


# New Bounds for Private Graph Optimization Problems via Synthetic Graphs

Anders Aamand ✉ 

BARC, University of Copenhagen, Denmark

Rasmus Pagh ✉ 

BARC, University of Copenhagen, Denmark

Lukas Retschmeier ✉ 

BARC, University of Copenhagen, Denmark

---

## Abstract

We consider the graph optimization problems of privately releasing the edges of minimum-weight *spanning trees*, *perfect matchings*, and *shortest paths* as introduced by Sealfon [PODS 2016]. Given a public graph topology  $G = (V, E)$ , together with private edge weights  $\mathbf{W} \in \mathbb{R}^E$ , we want to publish an approximate solution to those problems under *edge-weight* differential privacy. We show new asymptotically tight additive error bounds for all three problems under the  $\ell_1$  neighboring relationship. Interestingly, the mechanisms achieving these bounds are the simplest one can imagine: Construct a private synthetic graph by adding noise to the edge weights and then run a non-private graph algorithm.

Concretely, we improve Sealfon’s lower bound technique for spanning trees and matchings, increasing the bound from  $\Omega(n/\varepsilon)$  to  $\Omega((n/\varepsilon) \min(\log(1/\delta), \log n))$  which matches the known upper bound in the regime  $\delta = n^{-\Omega(1)}$ . For shortest paths, we show that adding Laplace noise to weights and capping the noisy edge weight at zero (so that the new weight is non-negative), yields a synthetic graph that, with probability  $1 - \exp(-\Omega(n))$ , preserves all  $s$ - $t$  shortest path distances within additive error  $O(n/\varepsilon)$ . This improves on Sealfon’s  $O((n \log n)/\varepsilon)$  upper bound and matches his lower bound within a constant factor.

Lastly, we present preliminary work on these problems under the  $\ell_\infty$  neighboring relationship. For instance, we transfer the packing lower bound proposed by Hladík and Tětek [FORC 2025] to the setting of perfect matching. While our analysis offers tighter bounds in some settings, several questions remain open both in terms of approximation guarantees and run times.

**2012 ACM Subject Classification** Security and Privacy → Privacy-preserving protocols

**Keywords and phrases** differential privacy, graph algorithms, lower bounds

**Funding** Aamand, Pagh and Retschmeier carried out this work at Basic Algorithms Research Copenhagen (BARC), which was supported by the VILLUM Foundation grant 54451. Additionally, *Providentia*, a Data Science Distinguished Investigator grant from the Novo Nordisk Fonden, supported Pagh and Retschmeier.

## 1 Introduction

Releasing the edges of *shortest path trees* (SPT), *minimum spanning trees* (MST), and *minimum-weight perfect matching* (MPM) belong to the most fundamental optimization problems on graphs. With the additional constraints of differential privacy they have been studied in the contexts of *route planning* [26], *synthetic data generation* [20] or *clustering* [24, 17, 15, 1].

In this work, we consider *edge-weight differential privacy* proposed by Sealfon [26], where the graph topology with  $n$  vertices and  $m$  edges itself  $G = (V, E)$  is assumed to be publicly known, but the weight vector  $\mathbf{W}$  should be kept private. Building on previous works, we consider both the  $\ell_1$  and  $\ell_\infty$  neighborhood relationships, where neighboring graphs  $G = (V, E, \mathbf{W})$  and  $G' = (V, E, \mathbf{W}')$  can differ by either  $\|\mathbf{W} - \mathbf{W}'\|_1 \leq 1$  or  $\|\mathbf{W} - \mathbf{W}'\|_\infty \leq 1$ .

While an  $\ell_1$  bound focuses more on the local impact, an  $\ell_\infty$  bound occurs naturally if each of the weights is a conjoint aggregate of some underlying private dataset as it is, for example, the case for Chow-Liu trees [3].

We distinguish between algorithms that are based on *input-perturbation* [26] where noise is added directly to the instance to create a synthetic private instance, and *in-place* algorithms that add noise throughout the individual steps of some algorithmic process [24, e.g. PAMST]. Input-perturbation has a number of desirable properties over *in-place* algorithms. First, they are often quite simple (e.g., [26] simply adds Laplace noise to each edge weight of a graph). Second, for in-place algorithms, the privacy/utility trade-off worsens if we have to answer multiple queries, but multiple queries is not a concern for input-perturbation algorithms since the synthetic instance is private and any subsequent computations on it does not violate privacy. Finally, it is often desirable to represent data as a synthetic private data set (e.g., a synthetic graph) on which one can run many different algorithms.

It is therefore interesting to investigate whether input perturbation algorithms come at a cost of a worse privacy/utility tradeoff compared to in-place algorithms. Recent work has demonstrated that this is not always the case [11, 22]. In particular for differentially private MST under the  $\ell_\infty$  neighboring relationship, an algorithm that adds noise to each edge weight and runs a non-private MST algorithm is in fact optimal for privately releasing an approximate MST [22].

Optimal bounds for other graph optimization problems are less understood. Besides releasing a minimum spanning tree, Sealfon [26] also considers privately releasing single source shortest path trees and a minimum-weight perfect matching under the  $\ell_1$  neighboring relation, and provides a lower bound showing that an error of  $\Omega(n)$  is necessary for all of them together with the mentioned upper bound. This leaves a multiplicative gap between upper and lower bounds of  $\mathcal{O}(\log n)$  for  $\varepsilon$ -DP and  $\mathcal{O}(\sqrt{\log(n) \log(1/\delta)})$  for  $(\varepsilon, \delta)$ -DP.

In this work we prove new tight bounds for these problems where it turns out that the bounds can be achieved by creating a private synthetic graph. Our work is thus guided by the following question:

**Question:** How much additive error is needed for privately releasing *minimum spanning trees*, *perfect matchings* and *shortest paths*? And in which cases is adding noise to the input weights and running a non-private graph algorithm optimal?

## 1.1 Our contributions

We make progress answering this question by showing that under the  $\ell_1$  neighboring relationship, releasing a private synthetic graph (either the same or a small variation of the one from [26]) gives optimal error guarantees up to constant factors. Our contributions include the following new tight upper and lower bounds for all three aforementioned graph optimization problems under the  $\ell_1$  neighboring relationship. A summary of all our new results together with known bounds appears in Tables 1 and 2.

