definition of ε'. Then we perform a final refinement sparsification. ∎

6 Spectral Sparsification of Binomial $\mathcal{T}$-Matrix Polynomials

We consider the matrices $W_p = (1-p)I + pD^{-1}M$ and the polynomials $f_p(x) = (1-p) + px$ for any $p \in (0,1)$. The coefficients of the polynomial $[f_p(x)]^N$ follow Binomial distribution $B(N,p)$. Moreover, for every $p \in (0,1)$ we have that the Nth power of matrix W_p satisfies $W_p^N = \sum_{i=0}^{N} B_{N,i}(p) \cdot (D^{-1}M)^i$, where $B_{N,i}(p) = \binom{N}{i} p^i (1-p)^{N-i}$. We note that in the case when $D - M$ is Laplacian matrix, the matrix W_p^N corresponds to the transition matrix of a p-lazy random walk process of length N (c.f. [18]).

Our main contribution in this section is to propose an efficient sparsification algorithm for a single Binomial $\mathcal{T}$-matrix polynomials of the form $D - D\sum_{i=0}^{N} B_{N,i}(p) \cdot (D^{-1}M)^i$. We give a simple preprocessing step that allows Algorithm **PwrSS** to efficiently sparsify matrices $D - DW_p^N$ for any $p \in (0,1)$. Our approach improves over [3, Theorem 2] by $\Theta(N^2)$-factor.

▶ Theorem 6.1. There is an Algorithm **LazySS** that takes as input $\mathcal{T}$-matrix $B = D - M$, number $N = 2^k$ for $k \in \mathbb{N}$, and parameters $\varepsilon, p \in (0, 1)$. Then it outputs a spectral sparsifier

$$D - \widehat{M_{p,N}} \approx_\varepsilon D - DW_p^N = D - D \sum_{i=0}^{N} B_{N,i}(p) \cdot (D^{-1}M)^i$$

that is $\mathcal{T}$-matrix with at most $O(\varepsilon^{-2} n \log n)$ non-zero entries. The Algorithm **LazySS** runs in time

$$\begin{cases} \widetilde{O}(m_B \log^2 n + \varepsilon^{-4} \cdot n \log^4 n \cdot \log^5 N) & \text{, if } p \in (0, 1/2]; \\ \widetilde{O}(\varepsilon^{-2} m_B \log^3 n \cdot \log^2 N + \varepsilon^{-4} \cdot n \log^4 n \cdot \log^5 N) & \text{, otherwise.} \end{cases}$$

We analyze next the algebraic structure of matrices $DW_p^{2^k}$ and $D - DW_p^N$. Moreover, we prove for them similar results as in Lemma 5.4 and Remark 4.4.

▶ Lemma 6.2. Suppose $D - M$ is $\mathcal{T}$-matrix. Then $D - DW_p^N$ is $\mathcal{T}$-matrix for every $N \in \mathbb{N}_+$. Also DW_p is SPSD matrix $\forall p \in (0, 1/2]$ and $DW_p^{2^k}$ is SPSD matrix $\forall p \in (0, 1)$ and $\forall k \in \mathbb{N}_+$.

Proof. By definition of W_p, we have $D - DW_p = p(D - M)$ is $\mathcal{T}$-matrix. Suppose $D - M$ is Laplacian matrix, then $D - DW_p$ is Laplacian matrix and by Remark 4.4 $D - DW_p^N$ is Laplacian matrix for every $N \in \mathbb{N}_+$. Suppose now that $D - M$ is SDDM matrix, then $D - DW_p$ is SDDM matrix and by Lemma A.2 $D - DW_p^N$ is SDDM matrix for every $N \in \mathbb{N}_+$.

By definition $DW_p = (1 - p)D + pM$ and since $D - M$ is diagonally dominant, it holds that DW_p is SPSD matrix for every $p \in (0, 1/2]$. Moreover, since DW_p is symmetric matrix by Lemma 5.4 it holds that $D(D^{-1} \cdot DW_p)^{2^k} = DW_p^{2^k}$ is SPSD matrix for every $k \in \mathbb{N}_+$. ◼

Proof of Theorem 6.1. The statement follows by Lemma 6.2 and Theorem 5.1. ◼

7 Spectral Sparsification of $\mathcal{T}$-Matrix Polynomials Induced by MDBD

In this section we prove Theorem 2.3. For completeness we present in Appendix C the pseudo code of Algorithm **SS_MDBD**. Our result builds upon the following key algorithmic idea.

▶ Lemma 7.1. Suppose $D - M$, $D - \widehat{M_1} \approx_\varepsilon D - M$ and $D - \widehat{M_2} \approx_\varepsilon D - MD^{-1}M$ are $\mathcal{T}$-matrices. Then the spectral sparsifier $D - \widehat{M_{p,2}} \approx_\varepsilon D - DW_p^2$ is $\mathcal{T}$-matrix for every $p \in (0, 1)$, where the sparse matrix $\widehat{M_{p,2}} = (1 - p)^2 D + 2p(1 - p)\widehat{M_1} + p^2 \widehat{M_2}$.

Proof. Notice that $D - DW_p^2 = 2p(1 - p)[D - M] + p^2[D - MD^{-1}M] \approx_\varepsilon D - \widehat{M_{p,2}}$. ◼

Let $\delta = 1 - \sum_{i=1}^{T} \alpha_i$. We denote a $\mathcal{T}$-matrix polynomial induced by MDBD $\mathcal{B}_{N,T}(\alpha, p)$ as $\mathrm{P}_{\mathcal{B}} \triangleq \sum_{j=1}^{T} \alpha_j (D - DW_{p_j}^N) = (1 - \delta)D - D \sum_{i=0}^{N} \gamma_i (D^{-1}M)^i$, where $\gamma_i = \sum_{j=1}^{T} \alpha_j B_{N,i}(p_j)$. We prove now Theorem 2.3.

Proof of Theorem 2.3. Let $\varepsilon' = \varepsilon / 2 \log N$. We perform first a preprocessing step. We apply Lemma 4.3 to obtain a sparsifier $D - \widehat{M} \approx_{\varepsilon'} D - M$. Then depending on whether M is SPSD matrix we use either Lemma 4.7 or Theorem 5.3 to obtain a sparsifier $D - \widehat{M_2} \approx_{\varepsilon'} D - MD^{-1}M$. The run time is at most $\widetilde{O}(\varepsilon^{-2} m_B \log^3 n \cdot \log^2 N)$ or $\widetilde{O}(m_B \log^2 n + \varepsilon^{-4} n \log^4 n)$ respectively. Moreover, the sparsifiers satisfy $nnz(\widehat{M_1}), nnz(\widehat{M_2}) \leqslant O(\varepsilon^{-2} n \log n \cdot \log^2 N)$.

We combine Lemma 7.1 and Theorem 5.1 to find each sparsifier $D - \widehat{M_{p_j,N}} \approx_\varepsilon D - DW_{p_j}^N$ by initializing Algorithm **PwrSS** with a sparsifier $D - \widehat{M_{p_j,2}} \approx_{\varepsilon'} D - DW_{p_j}^2$. Let $\widehat{M_{tmp}} =$

$\sum_{j=1}^{T} \alpha_j \widehat{M_{p_j,N}}$, then by Fact 3.1.b it holds $(1-\delta)D-\widehat{M_{tmp}} \approx_{\varepsilon} \mathrm{P}_{\mathcal{B}}$. The run time of this phase is $\widetilde{O}(\varepsilon^{-4}nT\log^4 n \cdot \log^5 N)$ and each sparsifier satisfies $nnz(\widehat{M_{p_i,N}}) \leqslant O(\varepsilon^{-2}n\log n \cdot \log^2 N)$.

