A MEAN PERCENTILE RANK

We begin our definition of MPR by defining percentile rank (PR). First, given a set J, let $p_{i,J} = \Pr(J \cup \{i\} \mid J)$. The percentile rank of an item i given a set J is defined as

$$\text{PR}_{i,J} = \frac{\sum_{i' \notin J} \mathbb{1}(p_{i,J} \geq p_{i',J})}{|\mathcal{Y} \backslash J|} \times 100\%$$

where $\mathcal{Y} \backslash J$ indicates those elements in the ground set $\mathcal{Y}$ that are not found in J.

For our evaluation, given a test set Y, we select a random element $i \in Y$ and compute $\text{PR}_{i,Y \backslash \{i\}}$. We then average over the set of all test instances $\mathcal{T}$ to compute the mean percentile rank (MPR):

$$\text{MPR} = \frac{1}{|\mathcal{T}|} \sum_{Y \in \mathcal{T}} \text{PR}_{i,Y \backslash \{i\}}.$$

B HYPERPARAMETERS FOR EXPERIMENTS IN TABLE 2

Preventing numerical instabilities: The first term on the right side of Eq. 2 will be singular whenever $|Y_i| > K$, where Y_i is an observed subset. Therefore, to address this in practice we set K to the size of the largest subset observed in the data, K', as in Gartrell et al. (2017). However, this does not entirely address the issue, as the first term on the right side of Eq. 2 may still be singular even when $|Y_i| \leq K$. In this case though, we know that we are not at a maximum, since the value of the objective function is $-\infty$. Numerically, to prevent such singularities, in our implementation we add a small $\epsilon \boldsymbol{I}$ correction to each $\boldsymbol{L}_{Y_i}$ when optimizing Eq. 2 (we set $\epsilon = 10^{-5}$ in our experiments).

We perform a grid search using a held-out validation set to select the best performing hyperparameters for each model and dataset. The hyperparameter settings used for each model and dataset are described below.

Symmetric low-rank DPP (Gartrell et al., 2016). For this model, we use K for the number of item feature dimensions for the symmetric component $\boldsymbol{V}$, and α for the regularization hyperparameter for $\boldsymbol{V}$. We use the following hyperparameter settings:

- Both Amazon datasets: $K = 30, \alpha = 0$.
- UK Retail dataset: $K = 100, \alpha = 1$.
- Instacart dataset: $K = 100, \alpha = 0.001$.
- Million Song dataset: $K = 150, \alpha = 0.0001$.

Baseline NDPP (Gartrell et al., 2019). For this model, to ensure consistency with the notation used in Gartrell et al. (2019), we use D to denote the number of item feature dimensions for the symmetric component $\boldsymbol{V}$, and D' to denote the number of item feature dimensions for the nonsymmetric components, $\boldsymbol{B}$ and $\boldsymbol{C}$. As described in Gartrell et al. (2019), α is the regularization hyperparameter for the $\boldsymbol{V}$, while β and γ are the regularization hyperparameters for $\boldsymbol{B}$ and $\boldsymbol{C}$, respectively. We use the following hyperparameter settings:

- Both Amazon datasets: $D = 30, \alpha = 0$.
- Amazon apparel dataset: $D' = 30$.
- Amazon three-category dataset: $D' = 100$.
- UK Retail dataset: $D = 100, D' = 20, \alpha = 1$.
- All datasets: $\beta = \gamma = 0$.

Scalable NDPP. As described in Section 3, we use K to denote the number of item feature dimensions for the symmetric component $\boldsymbol{V}$ and the dimensionality of the nonsymmetric component $\boldsymbol{C}$. α is the regularization hyperparameter. We use the following hyperparameter settings:

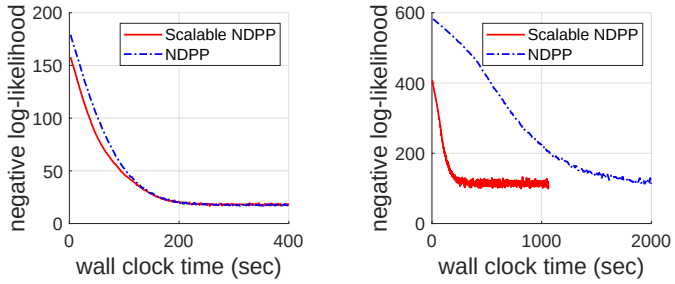
- Amazon apparel dataset: $K = 30, \alpha = 0$.
- Amazon three-category dataset: $K = 100, \alpha = 1$.
- UK dataset: $K = 100, \alpha = 0.01$.

- Instacart dataset: $K = 100, \alpha = 0.001$.
- Million Song dataset: $K = 150, \alpha = 0.01$.

For all of the above model configurations we use a batch size of 200 during training, except for the scalable NDPPs trained on the Amazon apparel, Amazon three-category, Instacart, and Million Song datasets, where a batch size of 800 is used.

C TRAINING TIME

In Fig. 1, we report the wall-clock training time of the decomposition of Gartrell et al. (2019) (NDPP) and our scalable NDPP for the Amazon: 3-category (Fig. 1(a)) and UK Retail (Fig. 1(b)) datasets. For the Amazon: 3-category dataset, both approaches show comparable results, with the scalable NDPP converging 1.07 times faster than NDPP. But for the UK Retail dataset, which has a much larger ground set, our scalable NDPP achieves convergence about 8.31 times faster. We do not have a timing comparison for the Instacart dataset because the model with the decomposition of Gartrell et al. (2019) cannot be trained on this dataset due to prohibitive memory and computational costs.



(a) **Amazon: 3-category** (b) **UK Retail**

Figure 1: The negative log-likelihood of the training set for Gartrell et al. (2019)'s NDPP (blue, dashed) and our scalable NDPP (red, solid) versus wall-clock time for the (a) Amazon: 3-category and (b) UK Retail datasets.