**Improved lower bound technique.** We improve a lower bound technique for  $(\varepsilon, \delta)$ -DP under the  $\ell_1$  neighborhood due to Sealfon. Instead of encoding a random binary vector into a sparse graph, we encode a random  $\mathbf{X} \in_R [n]^d$  into a dense graph, which lifts the existing lower bound by a logarithmic factor. This gives tight bounds of  $\Theta(\frac{n}{\varepsilon} \log n)$  for sufficiently small  $\delta = n^{-\Omega(1)}$ . In particular, we are proving the following theorem, which with slight modifications also holds for *minimum-weight perfect matchings*:

► **Theorem (Lower bound (informal)).** *Let  $G$  be some graph topology with  $n$  vertices and let  $\varepsilon > 0$  and  $\delta = n^{-\Omega(1)}$  be two constants. Now assume  $\mathcal{B}$  to be some  $(\varepsilon, \delta)$ -differentially private*

Problem	Privacy	Error Upper Bound	Error Lower Bound
MST	$\varepsilon$ -DP	$\mathcal{O}(n \cdot \log n)^A$	$\Omega(n \cdot \log n)^B$
	$(\varepsilon, \delta)$ -DP	$\mathcal{O}(n \cdot \sqrt{\log(n) \log(1/\delta)})^A$	$\Omega(n \cdot \min(\log(1/\delta), \log(n)))^{\text{Theorem 5}}$
SP	$\varepsilon$ -DP	$\mathcal{O}(n)^{\text{Theorem 1}}$	$\Omega(n)^A$
	$(\varepsilon, \delta)$ -DP	$\mathcal{O}(n)^{\text{Theorem 1}}$	$\Omega(n)^A$
MWPM	$\varepsilon$ -DP	$\mathcal{O}(n \cdot \log(n))^A$	$\Omega(n \cdot \log n)^{\text{Theorem 11}}$
	$(\varepsilon, \delta)$ -DP	$\mathcal{O}(n \cdot \sqrt{\log(n) \log(1/\delta)})^A$	$\Omega(n \cdot \min(\log(1/\delta), \log(n)))^{\text{Theorem 11}}$

■ **Table 1** Landscape of private graph optimization problems under the  $\ell_1$  neighboring relation: MINIMUM SPANNING TREE (**MST**), Single-Source SHORTEST PATH TREE (**SSPT**) and MINIMUM-WEIGHT PERFECT MATCHING (**MWPM**). All bounds have a multiplicative factor  $(1/\varepsilon)$ , omitted for the sake of clarity. **References:** **A)** Sealfon [26], **B)** Hladík and Tětek [11]

mechanism respecting the  $\ell_1$  neighboring relation that given  $G$  together with some weights  $\mathbf{W}$  outputs some approximate minimum-weight spanning tree (or a minimum-weight perfect matching)  $T$  that with probability  $\geq 1/2$  satisfies  $|w(T) - w(T^*)| \leq \alpha$  where  $T^*$  is the real mst. Then

$$\alpha = \Omega\left(\frac{1}{\varepsilon} \cdot (n \cdot \min(\log(1/\delta), \log(n)))\right)$$

The idea is that by a fundamental property of privacy, we can not release a single coordinate of  $X_i$  with probability more than  $\varepsilon^e/n + \delta$ , and hence the output of any private MST-algorithm on the graph that encodes  $\mathbf{X}$  could be used to reidentify a large part of the input, contradicting privacy.

**Tight bounds for private all pairs shortest path release.** The existing upper bounds for the problems above are obtained by bounding the maximum amount of noise for any single edge and then union bounding over all edges. In the case of MST, this is in fact tight, as showed in [11] under pure DP and in this paper under approximate DP (when  $\delta = n^{-\Omega(1)}$ ).

For releasing a private graph preserving all shortest path distances, one can either bound the error of each edge or, more complicated, union bound over all paths in the graphs using a concentration bound on the sum of the Laplace variables associated with a single path. Both of these approaches lead to error  $O(n \log(n)/\varepsilon)$ . Of particular interest to us is the latter approach, where we get good error guarantees for a single path and where the additive approximation directly relates to the number  $n^{O(n)}$  of possible paths in the union bound. One can see that a union bound over  $2^{O(n)}$  sets of edges would lead to a better approximation guarantee of  $O(n/\varepsilon)$  and this is exactly our contribution: We prove a *combinatorial* claim that there exists a family  $\mathcal{F} = \{S_1, \dots, S_{4^n}\}$  where each  $S_i$  is a set of edges, such that as long as the total weight of each set  $S_i$  does not change by more than  $O(n/\varepsilon)$  after adding noise, then the shortest path distances in the synthetic graph are also preserved up to additive error  $O(n/\varepsilon)$ . By the union bound, this happens with extremely high probability  $1 - \exp(-\Omega(n))$ . Our result is as follows.

► **Theorem 1 (Upper bound).** Denote with  $d_G(s, t)$  the distance between  $s$  and  $t$  in some graph  $G$ . There exists an algorithm, that is  $\varepsilon$ -DP under the  $\ell_1$  neighborhood relation, which

on input  $G$  releases a private synthetic graph  $\tilde{G}$  such that with probability  $1 - \exp(-\Omega(n))$ , it holds for all  $s, t \in V$  that  $|d_G(s, t) - d_{\tilde{G}}(s, t)| \leq 5n/\varepsilon$ .

Our algorithm has three benefits over the one from [26]: (1) It is slightly simpler, defining the noisy edge weight  $\tilde{w}(e) = \max(0, w(e) + \text{Lap}(1/\varepsilon))$  whereas Sealfon defines  $w(e) = \max(0, w(e) + \text{Lap}(1/\varepsilon) + C(\log n)/\varepsilon)$  for a large constant  $C$ , (2) it offers optimal error bounds, and (3) the error probability is  $\exp(-\Omega(n))$ , rather than  $1/\text{poly}(n)$ .

**Additional results.** Furthermore, we extend the argument due to Hladík and Tětek, which is based on a classic packing lower bound, to perfect matchings for  $\varepsilon$ -DP:  $\Omega((n \log n)/\varepsilon)$  for  $\ell_1$  and  $\Omega((n^2 \log n)/\varepsilon)$  for  $\ell_\infty$ . While the  $\ell_1$  result matches exactly the input perturbation approach, we provide a matching upper bound for the latter one using an (inefficient) application of the exponential mechanism.