However, matrix $\widehat{M_{tmp}}$ can be dense. We apply Lemma 4.3 to obtain a sparsifier $D-\widehat{M} \approx_{\varepsilon} D - \frac{1}{1-\delta}\widehat{M_{tmp}} \approx_{\varepsilon} \frac{1}{1-\delta}\mathrm{P}_{\mathcal{B}}$ in $\widetilde{O}(\min\{n^2\log^2 n, \varepsilon^{-2}nT\log^3 n \cdot \log^2 N\})$ time such that $nnz(\widehat{M}) \leqslant O(\varepsilon^{-2}n\log n)$. ■

8 Solving Transpose Bernstein-Vandermonde Linear Systems

▶ Problem 1. Suppose a vector $\gamma \in (0,1)^{N+1}$ is generated by a convex combination of exactly $N+1$ discrete Binomial distributions $B(p_i, N)$ such that $0 < p_i \neq p_j < 1$ for all $i \neq j$. Find the vector $\alpha \in (0,1)^{N+1}$ that satisfies $\gamma_i = \sum_{j=1}^{N+1} \alpha_j \cdot B_{N,i}(p_j)$ for all $i \in [0:N]$.

The Bernstein basis is a well studied primitive in the literature for polynomial interpolations [6, 7]. It is defined by $B_{N,k}(p) = \binom{N}{k}p^k(1-p)^{N-k}$ for any $k \in [0:N]$. Suppose $\mathcal{B}_{N,T}(\alpha,p)$ is MDBD with $T = N+1$. We denote a Bernstein basis matrix by $[\mathbf{B}_N(p)]_{ji} = B_{N,i}(p_j), \quad \forall i,j \in [1:N+1]$. Moreover, we note that the vector α is a solution of the linear system $\mathbf{B}_N(p)^{\mathrm{T}}\alpha = \gamma$.

8.1 Fast Algorithm for Solving Transpose Bernstein-Vandermonde Systems

We present a nearly linear time algorithm that solves Problem 1. For completeness, we show in Appendix D that the Bernstein basis matrix $\mathbf{B}_N(p)$ has full rank. Our approach consist of reducing a transpose Bernstein-Vandermonde system to a transpose Vandermonde system, which can be solved efficiently using an algorithm proposed by Gohberg and Olshevsky [9].

▶ Theorem 8.1. [9] Suppose $\alpha \in \mathbb{R}^{N+1}$ is a vector. There is an algorithm that takes as input a vector p such that $0 < p_i \neq p_j < 1$ for all $i \neq j$, transpose Vandermonde matrix $\mathbf{V}(p)^{\mathrm{T}}$ and vector $\gamma = \mathbf{V}(p)^{\mathrm{T}}\alpha$. Then the algorithm recovers exactly the vector $\alpha = [\mathbf{V}(p)^{\mathrm{T}}]^{-1}\gamma$ in $O(N\log^2 N)$ time.

We show now that the Bernstein basis matrix $\mathbf{B}_N(p)$ can be decomposed as follows.

▶ Lemma 8.2. Suppose $p \in (0,1)^{N+1}$ is vector such that $0 < p_i \neq p_j < 1$ for all $i \neq j$, $\mathbf{V}(p)$ is Vandermonde matrix defined by $[\mathbf{V}(p)]_{ji} = (\frac{p_j}{1-p_j})^i$, $D_p = \mathrm{diag}(\{(1-p_j)^N\}_{j=1}^{N+1})$ and $D_{CN} = \mathrm{diag}(\{\binom{N}{i}\}_{i=0}^{N})$ are positive diagonal matrices. Then it holds that $\mathbf{B}_N(p) = D_p \cdot \mathbf{V}(p) \cdot D_{CN}$.

Proof. By definition $[D_p \cdot \mathbf{V}(p) \cdot D_{CN}]_{j,i} = (1-p_j)^N(\frac{p_j}{1-p_j})^i\binom{N}{i} = B_{N,i}(p_j) = [\mathbf{B}_N(p)]_{ji}$. ■

We are ready now to prove Theorem 2.4.

Proof of Theorem 2.4. By Lemma D.1 the Bernstein matrix $\mathbf{B}_N(p)$ is invertible. Given a vector $\gamma \in (0,1)^{N+1}$ we want to find the vector $\alpha = [\mathbf{B}_N(p)^{\mathrm{T}}]^{-1}\gamma$. By Lemma 8.2 we have $[\mathbf{B}_N(p)^{\mathrm{T}}]^{-1} = [D_p]^{-1} \cdot [\mathbf{V}(p)^{\mathrm{T}}]^{-1} \cdot [D_{CN}]^{-1}$. Moreover, we can compute a vector $\gamma' = [D_{CN}]^{-1}\gamma$ in $O(n)$ time. Using Theorem 8.1, we obtain a vector $\gamma'' = [\mathbf{V}(p)^{\mathrm{T}}]^{-1}\gamma'$ in $O(N\log^2 N)$ time. The desired vector $\alpha = [D_p]^{-1}\gamma''$ takes further $O(n)$ time to compute. ■

9 Representational Power of MDBD

In this section we prove Theorem 2.1 and demonstrate that the MDBD approximate large class of probability distributions. We also show in Appendix E that the Uniform distribution and Exponential families can be approximated by Theorem 2.1. Our approach of proving Theorem 2.1 uses the following result proposed by Hald [10] on mixture of Binomial distributions.

▶ Theorem 9.1. [10] Let $w(x)$ be a probability density function that is four times differentiable. Then for every $N \in \mathbb{N}$ and $i \in [0, N]$ the Bernstein basis $B_{N,i}(p)$ satisfies

$$\int_0^1 w(p) \cdot B_{N,i}(p)dp = \frac{w(i/N)}{N} \cdot \left[1 + \frac{b_1(i/N)}{N} + \frac{b_2(i/N)}{N^2} + O\left(\frac{1}{N^3}\right)\right] \tag{2}$$

where the functions are defined by $b_1(x) = \frac{1}{w(x)}[-w(x) + (1-2x)w'(x) + \frac{1}{2}x(1-x)w''(x)]$ and $b_2(x) = \frac{1}{w(x)}[w(x) - 3(1-2x)w'(x) + (1-6x+6x^2)w''(x) + \frac{5}{6}x(1-x)(1-2x)w'''(x) + \frac{1}{8}x^2(1-x)^2 w'^v(x)]$.

We distinguish two types of approximation errors. The error term $(1+\eta_i)$ (c.f. Equation 1 in Theorem 2.1) is caused by the error introduced in Equation 2. The second error type is due to the integral discretization with finite summation. The later approximation error in analyzed by Cruz-Uribe and Neugebauer [6, 7]. We summarize next their result on the Trapezoid method.