D BENCHMARK ALGORITHMS FOR MAP INFERENCE

We test following approximate algorithms for MAP inference:

Greedy local search. This algorithm starts from the output of greedy, Y^G, and replaces $i \in Y^G$ with $j \notin Y^G$ that gives the maximum improvement of the determinant, if such i, j exist. Kathuria & Deshpande (2016) showed that running the search for such a swap $O(k^2 \log(k/\varepsilon))$ times with an accuracy parameter ε gives a tight approximation guarantee for MAP inference for symmetric DPPs. We set the number of swaps to $\lfloor k^2 \log(10k) \rfloor$ for $\varepsilon = 0.1$ and use greedy local search as a baseline, since it is strictly an improvement on the greedy solution. The proposed greedy conditioning can be used for fast greedy local search. Specifically, for each $i \in Y^G$, Algorithm 1 can compute marginal improvements conditioned by $Y^G \setminus \{i\}$ in time $O(MKk)$ thus its runtime can be $O(MKk^4 \log(k/\varepsilon))$. However, it is the slowest among all of our benchmark algorithms.

Stochastic greedy. This algorithm computes marginal gains of a few items chosen uniformly at random and selects the best among them. Mirzasoleiman et al. (2015) proved that $(M/k) \log(1/\varepsilon)$ samples are enough to guarantee an $(1 - 1/e - \varepsilon)$-approximation ratio for submodular functions (i.e., symmetric DPPs). We choose $\varepsilon = 0.1$ and set the number of samples to $\lfloor (M/k) \log(10) \rfloor$. Under this setting, the time complexity of stochastic greedy is $O(MKk^2 \log(1/\varepsilon))$, which is better than the naïve exact greedy algorithm. However, we note that it is worse than that of our efficient greedy implement (Algorithm 1). This is because the stochastic greedy uses different random samples for every iteration and this does not take advantage of the amortized computations in Lemma 2. In our experiments, we simply modify line 10 in Algorithm 1 for stochastic greedy (argmax is operated

on a random subset of marginal gains), hence it can run in $O(MKk + (M/k)\log(1/\varepsilon))$ time. In practice, we observe that stochastic greedy is slightly slower than exact greedy due to the additional costs of random sampling process.

MCMC sampling. We also compare inference algorithms with sampling from a nonsymmetric DPP. To the best of our knowledge, exact sampling of a non-Hermitian DPP was studied in Poulson (2019), which requires the Cholesky decomposition with $O(M^3)$ complexity. This is infeasible for a large M. To resolve this, Markov Chain Monte-Carlo (MCMC) based sampling is preferred Li et al. (2016) for symmetric DPPs. In particular, we consider a Gibbs sampling for k-DPP, which begins with a random subset Y with size k, and picks $i \in Y$ and $j \notin Y$ uniformly at random. Then, it swaps them with probability

$$\frac{\det(\boldsymbol{L}_{Y\cup\{j\}\setminus\{i\}})}{\det(\boldsymbol{L}_{Y\cup\{j\}\setminus\{i\}}) + \det(\boldsymbol{L}_Y)} \tag{11}$$

and repeat this process for several steps. Li et al. (2016) showed that $O(Nk\log(k/\varepsilon))$ swaps are enough to approximate the ground-truth distribution under symmetric DPPs. However, for a fair runtime comparison to Algorithm 1, we set the number of swaps to $\lfloor 3N/K \rfloor$.

We provide the wall-clock time of the above algorithms for real-world datasets in Table 4. Observe that the greedy algorithm is the fastest method for all datasets except Million Song. For Million Song, MCMC sampling is faster than other approaches, but it has much larger relative errors in terms of log-determinant (see Table 3), which is not suitable for our purposes.

Table 4: Wall-clock time (in milliseconds) of MAP inference algorithms on NDPPs learned from real-world datasets.

Algorithms	**Amazon: Apparel**	**Amazon: 3-category**	**UK Retail**	**Instacart**	**Million Song**
Greedy local search (Kathuria & Deshpande, 2016)	5.78 ms	9.67 ms	58.74 ms	1.024 s	7.277 s
Greedy (Algorithm 1)	**0.14 ms**	**0.34 ms**	**1.60 ms**	**36.16 ms**	338.09 ms
Stochastic greedy (Mirzasoleiman et al., 2015)	0.25 ms	0.47 ms	1.79 ms	36.94 ms	348.67 ms
MCMC sampling (Li et al., 2016)	0.19 ms	0.35 ms	2.85 ms	42.85 ms	**303.20 ms**

E Corollary of Theorem 2

Theorem 2 requires the technical condition $\sigma_{\min} > 1$ but in practice there is no particular evidence that this condition holds. While this condition can be achieved by multiplying $\boldsymbol{L}$ by a constant, this leads to a (potentially large) additive term in Eq. 10. Here, we provide Corollary 1 which excludes the $\sigma_{\min} > 1$ assumption from Theorem 2, and quantifies this additive term.

Corollary 1. *Consider a nonsymmetric low-rank DPP $\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^\top + \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^\top$, where $\boldsymbol{V}, \boldsymbol{B}$ are of rank K, and $\boldsymbol{C} \in \mathbb{R}^{K\times K}$. Given a cardinality budget k, let $\sigma_{\min}$ and $\sigma_{\max}$ denote the smallest and largest singular values of $\boldsymbol{L}_Y$ for all $Y \subseteq [\![M]\!]$ and $|Y| \leq 2k$. Let $\kappa := \sigma_{\max}/\sigma_{\min}$. Then,*

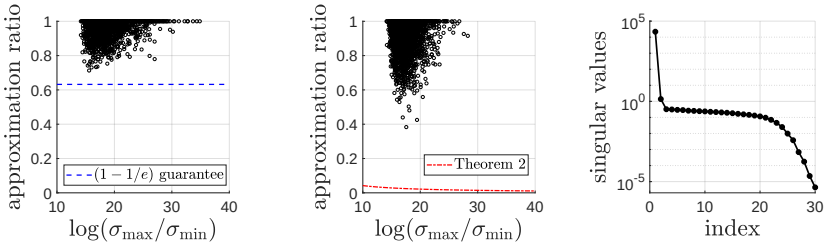
$$\log\det(\boldsymbol{L}_{Y^G}) \geq \frac{4(1-e^{-1/4})}{2\log\kappa+1}\log\det(\boldsymbol{L}_{Y^*}) - \left(1 - \frac{4(1-e^{-1/4})}{2\log\kappa+1}\right) k\,(1-\log\sigma_{\min}) \tag{12}$$

where Y^G is the output of Algorithm 1 and Y^* is the optimal solution of MAP inference in Eq. 4.