**Organization.** Because of space restrictions, most technical details have been moved to the appendix. After introducing the necessary notation in Appendix A, we will prove the results in Appendix B, where we consider the  $\ell_1$  and  $\ell_\infty$  neighboring relationship independently. Finally, in Section 2, we will wrap up and state future research directions.

## 2 Conclusion and Open Problems

Our main contribution under  $\ell_1$  is improving Sealfon’s lower bound technique and showing that a union bound on the maximum error is not required for privately releasing *shortest path trees*, interestingly removing the separation in utility between pure and approximate DP. We have given tight bounds for nearly all the problems  $\ell_1$  (Table 1) and started to explore the  $\ell_\infty$  neighboring relation (Table 2) as well. While it seems perturbing the input is the optimal thing to do under  $\ell_1$ , we leave the question of finding tight bounds under  $\ell_\infty$  to further research.

Pagh, Retschmeier, Wu, and Zhang have shown that adding exponential noise to each edge and only releasing the output of any MST algorithm exactly matches the output distribution of running a private version of Kruskal’s algorithm. Does a similar technique that is based on perturbing the input also hold for the other problems? One could combine this with the textbook 2-approximation algorithm for releasing a maximum matching, but this adds a constant multiplicative error but matches the lower bound if the weight of the matching is roughly  $\mathcal{O}(n \log n)$ . Furthermore, the authors also used a connection to top- $k$  selection to prove a lower bound of  $\tilde{\Omega}(n^{3/2})$  for approximate DP that holds for small  $\delta$ . Can we make a similar argument for releasing a perfect matching?

Tight bounds for  $\ell_\infty$  under approximate DP are only known for privately releasing an MST. In this case, one can save a factor of  $\mathcal{O}(\sqrt{n})$ , obtaining an error of  $\tilde{\Theta}(n^{3/2})$  [22, 24, 20] by developing more sophisticated private algorithms. Is something similar possible for perfect matchings and shortest paths? The upper bounds based on the exponential mechanism presented for  $\varepsilon$ -DP are tight, but, unlike for the MST problem [11], we don’t know how to sample from this distribution efficiently. It would be of great interest to find efficient algorithms as well. Can one privatize standard approaches like Edmond’s Blossom or Dijkstra’s algorithm [7, 4]?

## References

- 1 MohammadHossein Bateni, Soheil Behnezhad, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, Raimondas Kiveris, Silvio Lattanzi, and Vahab S. Mirrokni. Affinity clustering: Hierarchical clustering at scale. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6864–6874, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/2e1b24a664f5e9c18f407b2f9c73e821-Abstract.html>.
- 2 Greg Bodwin, Chengyuan Deng, Jie Gao, Gary Hoppenworth, Jalaj Upadhyay, and Chen Wang. The Discrepancy of Shortest Paths. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:20, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-322-5. doi: 10.4230/LIPIcs.ICALP.2024.27. URL <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2024.27>.
- 3 C. K. Chow and Chao-Ming Liu. Approximating discrete probability distributions with dependence trees. volume 14, pages 462–467, 1968. URL <https://api.semanticscholar.org/CorpusID:27127853>.
- 4 Edsger W Dijkstra. A note on two problems in connexion with graphs. volume 1, pages 269–271. Springer, 1959.
- 5 Michael Dinitz, George Z. Li, Quanquan C. Liu, and Felix Zhou. Differentially private matchings, 2025. URL <https://arxiv.org/abs/2501.00926>.
- 6 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC’06, page 265–284, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3540327312. doi: 10.1007/11681878\_14. URL [https://doi.org/10.1007/11681878\\_14](https://doi.org/10.1007/11681878_14).
- 7 Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. doi: 10.4153/CJM-1965-045-4.
- 8 Chenglin Fan, Ping Li, and Xiaoyun Li. Private graph all-pairwise-shortest-path distance release with improved error rate. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 17844–17856. Curran Associates, Inc., 2022. URL [https://proceedings.neurips.cc/paper\\_files/paper/2022/file/71b17f00017da0d73823ccf7fbce2d4f-Paper-Conference.pdf](https://proceedings.neurips.cc/paper_files/paper/2022/file/71b17f00017da0d73823ccf7fbce2d4f-Paper-Conference.pdf).
- 9 Badih Ghazi, Jelani Osei Nelson, Justin Y. Chen, Pasin Manurangsi, Ravi Kumar, Shyam Narayanan, and Yinzhan Xu. Differentially private all-pairs shortest path distances: Improved algorithms and lower bounds. In *SODA 2023*, 2023.
- 10 Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate estimation of the degree distribution of private networks. In *2009 Ninth IEEE International Conference on Data Mining*, pages 169–178, 2009. doi: 10.1109/ICDM.2009.11.
- 11 Richard Hladík and Jakub Tětek. Near-universally-optimal differentially private minimum spanning trees. Stanford University, CA, USA, 2025. Symposium on Foundations of Responsible Computing (FORC). URL <https://arxiv.org/abs/2404.15035>.
- 12 Justin Hsu, Zhiyi Huang, Aaron Roth, Tim Roughgarden, and Zhiwei Steven Wu. Private matchings and allocations. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC ’14, page 21–30, New York, NY, USA, 2014. Association