▶ Theorem 9.2. [7, 6] Suppose f be continuous and twice differentiable function, $T \in \mathbb{N}$ is number, and the discrete approximator of f is defined by $A_T(f) = [\frac{1}{2}(f(0) + f(1)) + \sum_{i=1}^{T-1} f(i/T)]/T$. Then the approximation error is given by the expression $E_T(f) = |A_T(f) - \int_0^1 f(t)dt| = |\sum_{i=1}^{T} L_i|$, where $L_i = \frac{1}{2}\int_{x_{i-1}}^{x_i} \left[\frac{1}{4T^2} - (t-c_i)^2\right] f''(t)dt$ and $c_i = (x_{i-1} + x_i)/2$.

The rest of this section is devoted to proving Theorem 2.1. We use the following two results established by Cruz-Uribe and Neugebauer, and Doha et al.

▶ Lemma 9.3. [7, 6] The Bernstein basis satisfies $\int_0^1 B_{N,i}(x)dx = \frac{1}{N+1}$ for every $i \in [0 : N]$.

▶ Lemma 9.4. [8] The pth derivative of a Bernstein basis satisfies for every $i \in [0 : N]$ that

$$\frac{d^p B_{N,i}(x)}{dx^p} = \frac{N!}{(N-p)!} \sum_{k=\max\{0,i+p-N\}}^{\min\{i,p\}} (-1)^{k+p} \cdot \binom{p}{k} \cdot B_{N-p,i-k}(x).$$

We propose an upper bound on the integral of pth order derivative of Bernstein basis.

▶ Corollary 9.5. For every $p \in [1 : N-1]$ and $i \in [p+1 : N-(p+1)]$ such that $i + p \leqslant N$, the Bernstein basis satisfies

$$\int_0^1 \left|\frac{d^p B_{N,i}(x)}{dx^p}(t)\right| dt \leqslant \frac{N!}{(N-p)!} \cdot \frac{2^p}{N+1}.$$

Proof. We combine Lemma 9.3 and Lemma 9.4 to obtain

$$\int_0^1 \left|\frac{d^p B_{N,i}(x)}{dx^p}(t)\right| dt \leqslant \frac{N!}{(N-p)!} \sum_{k=0}^{p} \binom{p}{k} \cdot \int_0^1 B_{N-p,i-k}(t)dt = \frac{N!}{(N-p)!} \cdot \frac{2^p}{N+1}.$$

∎

We are now ready to prove Theorem 2.1.

Proof of Theorem 2.1. Recall that $F_i(x) = w(x)B_{N,i}(x)$. By Theorem 9.2 we have

$$\left|\sum_{k=1}^{T} L_k\right| = \left|\frac{1}{2}\sum_{k=1}^{T}\int_{x_{k-1}}^{x_k} \left[\frac{1}{4T^2} - (t-c_k)^2\right] \cdot \frac{d^2F_i(x)}{dx^2}(t)dt\right| \leqslant \frac{1}{8T^2}\int_0^1 \left|\frac{d^2F_i(x)}{dx^2}(t)\right| dt.$$

Since $|\frac{d^2F_i(x)}{dx^2}| = w'' \cdot B_{N,i} + 2 \cdot w' \cdot B'_{N,i} + w \cdot B''_{N,i}$, we consider following three cases:
Case 1: We combine $\max_{x\in[0,1]} |w''(x)| \leqslant 2\kappa \cdot N^2$ and Lemma 9.3 to obtain

$$\int_0^1 |w''(t) \cdot B_{N,i}(t)|dt \leqslant 2\kappa \cdot N.$$

Case 2: Using $\max_{x\in[0,1]} |w'(x)| \leqslant \frac{1}{2}\kappa \cdot N$ and Corollary 9.5 it holds

$$\int_0^1 |w'(t) \cdot B'_{N,i}(t)|dt \leqslant \kappa \cdot N.$$

Case 3: Combining $\max_{x\in[0,1]} |w(x)| \leqslant \kappa$ and Corollary 9.5 yields

$$\int_0^1 |w(t) \cdot B''_{N,i}(t)|dt \leqslant \kappa \cdot \int_0^1 B''_{N,i}(t)dt \leqslant 4\kappa \cdot N.$$

The desired result follows from the preceding three cases and Theorem 9.1. ∎

10 Faster SDDM Solver

We give a simple preprocessing step that speeds up the run time of Spielman and Peng's [16] SDDM Solver from $\widetilde{O}(\varepsilon^{-2} m_B \log^3 n \cdot \log^2 \kappa_B)$ to $\widetilde{O}(m_B \log^2 n + \varepsilon^{-4} n \log^4 n \cdot \log^5 \kappa_B)$. We show that it suffices to find a sparse approximate inverse chain of the sparsifier $D - \widehat{M} \approx_{\varepsilon/8} D - M$. This allows us to sparsify matrix $D - \widehat{M}D^{-1}\widehat{M}$ instead of $D - MD^{-1}M$, which yields the improvement.

Proof of Theorem 2.6. We argue similarly as in [16], except that we use a preprocessing step. We apply Lemma 4.3 to obtain in $\widetilde{O}(m_B \log^2 n)$ time a sparsifier $D - \widehat{M} \approx_{\varepsilon/8} D - M = B$ with $nnz(\widehat{M}) \leqslant O(\varepsilon^{-2} n \log n)$. Then by Fact 3.1.d we have $(D - \widehat{M})^{-1} \approx_{\varepsilon/4} (D - M)^{-1}$.

We construct now a sparse approximate inverse chain of matrix $\widehat{B} = D - \widehat{M}$. We note that its condition number satisfies $\kappa_{\widehat{B}} \leqslant \frac{1+\varepsilon/8}{1-\varepsilon/8}\kappa_B = t_{\widehat{B}}$. We denote by $\varepsilon' = \varepsilon/(16 \log t_{\widehat{B}})$. Spielman and Peng [16] showed that $O(\log t_{\widehat{B}})$ iterations suffice for the following iterative procedure to output such chain with $\varepsilon/4$-factor approximation.

By Lemma 4.7 we find in $\widetilde{O}(\varepsilon^{-4} n \log^4 n \cdot \log^2 \kappa_B)$ time a sparsifier $D - \widehat{M_2} \approx_{\varepsilon'/2} D - \widehat{M}D^{-1}\widehat{M}$ with $nnz(\widehat{M_2}) \leqslant O(\varepsilon^{-2} n \log n \cdot \log^2 \kappa_B)$. Each consecutive call outputs a sparsifier with the same number of non-zero entries in time $\widetilde{O}(\varepsilon^{-4} n \log^4 n \cdot \log^4 \kappa_B)$, and there are $O(\log t_{\widehat{B}})$ such calls. By Corollary 4.5 both $D - \widehat{M}$ and $D - \widehat{M}D^{-1}\widehat{M}$ are SDDM matrices and thus invertible. We combine now Fact 3.1 and apply recursively $O(\log t_{\widehat{B}})$ times the relation

$$\begin{aligned}
& [D^{-1} + (I + D^{-1}\widehat{M})[D - \widehat{M_2}]^{-1}(I + \widehat{M}D^{-1})]/2 \\
\approx_{\varepsilon'} \quad & [D^{-1} + (I + D^{-1}\widehat{M})[D - \widehat{M}D^{-1}\widehat{M}]^{-1}(I + \widehat{M}D^{-1})]/2 \\
= \quad & (D - \widehat{M})^{-1}.
\end{aligned}$$

The final step of the recurrence is analyzed by combining Corollary 5.5 in [16] and Lemma 5.5 to obtain $D \approx_{\varepsilon/16} D - \widehat{M}_{\log t_{\widehat{B}}} \approx_{\varepsilon/16} D - D(D^{-1}\widehat{M})^{t_{\widehat{B}}}$. Hence, the statement follows by Fact 3.1. ∎

Acknowledgements We are grateful to Arijit Ghosh and Kunal Dutta for the helpful discussions on mixture of Binomial distributions, and to Arnur Nigmetov and Shay Moran for pointing us to the Bernstein interpolation polynomials. We would also like to thank Kurt Mehlhorn for the insightful comments and suggestions to the early version of the manuscript.