The proof of Corollary 1 is provided in Appendix G.5. Note that instead of $\log(\sigma_{\max})/\log(\sigma_{\min})$, Corollary 1 has a $\log(\sigma_{\max}/\sigma_{\min})$ term in the denominator.

F Performance guarantee for greedy MAP inference

The matrices learned on real datasets are too large to compute the exact MAP solution, but we can compute exact MAP for small matrices. In this section, we explore the performance of the greedy algorithm studied in Theorem 2 for 5×5 synthetic kernel matrices. More formally, we first pick $K = 3$ singular values s_1, s_2, s_3 from a kernel learned for the "Amazon: 3-category" dataset (a plot of these singular values can be seen in Fig. 2(c)) and generate $\boldsymbol{L} = \boldsymbol{V}_1 \texttt{diag}([s_1, s_2, s_3])\boldsymbol{V}_2^\top$, where $\boldsymbol{V}_1, \boldsymbol{V}_2 \in \mathbb{R}^{5\times 3}$ are random orthonormal matrices. To ensure that $\boldsymbol{L}$ is a P_0 matrix, we repeatedly



(a) Symmetric DPP (b) Nonsymmetric DPP (c) Singular values

Figure 2: Approximation ratios of greedy with respect to different values of $\log(\sigma_{\max}/\sigma_{\min})$ from Corollary 1 under (a) symmetric DPP and (b) nonsymmetric DPP. (c) The singular values of the kernels learned for the "Amazon: 3-category" dataset. We construct $10{,}000$ random P_0 matrices $\boldsymbol{L} \in \mathbb{R}^{5\times 5}$, with rank $K = 3$, whose singular values are from the learned kernels.

sample $\boldsymbol{V}_1, \boldsymbol{V}_2$ until all principal minors of $\boldsymbol{L}$ are nonnegative. We also evaluate the performance of the symmetric DPP, where the kernel matrices are generated similarly to the NDPP, except we set $\boldsymbol{V}_1 = \boldsymbol{V}_2$. We set $k = 3$ and generate $10{,}000$ random kernels for both symmetric DPPs and NDPPs.

The results for symmetric and nonsymmetric DPPs are shown in Fig. 2(a) and Fig. 2(b), respectively. We plot the approximation ratio of Algorithm 1, i.e., $\log\det(\boldsymbol{L}_{Y^G})/\log\det(\boldsymbol{L}_{Y^*})$, with respect to $\log(\sigma_{\max}/\sigma_{\min})$, from Corollary 1. We observe that the greedy algorithm for both often shows approximation ratios close to 1. However, the worst-case ratio for NDPPs is worse than that of symmetric DPPs; $\log\det(\boldsymbol{L}_Y)$ for $\boldsymbol{L} \in P_0^+$ is non-submodular, and the greedy algorithm with a nonsubmodular function does not have as tight of a worst-case bound as in the symmetric case.

G PROOFS

G.1 PROOF OF LEMMA 1

Lemma 1. *Let $\ell \leq M$ be an even integer and let $\boldsymbol{A} \in \mathbb{R}^{M \times M}$ be a skew-symmetric matrix with rank ℓ. Then, there exist $\boldsymbol{B} \in \mathbb{R}^{M \times \ell}$ and positive numbers $\lambda_1, \ldots, \lambda_{\ell/2}$, such that $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^\top$, where $\boldsymbol{C} \in \mathbb{R}^{\ell \times \ell}$ is the block-diagonal matrix with $(\ell/2)$ diagonal blocks of size 2 given by Σ_i, $i = 1, \ldots, \ell/2$.*

Proof. First, $\boldsymbol{A} = \mathbf{P}\boldsymbol{\Sigma}\mathbf{P}^\top$ for some orthogonal matrix $\mathbf{P} \in \mathbb{R}^{M \times M}$ and

$$\boldsymbol{\Sigma} = \begin{pmatrix} 0 & \lambda_1 & & & & & & & \\ -\lambda_1 & 0 & & & & & & & \\ & & 0 & \lambda_2 & & & \multicolumn{3}{c}{\multirow{3}{*}{0}} \\ & & -\lambda_2 & 0 & & & & & \\ & & & & \ddots & & & & \\ & & & & & 0 & \lambda_{\ell/2} & & \\ \multicolumn{4}{c}{\multirow{3}{*}{0}} & & -\lambda_{\ell/2} & 0 & & \\ & & & & & & & 0 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 0 \end{pmatrix} \tag{13}$$

(see, e.g.,(Thompson, 1988, Proposition 2.1), which is easily extended to the case when M is odd).

Let $\boldsymbol{C}$ be the $\ell \times \ell$ supmatrix of $\boldsymbol{\Sigma}$ obtained by keeping its first ℓ rows and columns and let $\mathbf{Q} = \begin{pmatrix} \boldsymbol{I}_\ell \\ 0 \end{pmatrix}$, where $\boldsymbol{I}_\ell$ is the $\ell \times \ell$ identity matrix. Then, $\boldsymbol{\Sigma} = \boldsymbol{Q}\boldsymbol{C}\boldsymbol{Q}^\top$ and one can write $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{Q}\boldsymbol{C}\boldsymbol{Q}^\top\boldsymbol{P}^\top$. Setting $\boldsymbol{B} = \boldsymbol{P}\boldsymbol{Q}$ proves the lemma. □

G.2 PROOF OF THEOREM 1

Theorem 1. *Given an NDPP with kernel $\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^\top + \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^\top$, parameterized by $\boldsymbol{V}$ of rank K, $\boldsymbol{B}$ of rank K, and a $K \times K$ matrix $\boldsymbol{C}$, we can compute the regularized log-likelihood (Eq. 2) and its gradient in $O(MK^2 + K^3 + nK'^3)$ time, where K' is the size of the largest of the n training subsets.*