- for Computing Machinery. ISBN 9781450327107. doi: 10.1145/2591796.2591826. URL <https://doi.org/10.1145/2591796.2591826>.
- 13 Svante Janson. Tail bounds for sums of geometric and exponential variables. volume 135, pages 1–6. Elsevier, 2018.
  - 14 Vojtěch Jarník. O jistém problému minimálním. (z dopisu panu o. borůvkovi) [on a certain problem of minimization]. *Práce moravské přírodovědecké společnosti*, 6:57–63, 1930. URL <https://dml.cz/handle/10338.dmlcz/500726?show=full>.
  - 15 Rajesh Jayaram, Vahab Mirrokni, Shyam Narayanan, and Peilin Zhong. Massively parallel algorithms for high-dimensional euclidean minimum spanning tree. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 3960–3996. SIAM, 2024. doi: 10.1137/1.9781611977912.139. URL <https://doi.org/10.1137/1.9781611977912.139>.
  - 16 Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *Proceedings of the 10th Theory of Cryptography Conference on Theory of Cryptography, TCC’13*, page 457–476, Berlin, Heidelberg, 2013. Springer-Verlag. ISBN 9783642365935. doi: 10.1007/978-3-642-36594-2\_26. URL [https://doi.org/10.1007/978-3-642-36594-2\\_26](https://doi.org/10.1007/978-3-642-36594-2_26).
  - 17 Chih Lai, Taras Rafa, and Dwight E. Nelson. Approximate minimum spanning tree clustering in high-dimensional space. volume 13, pages 575–597, 2009. doi: 10.3233/IDA-2009-0382. URL <https://doi.org/10.3233/IDA-2009-0382>.
  - 18 Yang Li, Michael Purcell, Thierry Rakotoarivelo, David Smith, Thilina Ranbaduge, and Kee Siong Ng. Private graph data release: A survey. volume 55, New York, NY, USA, February 2023. Association for Computing Machinery. doi: 10.1145/3569085. URL <https://doi.org/10.1145/3569085>.
  - 19 Colin McDiarmid. *On the method of bounded differences*, page 148–188. London Mathematical Society Lecture Note Series. Cambridge University Press, 1989.
  - 20 Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the nist contest: A scalable and general approach to differentially private synthetic data. volume 11, Dec. 2021. doi: 10.29012/jpc.778. URL <https://journalprivacyconfidentiality.org/index.php/jpc/article/view/778>.
  - 21 Tamara T. Mueller, Dmitrii Usynin, Johannes C. Paetzold, Daniel Rueckert, and Georgios Kaissis. Sok: Differential privacy on graph-structured data, 2022.
  - 22 Rasmus Pagh, Lukas Retschmeier, Hao Wu, and Hanwen Zhang. Optimal bounds for private minimum spanning trees via input perturbation. In *ACM SIGMOD Symposium on Principles of Database Systems, PODS ’25*, New York, NY, USA, 2024. Association for Computing Machinery. URL <https://arxiv.org/abs/2412.10130>.
  - 23 Rafael Pinot. Minimum spanning tree release under differential privacy constraints, 2018. URL <https://arxiv.org/abs/1801.06423>.
  - 24 Rafael Pinot, Anne Morvan, Florian Yger, Cedric Gouy-Pailler, and Jamal Atif. Graph-based Clustering under Differential Privacy. In *Conference on Uncertainty in Artificial Intelligence (UAI 2018)*, pages 329–338, Monterey, California, United States, August 2018. Conference on Uncertainty in Artificial Intelligence (UAI 2018). URL <https://hal.science/hal-02170699>.
  - 25 Robert Clay Prim. Shortest connection networks and some generalizations. volume 36, pages 1389–1401. Nokia Bell Labs, 1957. doi: 10.1002/j.1538-7305.1957.tb01515.x.
  - 26 Adam Sealfon. Shortest paths and distances with differential privacy. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS ’16*, page 29–41, New York, NY, USA, 2016. Association for Computing Machinery.

- ISBN 9781450341912. doi: 10.1145/2902251.2902291. URL <https://doi.org/10.1145/2902251.2902291>.
- 27 Salil Vadhan. *The Complexity of Differential Privacy*, pages 347–450. Springer International Publishing, Cham, 2017. ISBN 978-3-319-57048-8. doi: 10.1007/978-3-319-57048-8\_7. URL [https://doi.org/10.1007/978-3-319-57048-8\\_7](https://doi.org/10.1007/978-3-319-57048-8_7).



We will state all the proofs for the new results here in the appendix.

## A Preliminaries

We consider a graph  $G = (V, E, \mathbf{W})$ , where the set of  $n$  vertices  $V$  and the set of  $m$  edges  $E$  are public and the weight vector  $\mathbf{W} = (w_1, \dots, w_m) \in \mathbb{R}^m$  is private. We use  $\text{inc}_G(v)$  to refer to the set of all incident edges of a vertex  $v$ . We denote with  $\mathcal{G} = \{(V, E)\}$  the set of all unweighted graphs (*graph topologies*) and  $\mathcal{G}_w$  all weighted ones. For a subset  $S \subseteq E$  of edges, we denote the *cost* of  $S$  as  $w(S) = \sum_{e \in S} w_e$ .

We write the concatenation of two vectors  $\mathbf{W} = (w_1, \dots, w_n)$ ,  $\mathbf{W}' = (w'_1, \dots, w'_m)$  as  $\mathbf{W} \oplus \mathbf{W}' = (w_1, \dots, w_n, w'_1, \dots, w'_m)$ , and we quickly write  $\mathbf{W}_{<i} := (w_1, \dots, w_{i-1})$  and  $\mathbf{W}_{>i} := (w_{i+1}, \dots, w_d)$ . For two vectors  $\mathbf{W}$  and  $\mathbf{W}'$ , we define the *hamming distance*  $d_H(\mathbf{W}, \mathbf{W}') := \sum_{i=1}^n \mathbb{1}(w_i \neq w'_i)$  to be the number of coordinates in which they differ. In case  $S, S'$  are sets, we simply introduce it as  $d_H(S, S') := |S \setminus S'| = |S' \setminus S|$ . We use the operator  $X \in_R A$  to denote that the random variable  $X$  is drawn uniformly from the set  $A$ .

**Differential privacy.** Throughout this work, we require three different notions of neighboring datasets.

► **Definition 2 (Neighboring datasets).** *For some fixed constants  $\Delta_1$  and  $\Delta_\infty$ , we say that two vectors  $\mathbf{W} = (w_1, \dots, w_n)$  and  $\mathbf{W}' = (w'_1, \dots, w'_n)$  are neighboring*

- *under the hamming neighboring relationship ( $\mathbf{W} \sim_H \mathbf{W}'$ ), if  $d_H(\mathbf{W}, \mathbf{W}') \leq 1$ ,*
- *under the  $\ell_1$  neighboring relationship ( $\mathbf{W} \sim_1 \mathbf{W}'$ ), if  $\|\mathbf{W} - \mathbf{W}'\|_1 \leq \Delta_1$ , and*
- *under the  $\ell_\infty$  neighboring relationship ( $\mathbf{W} \sim_\infty \mathbf{W}'$ ), if  $\|\mathbf{W} - \mathbf{W}'\|_\infty \leq \Delta_\infty$ .*

For simplicity, we assume throughout the paper that  $\Delta_\infty = \Delta_1 = 1$ , but all bounds can be generalized by scaling with a factor of  $\Delta_1$  or  $\Delta_\infty$  respectively.