This work has been funded by the Cluster of Excellence "Multimodal Computing and Interaction" within the Excellence Initiative of the German Federal Government.

References

1 Marshall W. Bern, John R. Gilbert, Bruce Hendrickson, Nhat Nguyen, and Sivan Toledo. Support-graph preconditioners. *SIAM Journal on Matrix Analysis and Applications*, 27(4):930–951, 2006.

2 Dehua Cheng, Yu Cheng, Yan Liu, Richard Peng, and Shang-Hua Teng. Scalable parallel factorizations of SDD matrices and efficient sampling for gaussian graphical models. *CoRR*, abs/1410.5392, 2014.

3 Dehua Cheng, Yu Cheng, Yan Liu, Richard Peng, and Shang-Hua Teng. Efficient sampling for gaussian graphical models via spectral sparsification. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 364–390, 2015.

4 Dehua Cheng, Yu Cheng, Yan Liu, Richard Peng, and Shang-Hua Teng. Spectral sparsification of random-walk matrix polynomials. *CoRR*, abs/1502.03496, 2015.

5 Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 181–190, 2015.

6 D. Cruz-Uribe and C. J. Neugebauer. An elementary proof of error estimates for the trapezoidal rule. *Mathematics Magazine*, 76(4):pp. 303–306, 2003.

7 D. Cruz-Uribe and C.J. Neugebauer. Sharp error bounds for the trapezoidal rule and Simpson's rule. *JIPAM. J. Inequal. Pure Appl. Math.*, 3(49), 2002.

8 E.H. Doha, A.H. Bhrawy, and M.A. Saker. On the derivatives of Bernstein polynomials: an application for the solution of high even-order differential equations. *Bound. Value Probl.*, 2011:16, 2011.

9 I. Gohberg and V. Olshevsky. Fast algorithms with preprocessing for matrix-vector multiplication problems. *Journal of Complexity*, 10(4):411 – 427, 1994.

10 A. Hald. The mixed binomial distributin and the posterior distribution of p for a continuous prior distribution. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30(2):pp. 359–367, 1968.

11 Jonathan A. Kelner and Alex Levin. Spectral sparsification in the semi-streaming setting. *Theory Comput. Syst.*, 53(2):243–262, 2013.

12 Ioannis Koutis, Alex Levin, and Richard Peng. Improved spectral sparsification and numerical algorithms for SDD matrices. In *29th International Symposium on Theoretical Aspects of Computer Science, STACS 2012, February 29th - March 3rd, 2012, Paris, France*, pages 266–277, 2012.

13 Ioannis Koutis, Gary L. Miller, and Richard Peng. A nearly-m log n time solver for sdd linear systems. In *Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 590–598, Washington, DC, USA, 2011. IEEE Computer Society.

14 Gary L. Miller and Richard Peng. Approximate maximum flow on separable undirected graphs. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete*

Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013, pages 1151–1170, 2013.
15 H. Minc. *Nonnegative Matrices.* John Wiley and Sons, New York, 1973.
16 Richard Peng and Daniel A. Spielman. An efficient parallel solver for SDD linear systems. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 333–342, 2014.
17 Richard Yang Peng. *Algorithm Design Using Spectral Graph Theory.* PhD thesis, Carnegie Mellon University, Pittsburgh, August 2013. CMU CS Tech Report CMU-CS-13-121.
18 Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Inf. Comput.*, 82(1):93–133, 1989.
19 Daniel A. Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 563–568, 2008.
20 Daniel A. Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.

A Spectral Sparsification of $\mathcal{T}$-Matrices

Our proof of Lemma A.2 is based on the Perron-Frobenius Theorem [15] for non-negative matrices.

▶ **Theorem A.1.** [15, Perron-Frobenius] Suppose A is symmetric nonnegative matrix. Then it has a nonnegative eigenvalue λ which is greater than or equal to the modulus of all other eigenvalues.

▶ **Lemma A.2.** Suppose $D - M$ is SDDM matrix. Then $D - D(D^{-1}M)^N$ is SDDM matrix $\forall N \in \mathbb{N}_+$.

Proof. Since $D-M$ is SDDM we have $D \succ M$. By Fact 3.1.e it holds $I \succ D^{-1/2}MD^{-1/2} \triangleq X$. Hence, the largest eigenvalue $\lambda(X) < 1$. By Theorem A.1 the spectral radius $\rho(X) < 1$, i.e. $|\lambda_i(X)| < 1$ for all i. Since X is symmetric it has the form $X = \sum_i \lambda_i u_i u_i^{\mathrm{T}}$. Moreover, we have $X^k = \sum_i \lambda_i^k u_i u_i^{\mathrm{T}}$ for every $k \in \mathbb{N}_+$. Thus the spectral radius of X^k satisfies $\rho(X^k) = \rho(X)^k < 1$.

Notice that $B_k = D - D(D^{-1}M)^k$ is symmetric non-negative matrix for every $k \in \mathbb{N}_+$. By definition $D - M$ is diagonally dominant and thus $D^{-1}M\mathbf{1} \preceq \mathbf{1}$ component-wise. This implies that B_k is diagonally dominant matrix. Notice that $B_k = D^{1/2}[I - X^k]D^{1/2}$ and since $\rho(X^k) < 1$ it follows that B_k is positive definite and hence SDDM matrix. ■

To prove Lemma 4.2 we use the following result that appears in Peng's thesis [17].

▶ **Lemma A.3.** [17] Suppose $D - A$ is Laplacian matrix (possibly $A_{ii} \neq 0$), and $\widetilde{D} - \widetilde{A}$ a sparsifier with $\widetilde{A}_{ii} = 0$ for every i such that $(1-\varepsilon)(D-A) \preceq \widetilde{D} - \widetilde{A} \preceq D - A$. Then the symmetric non-negative matrix $\widehat{A} = (D - \widetilde{D}) + \widetilde{A}$ satisfies $(1-\varepsilon)(D-A) \preceq D - \widehat{A} \preceq (D-A)$.

Proof of Lemma 4.2. Let $\widetilde{D_1} = \frac{1}{1+\varepsilon}\widetilde{D}$ and $\widetilde{A_1} = \frac{1}{1+\varepsilon}\widetilde{A}$. Then $\frac{1-\varepsilon}{1+\varepsilon}(D-A) \preceq \widetilde{D_1} - \widetilde{A_1} \preceq D - A$ and by Lemma A.3 the symmetric non-negative matrix $\widehat{A} = (D - \widetilde{D_1}) + \widetilde{A_1}$ satisfies $\frac{1-\varepsilon}{1+\varepsilon}(D-A) \preceq D - \widehat{A} \preceq D - A$. Since $\frac{1-\varepsilon}{1+\varepsilon} \geqslant 1 - 2\varepsilon$ for every $\varepsilon \in (0, \frac{1}{2})$ the statement follows. ■

A.1 Structural Result

Suppose $D - M$ is $\mathcal{T}$-matrix. We show that the matrix $D - MD^{-1}M$ can be expressed as a sum of a non-negative main diagonal matrix and a sum of Laplacian matrices.