Proof. We first show that the log-likelihood can be computed in time linear in M. Using the matrix determinant lemma, one can easily verify that the DPP normalization term can be computed as

$$\det(\boldsymbol{I} + \boldsymbol{L}) = \det\left(\boldsymbol{I} + \begin{pmatrix} \boldsymbol{V} & \boldsymbol{B}\boldsymbol{C} \end{pmatrix}\begin{pmatrix} \boldsymbol{V}^\top \\ \boldsymbol{B}^\top \end{pmatrix}\right) = \det\left(\boldsymbol{I}_{2K} + \begin{pmatrix} \boldsymbol{V}^\top \\ \boldsymbol{B}^\top \end{pmatrix}\begin{pmatrix} \boldsymbol{V} & \boldsymbol{B}\boldsymbol{C} \end{pmatrix}\right) \tag{14}$$

where $\boldsymbol{I}_{2K}$ is the identity matrix with dimension $2K$. As Eq. 14 requires a matrix-multiplication between $(2K) \times M$ matrices and the determinant of $(2K) \times (2K)$ matrices, this allows us to transform a $O(M^3)$ operation into an $O(MK^2 + K^3)$ one.

Having established that the normalization term in the likelihood can be computed in $O(MK^2 + K^3)$ time, we proceed with characterizing the complexity of the other terms in the likelihood. The first term in Eq. 2 consists of determinants of size $|Y_i|$. Assuming that these never exceed size K', each can be computed in at most $O(K'^3)$ time. The regularization term is a simple sum of norms that can be computed in $O(MK)$ time. Therefore, the full regularized log-likelihood can be computed in $O(MK^2 + K^3 + nK'^3)$ time.

To prove that the gradient of the log-likelihood can be computed in time linear in M, we begin by showing that the logarithm of DPP normalization term can be factorized as follows:

$$Z = \log\det(\boldsymbol{I} + \boldsymbol{L}) \tag{15}$$

$$= \log\det\left(\boldsymbol{I}_{2K} + \begin{pmatrix}\boldsymbol{V}^\top \\ \boldsymbol{B}^\top\end{pmatrix}(\boldsymbol{V} \quad \boldsymbol{B})\begin{pmatrix}\boldsymbol{I}_K & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}\end{pmatrix}\right) \tag{16}$$

$$= \log\det\left(\begin{pmatrix}\boldsymbol{I}_K & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}^{-1}\end{pmatrix} + \begin{pmatrix}\boldsymbol{V}^\top \\ \boldsymbol{B}^\top\end{pmatrix}(\boldsymbol{V} \quad \boldsymbol{B})\right) + \log\det\begin{pmatrix}\boldsymbol{I}_K & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}\end{pmatrix} \tag{17}$$

$$= \log\det\begin{pmatrix}\boldsymbol{I}_K + \boldsymbol{V}^\top\boldsymbol{V} & \boldsymbol{V}^\top\boldsymbol{B} \\ \boldsymbol{B}^\top\boldsymbol{V} & \boldsymbol{C}^{-1} + \boldsymbol{B}^\top\boldsymbol{B}\end{pmatrix} + \log\det(\boldsymbol{C}) \tag{18}$$

$$= \log\det\left(\boldsymbol{I}_K + \boldsymbol{V}^\top\boldsymbol{V}\right) + \log\det\left(\boldsymbol{C}^{-1} + \boldsymbol{B}^\top(\boldsymbol{I} - \boldsymbol{V}(\boldsymbol{I}_K + \boldsymbol{V}^\top\boldsymbol{V})^{-1}\boldsymbol{V}^\top)\boldsymbol{B}\right) + \log\det(\boldsymbol{C}) \tag{19}$$

where Eq. 17 follows from the determinant commutativity (i.e., $\det(\boldsymbol{AB}) = \det(\boldsymbol{A})\det(\boldsymbol{B})$) and Eq. 18 and Eq. 19 come from the Schur's determinant identity[3]. For simplicity, we write $\boldsymbol{X} = \boldsymbol{I} - \boldsymbol{V}(\boldsymbol{I}_K + \boldsymbol{V}^\top\boldsymbol{V})^{-1}\boldsymbol{V}^\top$ and $(\boldsymbol{C}^{-1})^\top = \boldsymbol{C}^{-\top}$ and note that $\boldsymbol{X}$ depends only on $\boldsymbol{V}$.

The gradient of Z has three parts: $\nabla Z = (\nabla_{\boldsymbol{V}} Z, \nabla_{\boldsymbol{B}} Z, \nabla_{\boldsymbol{C}} Z)$ where each can be computed as

$$\nabla_{\boldsymbol{V}} Z = \nabla_{\boldsymbol{V}}\log\det(\boldsymbol{I}_K + \boldsymbol{V}^\top\boldsymbol{V}) + \nabla_{\boldsymbol{V}}\log\det(\boldsymbol{C}^{-1} + \boldsymbol{B}^\top\boldsymbol{X}\boldsymbol{B}) \tag{20}$$

$$= 2\boldsymbol{V}(\boldsymbol{I}_K + \boldsymbol{V}^\top\boldsymbol{V})^{-1} - \boldsymbol{X}\boldsymbol{B}((\boldsymbol{C}^{-1} + \boldsymbol{B}^\top\boldsymbol{X}\boldsymbol{B})^{-1} + (\boldsymbol{C}^{-\top} + \boldsymbol{B}^\top\boldsymbol{X}\boldsymbol{B})^{-1})\boldsymbol{B}^\top\boldsymbol{X}\boldsymbol{V} \tag{21}$$

$$\nabla_{\boldsymbol{B}} Z = \nabla_{\boldsymbol{B}}\log\det(\boldsymbol{C}^{-1} + \boldsymbol{B}^\top\boldsymbol{X}\boldsymbol{B}) \tag{22}$$

$$= \boldsymbol{X}\boldsymbol{B}\left((\boldsymbol{C}^{-1} + \boldsymbol{B}^\top\boldsymbol{X}\boldsymbol{B})^{-1} + (\boldsymbol{C}^{-\top} + \boldsymbol{B}^\top\boldsymbol{X}\boldsymbol{B})^{-1}\right) \tag{23}$$

$$\nabla_{\boldsymbol{C}} Z = \nabla_{\boldsymbol{C}}\log\det(\boldsymbol{C}) + \nabla_{\boldsymbol{C}}\log\det(\boldsymbol{C}^{-1} + \boldsymbol{X}) \tag{24}$$

$$= \boldsymbol{C}^{-\top} - \boldsymbol{C}^{-\top}(\boldsymbol{C}^{-1} + \boldsymbol{B}^\top\boldsymbol{X}\boldsymbol{B})^{-\top}\boldsymbol{C}^{-\top} \tag{25}$$