► **Definition 3 (Dwork, McSherry, Nissim, and Smith (2006)  $(\varepsilon, \delta)$ -private algorithm).** *Let  $\varepsilon, \delta > 0$ ,  $G = (V, E)$  be a graph, and let  $\mathcal{S}(G) \subseteq 2^E$  be the possible output space. An algorithm  $\mathcal{M}$  is called  $(\varepsilon, \delta)$ -differentially private (DP), if for every  $G = (V, E, \mathbf{W})$ ,  $G' = (V', E', \mathbf{W}')$  such that  $G \sim G'$  for some neighboring relationship, and all  $Z \in \mathcal{S}(G)$ ,*

$$\Pr[\mathcal{M}(G) = Z] \leq e^\varepsilon \cdot \Pr[\mathcal{M}(G') = Z] + \delta.$$

In case of  $\delta = 0$ , we simply say, that the  $\mathcal{M}$  is  $\varepsilon$ -DP.

Although Definition 3 is phrased in the context of privately releasing a set of edges, it applies to any randomized algorithm  $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Y}$ , where  $\mathcal{X}$  is the input space, which is associated with a symmetric relation  $\sim$  that defines neighboring inputs and  $\mathcal{Y}$  the output space.

## B Results

We will first prove new tight bounds for the  $\ell_1$  neighboring relationship. All the following upper bounds are obtained by the simplest thing one can imagine: Adding a controlled amount of noise to each of the input weight and then release this privatized weight vector. In Appendix B.4, we will focus on  $\ell_\infty$ .

**Reidentification Lemma** Before proving our new bounds, we need a small lemma that bounds the probability that any  $(\varepsilon, \delta)$ -DP mechanism can leak parts of its input. This lemma is a generalization of [26, Lemma 5.3] which only holds if  $\mathbf{X}$  is a binary vector. We show that the probability that any  $(\varepsilon, \delta)$ -DP mechanism  $\mathcal{B} : \mathcal{X}^d \rightarrow \mathcal{X}^d$  on some random input  $\mathbf{X} \in_R [n]^d$  leaks one of its true input bits  $X_i$  is at most  $e^\varepsilon/d + \delta$ .



► **Lemma 4.** *Assuming two integers  $n, d > 1$ , if some mechanism  $\mathcal{B} : \mathcal{X}^d \rightarrow \mathcal{X}^d$  is  $(\varepsilon, \delta)$ -differentially private under the hamming neighborhood relationship, then for  $\mathbf{X} \in_R \mathcal{X}^d$  drawn uniformly random, we have  $\Pr[\mathcal{B}(\mathbf{X})_i = X_i] \leq \frac{e^\varepsilon}{n} + \delta$  for each  $i \in [d]$ .*

**Proof.** Let  $n > 1$  be some integer. We index the elements in  $\mathcal{X}$  by  $\{1, \dots, n\}$  and assume that  $\mathcal{X} = [n]$ . For some arbitrarily chosen  $i \in [d]$ , assume a random vector  $\mathbf{X} = \mathbf{X}_{< i} \oplus X_i \oplus \mathbf{X}_{> i} \in_R [n]^d$ . Recall that  $\oplus$  denotes concatenation, and we use the subscript  $\mathbf{X}_{< i}$  and  $\mathbf{X}_{> i}$  to split the vector at index  $i$ . Now we can bound the probability that the  $i$ 'th coordinate of the output of  $\mathcal{B}(\mathbf{X})$  outputs part of its input  $x_i$ :

$$\Pr[\mathcal{B}(\mathbf{X})_i = X_i] = \frac{1}{n^{d-1}} \sum_{\mathbf{X}_{< i} \in [n]^{i-1}} \sum_{\mathbf{X}_{> i} \in [n]^{d-i}} \left( \Pr_{X_i \in_R [n]} [\mathcal{B}(\mathbf{X}_{< i} \oplus X_i \oplus \mathbf{X}_{> i})_i = X_i] \right) \quad (1)$$

$$\leq n^{1-d} \sum_{\mathbf{X}_{< i} \in [n]^{i-1}} \sum_{\mathbf{X}_{> i} \in [n]^{d-i}} \left( e^\varepsilon \Pr_{X_i \in_R [n]} [\mathcal{B}(\mathbf{X}_{< i} \oplus (1) \oplus \mathbf{X}_{> i})_i = X_i] + \delta \right) \quad (2)$$

$$\leq n^{1-d} \sum_{\mathbf{X}_{< i} \in [n]^{i-1}} \sum_{\mathbf{X}_{> i} \in [n]^{d-i}} \left( e^\varepsilon \frac{1}{n} + \delta \right) \quad (3)$$

$$\leq \frac{e^\varepsilon}{n} + \delta \quad (4)$$

In line 2, we use the fact that each  $X_i$  is an i.i.d. random variable and in step 3, we use the privacy guarantees of mechanism  $\mathcal{B}$  and flip to a neighboring dataset where we explicitly set  $X_i = 1$  under our hamming neighboring relationship. ◀

## B.1 Private Minimum Spanning Trees

We now focus on the problem of privately releasing a minimum(-weight) spanning tree. A spanning tree is an acyclic subset of edges that connects the graph. For a graph  $G \in \mathcal{G}$ , we denote  $\mathcal{T}(G)$  to be the set of all spanning trees. Given a public graph topology  $G = (V, E)$  together with private weights  $\mathbf{W} \in \mathbb{R}^E$ , we want to release some  $T \in \mathcal{T}(G)$ , such that  $w(T)$  is minimized and releasing  $T$  satisfies edge-weight differential privacy.