Proof of Lemma 4.6. Let $M \in \mathbb{R}^{n \times n}$ and $nnz(M) = m$. We decompose the entries of matrix $MD^{-1}M$ into three types. We set type 1 to be the entries $(MD^{-1}M)_{ii} = \sum_{k=1}^{n} M_{ik}^2/D_k$ for all i. We note that all entries of type 1 can be computed in $O(m)$ time. We consider next the off-diagonal entries

$$(MD^{-1}M)_{ij} = \underbrace{(M_{ii}/d_i + M_{jj}/d_j) \cdot M_{ij}}_{\text{type 2}} + \underbrace{\sum_{k \neq \{i,j\}}^{n} M_{ik} M_{jk}/D_{kk}}_{\text{type 3}}.$$

Observe that the number of type 2 entries is at most m. Now for a fixed k we note that the corresponding entries that appear in type 1 and type 3 form a weighted clique (with self-loops) whose adjacency matrix is defined by $\frac{1}{d_k}\eta_k\eta_k^{\mathrm{T}}$.

Straightforward checking shows that $MD^{-1}M = \mathbf{B} + \sum_i \frac{1}{d_i}\eta_i\eta_i^{\mathrm{T}}$. By Remark 4.4 $D - MD^{-1}M$ is $\mathcal{T}$-matrix and thus diagonally dominant. Hence, the Laplacian matrices $\mathbf{L}_B$ and $\mathbf{L}_{N_i}$ for all i exist. Moreover, we can compute in $O(m)$ time the positive diagonal matrices $\mathbf{D}_B$ and $\mathbf{D}_{N_i} = (s_i/d_i)\mathrm{diag}(N_i)$ for all i. To see this, observe that s_i and d_i can be computed in $O(m)$ time for all i, and the number of elements in the disjoint union $|\sqcup_i N_i| \leqslant m$. ∎

B The Core Iterative Algorithm

We prove now Lemma 5.5 using similar arguments as in [4, Lemma 4.1].

Proof of Lemma 5.5. By Lemma 5.4, $D(D^{-1}M)^{2^k}$ is SPSD matrix for any $k \in \mathbb{N}_+$. By Remark 4.4 and Corollary 4.5 both $D - D(D^{-1}M)^{2^k}$ and $D - \widehat{M_{2^k}} D^{-1} \widehat{M_{2^k}}$ are $\mathcal{T}$-matrices. Hence, by Lemma 5.2 we have that $D - \widehat{M_{2^k}} D^{-1} \widehat{M_{2^k}} \approx_{\varepsilon'} D - D(D^{-1}M)^{2^{k+1}}$. Now by Lemma 4.7 we have $D - \widehat{M_{2^{k+1}}} \approx_{\varepsilon} D - \widehat{M_{2^k}} D^{-1} \widehat{M_{2^k}}$ and hence the statement follows. ∎

C Spectral Sparsification of $\mathcal{T}$-Matrix Polynomials Induced by MDBD

The design of our Algorithm **SS_MDBD** (c.f. Theorem 2.3) is inspired by Lemma 7.1 and Algorithm **PwrSS** (c.f. Theorem 5.1). We present now its pseudo code.

Algorithm 1

$(D, \widehat{M}) = \textbf{SS\_MDBD}(D, M, \mathcal{B}_{N,T}(\alpha, p), \varepsilon)$

1. Let $\varepsilon' = \varepsilon/2(1 + \log N)$, $\widehat{M_{tmp}} = 0$ and $\delta = 1 - \sum_{i=1}^{T} \alpha_i$, where $\delta \in [0, o(1)]$.
2. $(D, \widehat{M_1}, \widehat{M_2}) = \textbf{Preprocess}(D, M, \varepsilon')$.
3. For every $j \in \{1, \ldots, T\}$ do
   - 3.1 Set $p = p_{j,T} = j/(T+1)$ and $\widehat{M_{p,2}} = (1-p)^2 D + 2p(1-p)\widehat{M_1} + p^2 \widehat{M_2}$.
   - 3.2 $(D, \widehat{M_{p,N}}) = \textbf{PwrSS}(\widehat{M_{p,2}}, N, \varepsilon')$ where $D - \widehat{M_{p,N}} \approx_{\varepsilon} D - DW_p^N$ (c.f. Theorem 5.1).
   - 3.3 $\widehat{M_{tmp}} = \widehat{M_{tmp}} + \alpha_j \cdot \widehat{M_{p,N}}$.
4. Sparsify $D - \widehat{M} \approx_{\varepsilon} D - \frac{1}{1-\delta}\widehat{M_{tmp}}$ by Algorithm **ImpSS** (c.f. Theorem 3.2).
5. Return $(D, \widehat{M})$.

Algorithm 2

$(D, \widehat{M}_1, \widehat{M}_2) = \textbf{Preprocess}(D, M, \varepsilon)$
1. Sparsify $D - \widehat{M}_1 \approx_\varepsilon D - M$ by Algorithm **mKLP14** (c.f. Lemma 4.3).
2. Sparsify $D - \widehat{M}_2 \approx_\varepsilon D - MD^{-1}M$
2.1 If M is SPSD matrix call Algorithm **InitSS** (c.f. Theorem 5.3),
2.2 otherwise call Algorithm **mPS14** (c.f. Lemma 4.7).
3. Return $(D, \widehat{M}_1, \widehat{M}_2)$.

D Fast Algorithm for Solving Transpose Bernstein-Vandermonde

We proof below that the Bernstein basis matrix in Problem 1 has full rank.

▶ Lemma D.1. Suppose a vector $p \in (0,1)^{N+1}$ satisfies $0 < p_i \neq p_j < 1$ for all $i \neq j$. Then the Bernstein basis matrix $\mathbf{B}_N(p)$ has a full rank.

Proof. Suppose for contradiction that $\text{rank}(\mathbf{B}_N(p)) < N+1$. Then there is a vector $\lambda \in \mathbb{R}^{N+1}$ such that the linear combination of the columns of $\mathbf{B}_N(p)$ satisfies $\sum_{i=0}^{N} \lambda_j [\mathbf{B}_N(p)]_{:,i} = 0$. Let $f_\lambda(x)$ be a polynomial defined by $f_\lambda(x) \triangleq \sum_{i=0}^{N} \lambda_i \cdot B_{N,i}(x) = \sum_{i=0}^{N} \lambda_i \cdot \binom{N}{i} x^i (1-x)^{N-i}$. Notice that $f_\lambda(p_j) = 0$ for every $j \in [1 : N+1]$, i.e. $f_\lambda(x)$ has $N+1$ roots. However, since the polynomial $f_\lambda(x)$ has degree N it follows that $f_\lambda(x) \equiv 0$. Therefore, we obtained the desired contradiction. ∎

E Approximation of Discretized PDFs by MDBD

In this section we present a simple algorithm that takes as input a graph G, desired probability density function w (that satisfies certain technical conditions), and outputs a sparsifier of a matrix polynomial of the form $D - D\sum_{i=0}^{N} \gamma_i (D^{-1}M)^i$, such that each coefficient γ_i approximate the desired discretized function $w(i/N)/[\sum_{\ell=0}^{N} w(\ell/N)]$ for all $i \in [3 : N-3]$.