Observe that $\boldsymbol{X}$ combines a $M \times M$ identity matrix with $M \times K$ matrices, hence multiplying it with a $M \times K$ matrix (e.g., $\boldsymbol{XV}$ or $\boldsymbol{XB}$) can be computed in $O(MK^2)$ time. Since each of the remaining matrix inverses in Eq. 21, Eq. 23, and Eq. 25 involve a $K \times K$ matrix inverse, with a cost of $O(K^3)$ operations, we have a net computational cost of $O(MK^2 + K^3)$ for computing $\nabla\log\det(\boldsymbol{I} + \boldsymbol{L})$.

The gradient of the first term in Eq. 2 involves computing gradients of determinants of size at most K', which results in size K' matrix inverses, since for a matrix $\boldsymbol{A}$, $\frac{\partial}{\partial A_{ij}}(\log\det(\boldsymbol{A})) = (\boldsymbol{A}^{-1})^\top_{ij}$. Each of these inverses can be computed in $O(K'^3)$ time. The gradient of the simple sum-of-norms regularization term can be computed in $O(MK)$ time. Therefore, combining these results with the results above for the complexity of the gradient of the normalization term, we have the following overall complexity of the gradient for the full log-likelihood: $O(MK^2 + K^3 + nK'^3)$. □

G.3 Proof of Lemma 2

Lemma 2. *Given $\boldsymbol{B} \in \mathbb{R}^{M\times K}$, $\boldsymbol{C} \in \mathbb{R}^{K\times K}$, and $Y = \{a_1, \ldots, a_k\} \subseteq [\![M]\!]$, let $\boldsymbol{b}_i \in \mathbb{R}^{1\times K}$ be the i-th row in $\boldsymbol{B}$ and $\boldsymbol{B}_Y \in \mathbb{R}^{|Y|\times K}$ be a matrix containing rows in $\boldsymbol{B}$ indexed by Y. Then, it holds that*

$$\boldsymbol{B}_Y^\top(\boldsymbol{B}_Y\boldsymbol{C}\boldsymbol{B}_Y^\top)^{-1}\boldsymbol{B}_Y = \sum_{j=1}^{k}\boldsymbol{p}_j^\top\boldsymbol{q}_j, \tag{6}$$

where row vectors $\boldsymbol{p}_j, \boldsymbol{q}_j \in \mathbb{R}^{1\times K}$ for $j = 1, \ldots, k$ satisfy $\boldsymbol{p}_1 = \boldsymbol{b}_{a_1}/(\boldsymbol{b}_{a_1}\boldsymbol{C}\boldsymbol{b}_{a_1}^\top)$, $\boldsymbol{q}_1 = \boldsymbol{b}_{a_1}$, and

$$\boldsymbol{p}_{j+1} = \frac{\boldsymbol{b}_{a_j} - \boldsymbol{b}_{a_j}\boldsymbol{C}^\top\sum_{i=1}^{j}\boldsymbol{q}_i^\top\boldsymbol{p}_i}{\boldsymbol{b}_{a_j}\boldsymbol{C}(\boldsymbol{b}_{a_j} - \boldsymbol{b}_{a_j}\boldsymbol{C}^\top\sum_{i=1}^{j}\boldsymbol{q}_i^\top\boldsymbol{p}_i)^\top}, \qquad \boldsymbol{q}_{j+1} = \boldsymbol{b}_{a_j} - \boldsymbol{b}_{a_j}\boldsymbol{C}\sum_{i=1}^{j}\boldsymbol{p}_i^\top\boldsymbol{q}_i. \tag{7}$$

[3] $\det\begin{pmatrix}\boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D}\end{pmatrix} = \det(\boldsymbol{A})\det(\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})$.

Proof. We prove by induction on k. When $k = 1$, the result is trivial because

$$\boldsymbol{B}_Y^\top (\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top)^{-1} \boldsymbol{B}_Y = \boldsymbol{b}_{a_1}^\top (\boldsymbol{b}_{a_1} \boldsymbol{C} \boldsymbol{b}_{a_1}^\top)^{-1} \boldsymbol{b}_{a_1} = \boldsymbol{p}_1^\top \boldsymbol{q}_1. \tag{26}$$

Now we assume that the statement holds for $k-1$. Let $Y := \{a_1, \dots, a_{k-1}\}$ and $a := a_k$. From the inductive hypothesis, it holds

$$\boldsymbol{B}_Y^\top (\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top)^{-1} \boldsymbol{B}_Y = \sum_{j=1}^{k-1} \boldsymbol{p}_j^\top \boldsymbol{q}_j. \tag{27}$$

Now we write

$$\boldsymbol{B}_{Y\cup\{a\}}^\top \left(\boldsymbol{B}_{Y\cup\{a\}} \boldsymbol{C} \boldsymbol{B}_{Y\cup\{a\}}^\top \right)^{-1} \boldsymbol{B}_{Y\cup\{a\}} \tag{28}$$

$$= \boldsymbol{B}_{Y\cup\{a\}}^\top \left(\begin{pmatrix} \boldsymbol{B}_Y \\ \boldsymbol{b}_a \end{pmatrix} \boldsymbol{C} \begin{pmatrix} \boldsymbol{B}_Y^\top & \boldsymbol{b}_a^\top \end{pmatrix} \right)^{-1} \boldsymbol{B}_{Y\cup\{a\}} \tag{29}$$

$$= \begin{pmatrix} \boldsymbol{B}_Y^\top & \boldsymbol{b}_a^\top \end{pmatrix} \begin{pmatrix} \boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top & \boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{b}_a^\top \\ \boldsymbol{b}_a \boldsymbol{C} \boldsymbol{B}_Y^\top & \boldsymbol{b}_a \boldsymbol{C} \boldsymbol{b}_a^\top \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{B}_Y \\ \boldsymbol{b}_a \end{pmatrix}. \tag{30}$$