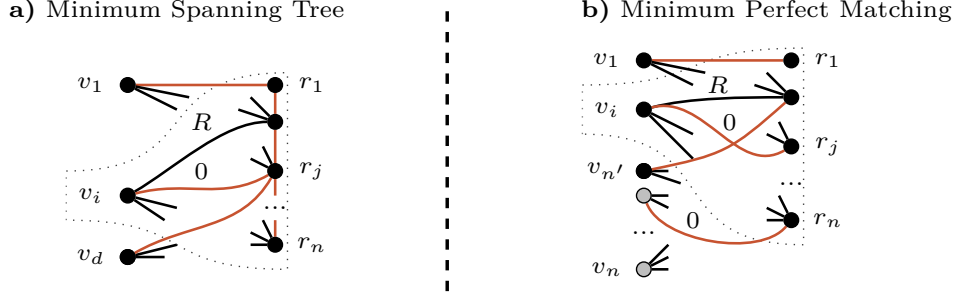
### Improving the lower bound for $(\varepsilon, \delta)$ -DP

Assuming  $(\varepsilon, \delta)$ -DP, we now show that the additive error of  $\Theta((n \log n)/\varepsilon)$  one obtains by adding Gaussian noise to each edge is asymptotically tight for small enough  $\delta \leq n^{-\Omega(1)}$ , thus closing the remaining gap [26] of a logarithmic factor.

The idea is to encode a random vector  $\mathbf{X} \in [n]^d$  into the mst of a dense graph  $G$  where releasing an overly accurate mst by a differentially private mechanism would contradict the upper bound in Lemma 4. As a by-product, we also get a lower bound for  $\varepsilon$ -DP, obtained by taking the limit of  $\delta$  towards 0. More precisely, we are going to prove the following lower bound:

► **Theorem 5 (Lower Bound MST).** *There exists a graph topology  $G = (V, E)$  and a distribution of weights  $\mathcal{D}_\omega$  on weights of  $E$  such that for any  $(\varepsilon, \delta)$ -DP protocol  $\mathcal{B}$  that outputs an approximate minimum-weight spanning tree  $T$  under the  $\ell_1$  neighboring relationship, if the weights  $\mathbf{W}$  of the graph are sampled according to  $\mathcal{D}_\omega$ , then with probability at least  $1/2$ , we have*

$$w(T) - w(T^*) \geq \Omega((n/\varepsilon) \cdot \min(\log(1/\delta), \log n))$$



■ **Figure 1** Encoding the vector  $\mathbf{X} = (X_1, \dots, X_d)$  into a minimum-weight spanning tree and a minimum-weight perfect matching of a dense graph. Each  $r_j$  represents the integer  $j$ , and we set the weight of the edge from  $v_i$  to  $r_j$  to 0, if  $X_i = j$  and  $R$  else. The corresponding edges are marked in red. **a)**: **b)**:  $n'$  is later chosen to be roughly  $\alpha n - \alpha^2 n \in \Theta(\alpha n)$ . In the case of matching, we add the gray dummy nodes that ensure that a perfect matching exists.

where  $T^*$  is the optimal mst in  $(G, \mathbf{W})$ .

We now describe how to encode a vector  $\mathbf{X} \in [n]^d$  into the mst of a dense graph. A visualization of this construction is depicted in Figure 1.

► **Definition 6 (MST Encoding). Encoding.** Let  $\text{ENC}_R : [n]^d \rightarrow \mathcal{G}_\omega$  be the encoding function that takes some vector  $\mathbf{X} \in [n]^d$  and returns a weighted graph  $G = (V, E, \mathbf{W})$ . The parameter  $R \in \mathbb{R}$  is used to control the edge weights. We construct a new connected graph  $G = (V, E, \mathbf{W})$  in the following way: First, add a path of length  $n - 1$  with vertices  $R := \{r_1, \dots, r_n\}$  where each  $r_i$  encodes some integer  $i \in [n]$  and set the weight of edge  $e = \{r_i, r_{i+1}\}$  to  $w_e = 0$  for each  $i \in [n - 1]$ . For each  $X_i$  of  $\mathbf{X}$ , add a new vertex  $v_i$  together with new edges  $e_i^1, \dots, e_i^n$  from  $v_i$  to all  $r_j \in R$ . Set the weights for each  $j \in [n]$  to  $w(e_i^j) = 0$  if  $j = X_i$  and  $w(e_i^j) = R$  otherwise.

**Decoding.** Let  $\text{DEC}_G : \mathcal{T} \rightarrow [n]^d$  be a function parameterized by a weighted graph  $G \in \mathcal{G}_\omega$  with vertices  $V = \{v_1, \dots, v_d, r_1, \dots, r_n\}$ , together with some spanning tree  $T \in \mathcal{T}(G)$ . This function decodes the original vector  $\mathbf{X} = (X_1, \dots, X_d)$ . We set for all  $v_i \in T$ :  $X_i := \min_{e \in \text{inc}_G(v_i)} \{w(e)\}$  to be the minimum incident edge of  $v_i \in T$ .

Clearly, for each  $v \in \{v_1, \dots, v_d\}$  there is at least one edge in  $\text{inc}(v_i) \in T$  and thus, each  $X_i$  of the decoded vector is defined. Given any mst algorithm  $\mathcal{A}$  and vector  $\mathbf{X} \in [n]^d$ , we shortly denote  $\text{ENCDEC}_R(\mathcal{A}, \mathbf{X}) := \text{DEC}(\mathcal{A}(\text{ENC}_R(\mathbf{X})))$ . By the previous construction, we clearly have  $\mathbf{X} = \text{DEC}(\mathcal{A}(\text{ENC}(\mathbf{X})))$ , if  $\mathcal{A}$  is deterministic.

We now establish a connection between two vectors that are close in hamming distance to the  $\ell_1$  sensitivity in the encoded graph.

► **Lemma 7 (Induced  $\ell_1$ ).** Given two neighboring vectors  $\mathbf{X} \sim_H \mathbf{X}' \in [n]^d$  under the hamming neighboring relationship and some real  $R \in \mathbb{R}^+$ . Then encoding both vectors into graph satisfies  $\|\text{ENC}_R(\mathbf{X}) - \text{ENC}_R(\mathbf{X}')\|_1 \leq 2\lceil R \rceil$

**Proof.** Observe that for two  $\mathbf{X} \sim_H \mathbf{X}'$  that differ in one coordinate and for any  $R$ , we have  $\text{ENC}_R(\mathbf{X}) \sim_1 G^{(1)} \sim_1 \dots \sim_1 G^{(2\lceil R \rceil)} \sim_1 \text{ENC}_R(\mathbf{X}')$  where each  $\mathbf{X}^{(i)}$  is the vector obtained by successively increasing and decreasing one coordinate in  $G^{(i-1)}$  by one. ◀

► **Lemma 8 (Group Privacy).** Let  $\mathbf{X} \sim_H \mathbf{X}' \in [n]^d$  be two vectors neighboring in hamming distance. Let  $\mathcal{M}$  be any mechanism that takes a weighted graph  $G \in \mathcal{G}_\omega$ , outputs an mst

$T \in \mathcal{T}(G)$  and preserves  $(\varepsilon, \delta)$ -DP under the  $\ell_1$  neighboring relationship. Fix some  $R \in \mathbb{R}^+$ . Then releasing  $\mathcal{M}(\text{ENC}(\mathbf{X}, R))$  satisfies  $(2R\varepsilon, 2Re^{2R\varepsilon}\delta)$ -DP.