E.1 Approximation Algorithm

We begin our discussion by presenting the pseudo code of Algorithm **Approx_DPDF**.

Algorithm 3 Approximate Discretized PDF by MDBD

$(D, \widehat{M}) = \textbf{Approx\_DPDF}\,(D, M, w, N, \varepsilon_I, \varepsilon_S, \kappa)$
1. Let $T = N\sqrt{\kappa/\varepsilon_I}$, $S_{T,T+1} = \sum_{i=1}^{T} w\left(\frac{i}{T+1}\right)$ and $S_{N+1,N} = \sum_{i=0}^{N} w\left(\frac{i}{N}\right)$.
2. Let $\alpha_j = \frac{N}{(T+1)\cdot S_{N+1,N}} \cdot w(p_j)$ and $p_j = \frac{j}{T+1}$ for all $j \in [1 : T]$.
3. Return $(D, \widehat{M}) = \textbf{SS\_MDBD}(D, M, \mathcal{B}_{N,T}(\alpha, p), \varepsilon_S)$.

Before we prove Lemma 2.2, we analyze the class of continuous probability density functions that admit a discretized approximation by MDBD.

▶ Lemma E.1. Suppose $w(x) = C \cdot f(x)$ is a twice differentiable probability density function such that
a) $1 \leqslant C \leqslant o(N)$, *b)* $0 \leqslant f(x) \leqslant 1$, *c)* $\frac{1}{2}[f(0) + f(1)] \geqslant \Omega(1)$ and *d)* $\int_0^1 \left|f^{(2)}(x)\right| dx \leqslant o(N)$. Then, for any $N \in \mathbb{N}$, it holds for $T = N\sqrt{\kappa/\varepsilon}$ that

$$1 - \frac{S_{T,T+1}/(T+1)}{S_{N+1,N}/N} = o(1), \text{ where } S_{T,T+1} = \sum_{k=1}^{T} w\left(\frac{k}{T+1}\right) \text{ and } S_{N+1,N} = \sum_{k=0}^{N} w\left(\frac{k}{N}\right)$$

Proof. We define the discrete approximator of f by

$$A_M(f) = \frac{1}{M}\left[\frac{1}{2}\left(f(0)+f(1)\right) + \sum_{i=1}^{M-1} f\left(\frac{i}{M}\right)\right].$$

By Theorem X, we have

$$\left|A_{T+1}(f) - \int_0^1 f(t)dt\right| \leqslant \frac{1}{8\left(T+1\right)^2}\int_0^1 \left|f^{(2)}(t)\right| dt \lesssim \frac{\varepsilon}{\kappa}\cdot o\left(\frac{1}{N}\right),$$

and similarly

$$\left|A_N(f) - \int_0^1 f(t)dt\right| \leqslant \frac{1}{8N^2}\int_0^1 \left|f^{(2)}(t)\right| dt \lesssim o\left(\frac{1}{N}\right).$$

By definition, $\int_0^1 f(t)dt = C^{-1} \in [1/o(N), 1]$ and thus

$$A_{T+1}(f) \in \left[\frac{1}{C} \pm \frac{\varepsilon}{\kappa}\cdot o\left(\frac{1}{N}\right)\right], \quad \text{and} \quad A_N(f) \in \left[\frac{1}{C} \pm o\left(\frac{1}{N}\right)\right].$$

Let $d \triangleq [f(0)+f(1)]/2$. Straightforward checking shows that

$$\frac{S_{T,T+1}}{T+1} = C\cdot\left[A_{T+1}(f) - \frac{d}{T+1}\right] \quad \text{and} \quad \frac{S_{N+1,N}}{N} = C\cdot\left[A_N(f) + \frac{d}{N}\right].$$

We show now the upper bound. By assumption $d \in [\Omega(1), 1]$ and since $C \leqslant o(N)$ we have

$$\begin{aligned}\Lambda &\triangleq \frac{S_{T,T+1}/(T+1)}{S_{N+1,N}/N} = \frac{A_{T+1}(f) - \frac{d}{T+1}}{A_N(f) + \frac{d}{N}} = \frac{\frac{1}{C} - \frac{d}{T+1} \pm \frac{\varepsilon}{\kappa}\cdot o\left(\frac{1}{N}\right)}{\frac{1}{C} + \frac{d}{N} \pm o\left(\frac{1}{N}\right)} \\ &\leqslant 1 - \frac{\left(1+\frac{1}{2}\sqrt{\frac{\varepsilon}{k}}\right)\cdot\frac{d}{N} - \left(1+\frac{\varepsilon}{k}\right)\cdot o\left(\frac{1}{N}\right)}{\frac{1}{C} + \left[\frac{d}{N} - o\left(\frac{1}{N}\right)\right]} \leqslant 1 - \frac{\Omega(1)}{N}\cdot\frac{1}{\frac{1}{C}+\frac{1}{N}} = 1 - o(1).\end{aligned}$$

The lower bound $\Lambda \geqslant 1 - o(1)$ can be proven using similar arguments. ∎

We show now the proof of Lemma 2.2.

Proof of Lemma 2.2. Let $S_{T,T+1} = \sum_{k=1}^{T} w\left(\frac{k}{T+1}\right)$ and $S_{N+1,N} = \sum_{k=0}^{N} w\left(\frac{k}{N}\right)$. We define the components of the desired discretized probability density function by

$$\beta_i = \frac{w\left(\frac{i}{N}\right)}{\sum_{k=0}^{N} w\left(\frac{k}{N}\right)} = \frac{w\left(\frac{i}{N}\right)}{S_{N+1,N}}.$$

By Theorem 2.1, it holds for all $i \in [3 : N-3]$ that

$$\left|\left(1+\eta_i\right)\frac{1}{N}w\left(\frac{i}{N}\right) - \sum_{j=1}^{T}\left[\frac{1}{T+1}w\left(\frac{j}{T+1}\right)\right] B_{N,i}\left(\frac{j}{T+1}\right)\right| \leqslant \frac{\varepsilon}{N}. \tag{3}$$

We define for all $j \in [1 : T]$ the scalars

$$\alpha_j = \frac{N}{(T+1)\cdot S_{N+1,N}}\cdot w\left(\frac{j}{T+1}\right) \quad \text{and} \quad \gamma_i = \sum_{j=1}^{T}\alpha_j \cdot B_{N,i}\left(\frac{j}{T+1}\right).$$

By multiplying Equation 3 with $N/S_{N+1,N}$, we obtain

$$\left|(1+\eta_i)\,\frac{w\left(\frac{i}{N}\right)}{S_{N+1,N}} - \gamma_i\right| \leqslant \frac{\varepsilon}{S_{N+1,N}}, \tag{4}$$

and notice that

$$\sum_{j=1}^{T} \alpha_j = \frac{N}{(T+1)\cdot S_{N+1,N}} \sum_{j=1}^{T} w\left(\frac{j}{T+1}\right) = \frac{S_{T,T+1}/(T+1)}{S_{N+1,N}/N}.$$