To handle the inverse matrix we employ the Schur complement, which yields

$$\begin{pmatrix} \boldsymbol{X} & \boldsymbol{y} \\ \boldsymbol{z} & w \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{X}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{pmatrix} + \frac{1}{(w - \boldsymbol{z} X^{-1} \boldsymbol{y})^{-1}} \begin{pmatrix} \boldsymbol{X}^{-1} \boldsymbol{y} \boldsymbol{z} \boldsymbol{X}^{-1} & -\boldsymbol{X}^{-1} \boldsymbol{y} \\ -\boldsymbol{z} \boldsymbol{X}^{-1} & 1 \end{pmatrix} \tag{31}$$

for any non-singular square matrix $\boldsymbol{X} \in \mathbb{R}^{k \times k}$, column vector $\boldsymbol{y} \in \mathbb{R}^k$ and row vector $\boldsymbol{z} \in \mathbb{R}^{1 \times k}$, unless $(w - \boldsymbol{z} X^{-1} \boldsymbol{y})^{-1} = 0$. Applying this, we have

$$\begin{aligned} \begin{pmatrix} \boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top & \boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{b}_a^\top \\ \boldsymbol{b}_a \boldsymbol{C} \boldsymbol{B}_Y^\top & \boldsymbol{b}_a \boldsymbol{C} \boldsymbol{b}_a^\top \end{pmatrix}^{-1} = \begin{pmatrix} (\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top)^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{pmatrix} + \frac{1}{\boldsymbol{b}_a \boldsymbol{C} \boldsymbol{b}_a^\top - \boldsymbol{b}_a \boldsymbol{C} \boldsymbol{B}_Y^\top (\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top)^{-1} \boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{b}_a^\top} \\ \begin{pmatrix} (\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top)^{-1} \boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{b}_a^\top \boldsymbol{b}_a \boldsymbol{C} \boldsymbol{B}_Y^\top (\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top)^{-1} & -(\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top)^{-1} \boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{b}_a^\top \\ -\boldsymbol{b}_a \boldsymbol{C} \boldsymbol{B}_Y^\top (\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top)^{-1} & 1. \end{pmatrix} \end{aligned} \tag{32}$$

Substituting Eq. 32 into Eq. 30, we obtain

$$\boldsymbol{B}_{Y\cup\{a\}}^\top \left(\boldsymbol{B}_{Y\cup\{a\}} \boldsymbol{C} \boldsymbol{B}_{Y\cup\{a\}}^\top \right)^{-1} \boldsymbol{B}_{Y\cup\{a\}} \tag{33}$$

$$= \boldsymbol{B}_Y^\top \left(\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top \right)^{-1} \boldsymbol{B}_Y + \frac{\left(\boldsymbol{b}_a^\top - \boldsymbol{B}_Y^\top (\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top)^{-1} \boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{b}_a^\top \right) \left(\boldsymbol{b}_a - \boldsymbol{b}_a \boldsymbol{C} \boldsymbol{B}_Y^\top (\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top)^{-1} \boldsymbol{B}_Y \right)}{\boldsymbol{b}_a \boldsymbol{C} \left(\boldsymbol{b}_a^\top - \boldsymbol{B}_Y^\top (\boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{B}_Y^\top)^{-1} \boldsymbol{B}_Y \boldsymbol{C} \boldsymbol{b}_a^\top \right)} \tag{34}$$

$$= \sum_{j=1}^{k-1} \boldsymbol{p}_j^\top \boldsymbol{q}_j + \frac{\left(\boldsymbol{b}_a^\top - \sum_{j=1}^{k-1} \boldsymbol{p}_j^\top \boldsymbol{q}_j \boldsymbol{C} \boldsymbol{b}_a^\top \right) \left(\boldsymbol{b}_a - \boldsymbol{b}_a \boldsymbol{C} \sum_{j=1}^{k-1} \boldsymbol{p}_j^\top \boldsymbol{q}_j \right)}{\boldsymbol{b}_a \boldsymbol{C} \left(\boldsymbol{b}_a^\top - \sum_{j=1}^{k-1} \boldsymbol{p}_j^\top \boldsymbol{q}_j \boldsymbol{C} \boldsymbol{b}_a^\top \right)} \tag{35}$$

$$= \sum_{j=1}^{k-1} \boldsymbol{p}_j^\top \boldsymbol{q}_j + \boldsymbol{p}_k^\top \boldsymbol{q}_k \tag{36}$$

where the third line holds from the inductive hypothesis Eq. 27 and the last line holds from the definition of $\boldsymbol{p}_k, \boldsymbol{q}_k \in \mathbb{R}^{1 \times K}$. □

G.4 Proof of Theorem 2

Theorem 2. *Consider a nonsymmetric low-rank DPP $\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^\top + \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^\top$, where $\boldsymbol{V}, \boldsymbol{B}$ are of rank K, and $\boldsymbol{C} \in \mathbb{R}^{K \times K}$. Given a cardinality budget k, let $\sigma_{\min}$ and $\sigma_{\max}$ denote the smallest and largest singular values of $\boldsymbol{L}_Y$ for all $Y \subseteq [\![M]\!]$ and $|Y| \leq 2k$. Assume that $\sigma_{\min} > 1$. Then,*

$$\log\det(\boldsymbol{L}_{Y^G}) \geq \frac{4(1 - e^{-1/4})}{2(\log \sigma_{\max} / \log \sigma_{\min}) - 1} \log\det(\boldsymbol{L}_{Y^*}) \tag{10}$$

where Y^G is the output of Algorithm 1 and Y^* is the optimal solution of MAP inference in Eq. 4.

Proof. The proof of Theorem 2 relies on an approximation guarantee of nonsubmodular greedy maximization (Bian et al., 2017, Theorem 1). We introduce their result in below.