**Proof.** Let  $\mathcal{M}$  be an  $(\varepsilon, \delta)$ -DP MST algorithm (respecting  $\sim_1$ ) and let  $G$  be the graph returned by  $\text{ENC}_R(\mathbf{X})$  on some input  $\mathbf{X} \in [n]^d$ . Then we know by Lemma 7 for all  $T \in \mathcal{T}(G)$ :

$$\begin{aligned} \Pr[\mathcal{M}(\text{ENC}(\mathbf{X})) = T] &\leq e^\varepsilon \Pr[\mathcal{M}(\mathbf{G}^{(1)}) = T] + \delta \\ &\leq e^\varepsilon \left( e^\varepsilon \Pr[\mathcal{M}(\mathbf{G}^{(2)}) = T] + \delta \right) + \delta \leq \dots \\ &\leq e^{2R\varepsilon} (\Pr[\mathcal{M}(\text{ENC}(\mathbf{X}')) = T] + 2Re^{2R\varepsilon}\delta) \end{aligned}$$

The last inequality follows because  $\delta \sum_{i=1}^{2R-1} e^{i\varepsilon} \leq 2Re^{2R\varepsilon}\delta$  (compare also [27, Lemma 2.2]).  $\blacktriangleleft$

We are now ready to combine everything and derive a lower bound.

**Proof of Theorem 5.** Assume that there exists an  $(\varepsilon, \delta)$ -dp algorithm  $\mathcal{B}$  with respect to the  $\ell_1$  neighboring relationship that takes a weighted graph and returns the edges of an approximate mst. For some fixed  $R \in \mathbb{R}^+$ , according to Lemma 7 and  $\mathbf{X} \sim_H \mathbf{X}'$  for  $\mathbf{X}, \mathbf{X}' \in [n]^n$ , we have for all possible outputs  $\mathbf{Y} \in [n]^n$ :

$$\Pr[\text{ENCDEC}_R(\mathcal{B}, \mathbf{X}) = \mathbf{Y}] \leq e^{2R\varepsilon} \Pr[\text{ENCDEC}_R(\mathcal{B}, \mathbf{X}') = \mathbf{Y}] + 2Re^{2R\varepsilon}\delta$$

which holds because of Lemma 8 and because decoding is only post-processing. Hence,  $\text{ENCDEC}$  satisfies  $(2R\varepsilon, 2Re^{2R\varepsilon}\delta)$ -dp. Now assume  $\mathbf{X} \in_R [n]^n$  to be uniformly drawn and set  $R = \min\left(\frac{\log n}{2c\varepsilon}, \frac{\log(1/\delta)}{2c\varepsilon}\right)$  for some real  $c$ . For the sake of contradiction, assume that  $\mathcal{B}$  approximates the real mst better than  $\frac{n-1}{100} \cdot R$ . This would imply that we leak at least  $\frac{99}{100}n$  coordinates where  $\text{ENCDEC}_R(\mathcal{B}, \mathbf{X})_i = X_i$ . This contradicts Lemma 4:

$$\begin{aligned} \Pr[\text{ENCDEC}_R(\mathcal{B}, \mathbf{X})_i = X_i] &\leq \frac{e^{2R\varepsilon}}{n} + 2Re^{2R\varepsilon}\delta \\ &\leq \frac{1}{n} \exp\left(\log(n^{1/c})\right) + \frac{\log(n)}{\varepsilon c} \exp\left(\log n^{1/c}\right) \delta \\ &= n^{(1-c)/c} + \frac{\log(n)}{\varepsilon c} n^{1/c} \delta \end{aligned}$$

For a sufficiently large constant  $c$  and considering that  $\delta \leq n^{-\Omega(1)}$ , we directly contradict  $\Pr[\text{ENCDEC}(\mathbf{X}, R)_i = X_i] \geq 0.99$  and note that we require  $R \leq \frac{\log n}{2c\varepsilon}$  to get any meaningful probability. Hence, the claim follows.  $\blacktriangleleft$

## B.2 Private All-Pairs Approximate Shortest Paths.

Is there any problem with stating the result for directed graphs? In any case we should be clear about whether it is directed or not. For a weighted graph  $G = (V, E, w)$  where  $w : E \rightarrow \mathbb{R}_{\geq 0}$ , we denote by  $\ell_G : V \times V \rightarrow \mathbb{R}_{\geq}$  the function such that for  $s, t \in V$ ,  $\ell_G(s, t)$  is the length of the shortest path between  $s$  and  $t$  in  $G$ . Sealfon [26], described an algorithm which for a given weighted graph  $G = (V, E, w)$ , produces a synthetic weighted graph  $\tilde{G} = (V, E, \tilde{w})$  such that with high probability in  $n$ ,  $|\ell_G(s, t) - \ell_{\tilde{G}}(s, t)| = O(\frac{n \log n}{\varepsilon})$  for all pairs  $s, t \in V$ . Moreover, the algorithm is  $\varepsilon$ -DP under the  $\ell_1$  neighbor relation. Sealfon also provided a lower bound showing that *any*  $\varepsilon$ -DP algorithm which releases an approximate shortest path must incur additive error  $\Omega(n/\varepsilon)$  even if we only care about a single pair  $s, t$ .

In this section, we show that with a slight modification (and simplification) of Sealfon's algorithm, we can release a synthetic graph  $\tilde{G}$  which simultaneously approximates all shortest paths in  $G$  within additive error  $O(n/\varepsilon)$  thus closing the gap between the upper and lower bound. The algorithm is presented in Algorithm 1.