Moreover, by Lemma E.1 there is a positive number $\delta_w = o(1)$ such that

$$\delta_w = 1 - \frac{S_{T,T+1}/(T+1)}{S_{N+1,N}/N}.$$

Hence, we have

$$\sum_{i=0}^{N} \gamma_i = \sum_{i=0}^{N}\sum_{j=1}^{T} \alpha_j \cdot B_{N,i}\left(\frac{j}{T+1}\right) = \sum_{j=1}^{T} \alpha_j \sum_{i=0}^{N} B_{N,i}\left(\frac{j}{T+1}\right) = \sum_{j=1}^{T} \alpha_j = 1 - \delta_w.$$

We set $p_j = \frac{j}{T+1}$ for all $i \in [1:T]$ and call Algorithm **SS_MDBD**$(D, M, \mathcal{B}_{N,T}(\alpha, p), \varepsilon_S)$, whose output by Theorem 2.3 is a spectral sparsifier that satisfies

$$D - \widehat{M} \approx_{\varepsilon_S} D - D\sum_{i=0}^{N} \frac{\gamma_i}{1-\delta_w}(D^{-1}M)^i.$$

Since $\delta_w = o(1)$, the statement follows by Equation 4. ■

E.2 Two Canonical PDFs

To illustrate our framework we analyze two canonical probability density functions, and show that they can be approximated by MDBD. In particular, we demonstrate that MDBD yields a multiplicative approximation for the Uniform distribution, and the Exponential family admits an additive approximation. Moreover, we show that both probability density functions satisfy all conditions in Lemma 2.2 and thus their corresponding discretized functions can be approximated, as well.

E.2.1 Uniform Distirbution

▶ Lemma E.2. Let $w(x) = 1$, $N \in \mathbb{N}_+$ and $\varepsilon > 0$. If $T \geqslant \Omega(N\varepsilon^{-1/2})$ then it holds that

$$\frac{1}{T+1}\sum_{j=1}^{T} B_{N,i}\left(\frac{j}{T+1}\right) \in \left[(1\pm\varepsilon)\frac{1}{N+1}\right], \text{ for all } i \in [3:N-3].$$

Proof. By Theorem 9.2 we have that

$$\left|\sum_{i=1}^{T} L_i\right| \leqslant \frac{1}{2}\sum_{i=1}^{T}\int_{x_{i-1}}^{x_i}\left|\frac{1}{4N^2} - (t-c_i)^2\right|\left|\frac{d^2B_{N,i}(x)}{dx^2}(t)\right| dt \leqslant \frac{1}{8N^2}\int_0^1\left|\frac{d^2B_{N,i}(x)}{dx^2}(t)\right| dt.$$

By combining Lemma 9.3 and Corollary 9.5 for every $i \in [3:N-3]$ it holds

$$\left|\frac{1}{N+1} - \frac{1}{T+1}\sum_{j=1}^{T} B_{N,i}\left(\frac{j}{T+1}\right)\right| \leqslant \left|\sum_{i=1}^{T} L_i\right| \leqslant \frac{1}{8T^2}\int_0^1\left|\frac{d^2B_{N,i}(x)}{dx^2}(x)\right| dx \leqslant \frac{N}{2T^2}.$$

We note that $\frac{N}{2T^2} \leqslant \frac{\varepsilon}{N+1}$ since $T \geqslant \Omega(N\varepsilon^{-1/2})$, and hence the statement follows. ■

▶ Remark E.3. All conditions of Lemma 2.2 hold. Note that $w(x) = 1 \cdot 1$, i.e., $f(x) = 1$. a) $C = 1$, b) f(x)=1, c) $\frac{1}{2}[f(0) + f(1)] = 1$ and d) $\int_0^1 |f^{(2)}(x)|dx = 0$, since $f^{(2)}(x) = 0$.

E.2.2 Exponential Families

▶ Lemma E.4. Let $N \in \mathbb{N}$, $\kappa \in [1, \sqrt{N}]$ and $w(x) = \frac{\kappa}{1-e^{-\kappa}} \cdot \exp\{-\kappa \cdot x\}$ is a probability density function. For any $\varepsilon > 0$ if $T \geqslant \Omega(N\sqrt{\kappa/\varepsilon})$ then for every $i \in [3 : N-3]$ it holds for the function $F_i(x) = w(x) \cdot B_{N,i}(x)$ that

$$\left|(1+\eta_i)\frac{w(i/N)}{N} - \frac{1}{T+1}\sum_{j=1}^{T} F_i\left(\frac{j}{T+1}\right)\right| \leqslant \frac{\varepsilon}{N}, \quad \text{where} \quad |\eta_i| \leqslant \frac{1}{4}.$$

Proof. The pth derivative of function $w(x)$ satisfies $w^{(p)}(x) = (-\kappa)^p \cdot w(x)$. Let $I = [0, 1]$ be an interval. Straightforward checking shows that

$$\max_{x\in I} |w^{(p)}(x)| = \kappa^p \cdot \max_{x\in I} |w(x)| < 2 \cdot \kappa^{p+1}. \tag{5}$$

By the definition of function $b_1(x)$ (c.f. Theorem 9.1) we have

$$b_1(x) = -\frac{\kappa^2}{2} \cdot x^2 + \left(2\kappa + \frac{\kappa^2}{2}\right) \cdot x - (1+\kappa),$$

and we can show that $\max_{x\in I} b_1(x) \leqslant 1 + \kappa^2/8 \leqslant 1 + N/8$. The function $b_2(x)$ satisfies

$$b_2(x) = 1 + 3(1-2x)\kappa + (1 - 6x + 6x^2)\kappa^2 - \frac{5}{6}x(1-x)(1-2x)\kappa^3 + \frac{1}{8}x^2(1-x)^2\kappa^4,$$

and we can show that $\max_{x\in I} b_2(x) \ll \kappa^4/8 \leqslant N^2/8$. By Theorem 2.1 it suffices to upper bound the following four cases.

Case 1: By Theorem 9.1 $\mu \leqslant \frac{1}{4}$, since $\max_{x\in I} |b_1(x)| \leqslant 1+N/8$ and $\max_{x\in I} |b_2(x)| \ll N^2/8$.

Case 2: We combine Equation 5 and Lemma 9.3 to obtain

$$\int_0^1 |w''(t) \cdot B_{N,i}(t)|dt < 2 \cdot \kappa^3 \cdot \int_0^1 |B_{N,i}(t)|dt = \frac{2 \cdot \kappa^3}{N+1}.$$

Case 3: By combining Equation 5 and Lemma 9.5 it holds

$$\int_0^1 |w'(t) \cdot B'_{N,i}(t)|dt < 2 \cdot \kappa^2 \cdot \int_0^1 |B'_{N,i}(t)|dt < 4 \cdot \kappa^2.$$

Case 4: We use again Equation 5 and Lemma 9.5 to obtain

$$\int_0^1 |w(t) \cdot B''_{N,i}(t)|dt < 2 \cdot \kappa \cdot \int_0^1 |B''_{N,i}(t)|dt < 8 \cdot \kappa \cdot N.$$

Recall that $F_i(x) = w(x)B_{N,i}(x)$. By combining $|\frac{d^2F_i(x)}{dx^2}| = w'' \cdot B_{N,i} + 2w' \cdot B'_{N,i} + w \cdot B''_{N,i}$ and Theorem 9.2 we have

$$\left|\sum_{i=1}^{T} L_i\right| \leqslant \frac{1}{8T^2}\int_0^1 \left|\frac{d^2F(x)}{dx^2}(t)\right| dt \lesssim \frac{\kappa \cdot N}{T^2}.$$

Hence, the statement follows. ■

▶ Remark E.5. All conditions of Lemma 2.2 hold. Note that $w(x) = \frac{\kappa}{1-e^{-\kappa}} \cdot \exp\{-\kappa \cdot x\}$. Thus $C = \frac{\kappa}{1-e^{-\kappa}}$, $f(x) = \exp\{-\kappa \cdot x\}$, $f^{(2)}(x) = \kappa^2 \cdot f(x)$ and $\kappa \in [1, \sqrt{N}]$. Notice that a) $C = o(N)$, b) $0 \leqslant f(x) \leqslant 1$, c) $\frac{1}{2} \leqslant \frac{1}{2}[f(0) + f(1)] < 1$ and for d) we have

$$\int_0^1 |f^{(2)}(x)| dx = \kappa^2 \int_0^1 |f(x)| dx = \frac{\kappa^2}{C} = (1 - e^{-\kappa})\kappa = o(N).$$

F Schur Complement

In this section we prove Lemma 5.2. We use the following result proposed by Peng et al. [4].