Theorem 3 ((Bian et al., 2017, Theorem 1)). *Consider a set function f defined on all subsets of $\{1,\dots,M\} = [\![M]\!]$ is monotone nondecreasing and nonnegative, i.e., $0 \leq f(Y) \leq f(X)$ for $\forall Y \subseteq X \subseteq [\![M]\!]$. Given a cardinality budget $k \geq 1$, let Y^* be the optimal solution of $\max_{|Y|=k} f(Y)$ and $Y^0 = \emptyset$, $Y^t := \{a_1,\dots,a_t\}, t = 1,\dots,k$ be the successive chosen by the greedy algorithm with budget k. Denote γ be the largest scalar such that*

$$\sum_{i \in X \setminus Y^t} (f(Y^t \cup \{i\}) - f(Y^t)) \geq \gamma (f(X \cup Y^t) - f(Y^t)), \tag{37}$$

for $\forall X \subseteq [\![M]\!], |X| = k$ and $t = 0,\dots,k-1$, and α be the smallest scalar such that

$$f(Y^{t-1} \cup \{i\} \cup X) - f(Y^{t-1} \cup X) \geq (1-\alpha)\left(f(Y^{t-1} \cup \{i\}) - f(Y^{t-1})\right). \tag{38}$$

for $\forall X \subseteq [\![M]\!], |X| = k$ and $i \in Y^{k-1} \setminus X$. Then, it holds that

$$f(Y^k) \geq \frac{1}{\alpha}\left(1 - e^{-\alpha\gamma}\right) f(Y^*). \tag{39}$$

In order to apply this result for MAP inference of NDPPs, the objective should be monotone nondecreasing and nonnegative. We first show that $\sigma_{\min} > 1$ is a sufficient condition for both monotonicity and nonnegativity.

Lemma 3. *Given a P_0 matrix $\boldsymbol{L} \in \mathbb{R}^{M \times M}$ and the budget $k \geq 0$, a set function $f(Y) = \log\det(\boldsymbol{L}_Y)$ defined on $Y \subseteq [\![M]\!]$ is monotone nondecreasing and nonnegative for $|Y| \leq k$ when $\sigma_{\min} > 1$.*

The proof of Lemma 3 is provided in Appendix G.6. Next, we aim to find proper bounds on α and γ. To resolve this, we provide the following upper and lower bounds of the marginal gain for $f(Y) = \log\det(\boldsymbol{L}_Y)$.

Lemma 4. *Let $f(Y) = \log\det(\boldsymbol{L}_Y)$ and assume that $\sigma_{\min} > 1$. Then, for $Y \subseteq [\![M]\!], |Y| < 2k$ and $i \notin Y$, it holds that*

$$f(Y \cup \{i\}) - f(Y) \geq \log \sigma_{\min}, \tag{40}$$

$$f(Y \cup \{i\}) - f(Y) \leq 2\log \sigma_{\max} - \log \sigma_{\min} \tag{41}$$

where $\sigma_{\min}$ and $\sigma_{\max}$ are the smallest and largest singular values of $\boldsymbol{L}_Y$ for all $Y \subseteq [\![M]\!], |Y| \leq 2k$.

The proof of Lemma 4 is provided in Appendix G.7. To bound γ, we consider $X \subseteq [\![M]\!], |X| = k$ and denote $X \setminus Y^t = \{x_1,\dots,x_r\} \neq \emptyset$. Then,

$$\sum_{i \in X \setminus Y^t} (f(Y^t \cup \{i\}) - f(Y)) = \sum_{j=1}^{r} f(Y^t \cup \{x_r\}) - f(Y^t) \geq r \log \sigma_{\min} \tag{42}$$

where the last inequality comes from Eq. 40. Similarly, we get

$$f(X \cup Y^t) - f(Y^t) = \sum_{j=1}^{r} f(\{x_1,\dots,x_j\} \cup Y^t) - f(\{x_1,\dots,x_{j-1}\} \cup Y^t) \tag{43}$$

$$\leq r(2\log \sigma_{\max} - \log \sigma_{\min}) \tag{44}$$

where the last inequality comes from Eq. 41. Combining Eq. 42 to Eq. 44, we obtain that

$$\frac{\sum_{i \in X \setminus Y^t} f(Y^t \cup \{i\}) - f(Y^t)}{f(X \cup Y^t) - f(Y^t)} \geq \frac{\log \sigma_{\min}}{2\log \sigma_{\max} - \log \sigma_{\min}} \tag{45}$$

which allows us to choose $\gamma = \left(2\frac{\log \sigma_{\max}}{\log \sigma_{\min}} - 1\right)^{-1}$.

To bound α, we similarly use Lemma 4 to obtain

$$\frac{f(X \cup Y^{t-1} \cup \{i\}) - f(X \cup Y^{t-1})}{f(Y^{t-1} \cup \{i\}) - f(Y^{t-1})} \geq \frac{\log \sigma_{\min}}{2\log \sigma_{\max} - \log \sigma_{\min}} \tag{46}$$

and we can choose $\alpha = 1 - \frac{\log \sigma_{\min}}{2\log \sigma_{\max} - \log \sigma_{\min}} = \frac{2(\log \sigma_{\max} - \log \sigma_{\min})}{2\log \sigma_{\max} - \log \sigma_{\min}}$.

Now let $\kappa = \frac{\log \sigma_{\max}}{\log \sigma_{\min}}$. Then $\gamma = \frac{1}{2\kappa - 1}$ and $\alpha = \frac{2(\kappa-1)}{2\kappa-1}$. Putting γ and α into Eq. 39, we have

$$\frac{1}{\alpha}(1 - e^{-\alpha\gamma}) \geq \frac{2\kappa - 1}{2(\kappa - 1)} \left(1 - e^{-\frac{2(\kappa-1)}{(2\kappa-1)^2}}\right) \tag{47}$$

$$\geq \frac{2\kappa - 1}{2(\kappa - 1)} \, 4\exp(-1/4) \frac{2(\kappa - 1)}{(2\kappa - 1)^2} \tag{48}$$

$$= \frac{4\exp(-1/4)}{2\kappa - 1} \tag{49}$$

where the second inequality holds from the fact that $\max_{\kappa \geq 1} \frac{2(\kappa-1)}{(2\kappa-1)^2} = \frac{1}{4}$ and $1 - e^{-x} \geq 4\exp(-1/4)x$ for $x \in [0, 1/4]$. □