▶ Lemma F.1. [4, Lemma 4.3] If M is SPSD matrix and $(1-\varepsilon)(D - M) \preceq D - \widehat{M} \preceq (1+\varepsilon)(D - M)$ then it holds that $(1-\varepsilon)(D + M) \preceq D + \widehat{M} \preceq (1+\varepsilon)(D + M)$.

We extend next two technical results on Schur complement that appeared in [17, 14, 3].

▶ Claim F.2. Suppose $X \triangleq \begin{bmatrix} P_1 & -M \\ -M & P_2 \end{bmatrix}$ where P_1 and P_2 are symmetric positive definite matrices and M is symmetric matrix. Then $v^{\mathrm{T}}[P_2 - MP_1^{-1}M]v = \min_u \begin{pmatrix} u \\ v \end{pmatrix}^{\mathrm{T}} X \begin{pmatrix} u \\ v \end{pmatrix}$ for every v.

Proof. Suppose $f_v(u) \triangleq \begin{pmatrix} u \\ v \end{pmatrix}^{\mathrm{T}} \begin{bmatrix} P_1 & -M \\ -M & P_2 \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = u^{\mathrm{T}} P_1 u - 2u^{\mathrm{T}} M v + v^{\mathrm{T}} P_2 v$. Notice that f is minimized when $u = P_1^{-1} M v$, since $\nabla f_v(u) = 2P_1 u - 2Mv$. Hence, it follows that $\min_u \begin{pmatrix} u \\ v \end{pmatrix}^{\mathrm{T}} X \begin{pmatrix} u \\ v \end{pmatrix} = v^{\mathrm{T}} P_2 v - v^{\mathrm{T}} M P_1^{-1} M v = v^{\mathrm{T}}[P_2 - MP_1^{-1}M]v$. ■

▶ Lemma F.3. (Schur Complement) Suppose D_1, D_2 are positive main diagonal matrices and M, Q are symmetric matrices. If $X_M \triangleq \begin{bmatrix} D_1 & -M \\ -M & D_2 \end{bmatrix} \approx_\varepsilon \begin{bmatrix} D_1 & -Q \\ -Q & D_2 \end{bmatrix} \triangleq X_Q$ then it holds that $D_2 - MD_1^{-1}M \approx_\varepsilon D_2 - QD_1^{-1}Q$.

Proof. Here we show the upper bound, but the lower bound follows by analogy. Let w be a vector such that $\begin{pmatrix} w \\ v \end{pmatrix}^{\mathrm{T}} X_Q \begin{pmatrix} w \\ v \end{pmatrix} = \min_u \begin{pmatrix} u \\ v \end{pmatrix}^{\mathrm{T}} X_Q \begin{pmatrix} u \\ v \end{pmatrix}$. We apply twice Claim F.2 to obtain the following chain of inequalities

$$\begin{aligned} v^{\mathrm{T}}[D_2 - MD_1^{-1}M]v &= \min_u \begin{pmatrix} u \\ v \end{pmatrix}^{\mathrm{T}} X_M \begin{pmatrix} u \\ v \end{pmatrix} \leqslant \begin{pmatrix} w \\ v \end{pmatrix}^{\mathrm{T}} X_M \begin{pmatrix} w \\ v \end{pmatrix} \\ &\leqslant (1+\varepsilon) \begin{pmatrix} w \\ v \end{pmatrix}^{\mathrm{T}} X_Q \begin{pmatrix} w \\ v \end{pmatrix} = (1+\varepsilon) \min_u \begin{pmatrix} u \\ v \end{pmatrix}^{\mathrm{T}} X_Q \begin{pmatrix} u \\ v \end{pmatrix} \\ &= (1+\varepsilon) v^{\mathrm{T}}[D_2 - QD_1^{-1}Q]v. \end{aligned}$$

■

We prove Lemma 5.2 by arguing in a similar manner to in [4, Lemma 4.4]. We present the proof here for completeness.

Proof of Lemma 5.2. For any symmetric matrix X, we denote by $\mathcal{P}_X = \begin{bmatrix} D & -X \\ -X & D \end{bmatrix}$. We prove the upper bound, but the lower bound follows by analogy. Straightforward checking shows that

$$\begin{aligned} \begin{pmatrix} u \\ v \end{pmatrix}^{\mathrm{T}} \mathcal{P}_{\widehat{M}} \begin{pmatrix} u \\ v \end{pmatrix} &= u^{\mathrm{T}} D u - v^{\mathrm{T}} \widehat{M} u - u^{\mathrm{T}} \widehat{M} v + v^{\mathrm{T}} D v \\ = \quad & \frac{1}{2} \Big[(u+v)^{\mathrm{T}} (D - \widehat{M})(u+v) + (u-v)^{\mathrm{T}} (D + \widehat{M})(u-v) \Big] \end{aligned}$$

By Lemma F.1 it holds that $D + \widehat{M} \approx_{\varepsilon} D + M$ and thus we have

$$\begin{aligned} \begin{pmatrix} u \\ v \end{pmatrix}^{\mathrm{T}} \mathcal{P}_{\widehat{M}} \begin{pmatrix} u \\ v \end{pmatrix} &= \frac{1}{2} \Big[(u+v)^{\mathrm{T}} (D - \widehat{M})(u+v) + (u-v)^{\mathrm{T}} (D + \widehat{M})(u-v) \Big] \\ &\leqslant (1+\varepsilon) \frac{1}{2} \left[(u+v)^{\mathrm{T}} (D - M)(u+v) + (u-v)^{\mathrm{T}} (D_2 + M)(u-v) \right] \\ &= (1+\varepsilon) \begin{pmatrix} u \\ v \end{pmatrix}^{\mathrm{T}} \mathcal{P}_{M} \begin{pmatrix} u \\ v \end{pmatrix}. \end{aligned}$$

Hence, it follows that $\mathcal{P}_{\widehat{M}} = \begin{bmatrix} D & -\widehat{M} \\ -\widehat{M} & D \end{bmatrix} \approx_{\varepsilon} \begin{bmatrix} D & -M \\ -M & D \end{bmatrix} = \mathcal{P}_M$. Now, by Lemma F.3 it holds that $D - \widehat{M} D^{-1} \widehat{M} \approx_{\varepsilon} D - M D^{-1} M$. ∎