G.5 Proof of Corollary 1

Corollary 1. *Consider a nonsymmetric low-rank DPP $\boldsymbol{L} = \boldsymbol{V}\boldsymbol{V}^\top + \boldsymbol{B}\boldsymbol{C}\boldsymbol{B}^\top$, where $\boldsymbol{V}, \boldsymbol{B}$ are of rank K, and $\boldsymbol{C} \in \mathbb{R}^{K \times K}$. Given a cardinality budget k, let $\sigma_{\min}$ and $\sigma_{\max}$ denote the smallest and largest singular values of $\boldsymbol{L}_Y$ for all $Y \subseteq [\![M]\!]$ and $|Y| \leq 2k$. Let $\kappa := \sigma_{\max}/\sigma_{\min}$. Then,*

$$\log\det(\boldsymbol{L}_{Y^G}) \geq \frac{4(1 - e^{-1/4})}{2\log\kappa + 1} \log\det(\boldsymbol{L}_{Y^*}) - \left(1 - \frac{4(1 - e^{-1/4})}{2\log\kappa + 1}\right) k \, (1 - \log\sigma_{\min}) \tag{12}$$

where Y^G is the output of Algorithm 1 and Y^* is the optimal solution of MAP inference in Eq. 4.

Proof. Now consider $\boldsymbol{L}' = (\frac{e}{\sigma_{\min}})\boldsymbol{L}$ where e is the exponential constant. Then, $\sigma'_{\min} = \sigma_{\min}(\frac{e}{\sigma_{\min}}) = e$ and $\sigma'_{\max} = \sigma_{\max}(\frac{e}{\sigma_{\min}})$. Using the fact that $\log\det(\boldsymbol{L}'_Y) = \log\det(\boldsymbol{L}_Y) - |Y|\log\sigma_{\min}$, we obtain the result. □

G.6 Proof of Lemma 3

Before stating the proof, we introduce interlacing properties of singular values.

Theorem 4 (Interlacing Inequality for Singular Values, (Thompson, 1972, Theorem 1))**.** *Consider a real matrix $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(M,N)}$ and its supmatrix $\boldsymbol{B} \in \mathbb{R}^{P \times Q}$ with singular values $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_{\min(P,Q)}$. Then, the singular values have the following interlacing properties:*

$$\sigma_i \geq \beta_i, \qquad \text{for } i = 1, \ldots, \min(P, Q), \tag{50}$$

$$\beta_i \geq \sigma_{i+M-P+N-Q}, \qquad \text{for } i = 1, \ldots, \min(P + Q - M, P + Q - N). \tag{51}$$

Note that when $M = N$ and $P = Q = N - 1$, it holds that $\beta_i \geq \sigma_{i+2}$ for $i = 1, \ldots, N - 2$.

We are now ready to prove Lemma 3.

Lemma 3. *Given a P_0 matrix $\boldsymbol{L} \in \mathbb{R}^{M \times M}$ and the budget $k \geq 0$, a set function $f(Y) = \log\det(\boldsymbol{L}_Y)$ defined on $Y \subseteq [\![M]\!]$ is monotone nondecreasing and nonnegative for $|Y| \leq k$ when $\sigma_{\min} > 1$.*

Proof. Since $\boldsymbol{L} \in P_0$, its all principal submatrices are also in P_0. By the definition of a P_0 matrix, it holds that

$$|\det(\boldsymbol{L}_Y)| = \det(\boldsymbol{L}_Y) = \prod_i \sigma_i(\boldsymbol{L}_Y) \tag{52}$$

where $\sigma_i(\boldsymbol{L}_Y)$ for $i \in [\![|Y|]\!]$ are singular values of $\boldsymbol{L}_Y$. Then, $F(Y) = \sum_i \log(\sigma_i(\boldsymbol{L}_Y))$ is nonnegative for all Y such that $|Y| \leq K$ due to $\sigma_i(\boldsymbol{L}_Y) \geq \sigma_{\min} > 1$. Similarly, we have $F(Y \cup \{a\}) - F(Y) = \sum_{i=1}^{|Y|+1} \log\sigma_i(\boldsymbol{L}_{Y\cup\{a\}}) - \sum_{i=1}^{|Y|} \log\sigma_i(\boldsymbol{L}_Y) \geq \log\sigma_{\min} > 0$ from Eq. 50. □

G.7 PROOF OF LEMMA 4

Lemma 4. *Let $f(Y) = \log\det(\boldsymbol{L}_Y)$ and assume that $\sigma_{\min} > 1$. Then, for $Y \subseteq [\![M]\!], |Y| < 2k$ and $i \notin Y$, it holds that*

$$f(Y \cup \{i\}) - f(Y) \geq \log \sigma_{\min}, \tag{40}$$
$$f(Y \cup \{i\}) - f(Y) \leq 2\log \sigma_{\max} - \log \sigma_{\min} \tag{41}$$

where $\sigma_{\min}$ and $\sigma_{\max}$ are the smallest and largest singular values of $\boldsymbol{L}_Y$ for all $Y \subseteq [\![M]\!], |Y| \leq 2k$.

Proof. For a P_0 matrix, we remark that its determinant is equivalent to the product of all singular values. For $Y \subseteq [\![M]\!]$ and $i \notin Y$, from the interlacing inequality of Eq. 50 we have that

$$F(Y \cup \{i\}) - F(Y) = \sum_{j=1}^{|Y|+1} \log \sigma'_j - \sum_{j=1}^{|Y|} \log \sigma_j \geq \log \sigma'_{|Y|+1} \geq \log \sigma_{\min} \tag{53}$$

where σ'_j and σ_j are the j-th largest singular value of $\boldsymbol{L}_{Y\cup\{i\}}$ and $\boldsymbol{L}_Y$, respectively. Similarly, using Eq. 51, we get

$$F(Y \cup \{i\}) - F(Y) \leq \log(\sigma'_1\sigma'_2) - \log \sigma_{|Y|} \leq 2\log \sigma_{\max} - \log \sigma_{\min}. \tag{54}$$

□