■ **Algorithm 1**

---

1: **Procedure** DPALLPAIRSSHORTESTPATH

**Input:** Graph  $G = (V, E, w)$ , where  $w : E \rightarrow \mathbb{R}_{\geq 0}$ , privacy parameter  $\varepsilon$

2:     For  $e \in E$ , draw  $X_e \sim \text{Lap}(1/\varepsilon)$

3:      $\tilde{w}(e) = \max(0, w(e) + X_e)$

4:     **Return**  $\tilde{G} = (V, E, \tilde{w})$

5: **end Procedure**

---

In words, the algorithm adds Laplace noise to each edge but caps the noised weight at 0 which can be seen as a simple post-processing step. It turns out that this capping not only gives us the desirable property that all edges in the synthetic graph have positive weight, it is also crucial for our analysis. To demonstrate this, we can consider a complete graph  $G = K_n$  initially with all weights set to 0. If we just add Laplace noise  $\sim \text{Lap}(1/\varepsilon)$  to each edge, then with probability  $\geq n^{-0.1}$ , the new weight of an edge  $\tilde{w}(e) = -\Omega(\frac{\log n}{\varepsilon})$  and these events are independent. Letting  $\tilde{E} = \{e \in E \mid \tilde{w}(e) = -\Omega(\frac{\log n}{\varepsilon})\}$ , we have that  $G = (V, \tilde{E})$  has the distribution of an Erdős-Rényi graph  $G(n, p)$  where  $p \geq n^{-0.1}$ . It is easy to check that such a graph with high probability contains a path of length  $\Omega(n)$  between each pair of nodes  $s, t$ , so it follows that the shortest path between  $s, t$  in  $\tilde{G}$  will now have length  $-\Omega(\frac{n \log n}{\varepsilon})$ .

To avoid negative edge weights, Sealfon adds a large constant multiple of  $\frac{\log n}{\varepsilon}$  to each new edge weight. Union bounding over all edges, it is now easy to see that  $|d_G(s, t) - d_{\tilde{G}}(s, t)| = O(\frac{n \log n}{\varepsilon})$  for all pairs  $s, t$  and this analysis is tight, as seen for example by letting  $G$  be a path of length  $n$ . Our main result on Algorithm 1 is as follows.

► **Theorem 1 (Upper bound).** *Denote with  $d_G(s, t)$  the distance between  $s$  and  $t$  in some graph  $G$ . There exists an algorithm, that is  $\varepsilon$ -DP under the  $\ell_1$  neighborhood relation, which on input  $G$  releases a private synthetic graph  $\tilde{G}$  such that with probability  $1 - \exp(-\Omega(n))$ , it holds for all  $s, t \in V$  that  $|d_G(s, t) - d_{\tilde{G}}(s, t)| \leq 5n/\varepsilon$ .*

We will show that Algorithm 1 implies such an algorithm. This improves over the result by Sealfon in three ways: (1) the error bound is optimal, (2) the algorithm is simpler, and (3) the error probability is exponentially small in  $n$ . To prove it, we require the following simple tail bound on sums of Laplace random variables, which follows directly from the analogue bound in Theorem 5.1 of [13] for sums of exponential random variables.

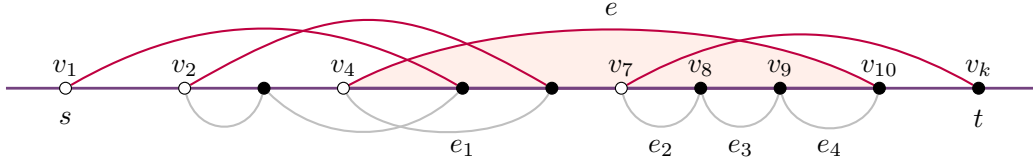
► **Lemma 9.** *Let  $X_1, \dots, X_n \sim \text{Lap}(1/\varepsilon)$  be i.i.d. random variables and  $X = \sum_{i=1}^n X_i$ . Then for  $\lambda > 1$ ,*

$$\Pr[|X| \geq (1 + \lambda)n/\varepsilon] \leq \frac{2}{\lambda} e^{-n(\lambda - 1 - \ln(\lambda))}.$$

*Thus, crudely, for  $\lambda \geq 6$*

$$\Pr[X \geq (1 + \lambda)n/\varepsilon] \leq e^{-n\lambda/2}.$$

**Proof of Theorem 1.** The fact that the algorithm is private follows from basic properties of the Laplace mechanism and post-processing, so we just have to analyze the error of the



■ **Figure 2** Sorted vertices by distance  $d(s, v_i)$  in the graph  $G$  and a path  $P$  from  $s$  to  $t$ . The red set of edges is constructed by iteratively removing edges that are covered by another edge, namely the gray edges. For example are the edges  $e_1, e_2, e_3, e_4$  are covered by  $e$ . The left-most node of each red edge is marked in white and the right-most node in black. As we prove in the claim, one can recover the red edges by matching the  $i$ 'th white node from the left to the  $i$ 'th black node from the left for  $i = 1, 2, \dots$ .

algorithm. For this it suffices to consider a fixed pair  $s, t$  since we can union bound over the  $\binom{n}{2}$  choices of such pairs.

Let thus  $s, t \in V$  be fixed and let  $P_0$  be a shortest path in  $G$  between  $s$  and  $t$ . For any subset of edges  $S \subset E$ , we let  $w(S) = \sum_{e \in S} w(e)$  and  $\tilde{w}(S) = \sum_{e \in S} \tilde{w}(e)$ . It follows immediately from Lemma 9 that with probability  $1 - e^{-n}$ ,  $|w(P_0) - \tilde{w}(P_0)| \leq 3n/\varepsilon$ . In this case,

$$d_{\tilde{G}}(s, t) \leq d_G(s, t) + 3n/\varepsilon \quad (5)$$

Our main challenge is therefore to show that for any other path  $P$  between  $s$  and  $t$ ,  $\tilde{w}(P)$  cannot be much smaller than  $w(P_0)$ . Note that a simple union bound over all  $n^{O(n)}$  such paths using Lemma 9, is insufficient since it only gives an error bound of  $O(\frac{n \log n}{\varepsilon})$ .

Our idea is to union bound over a more carefully selected family of sets of edges that we will construct below.

▷ **Claim 10 (Cover free set of edges).** Let  $v_1, \dots, v_n$  be the nodes of  $v$  sorted according to increasing distance from  $s$  in  $G$  (breaking ties arbitrarily except that we convene that  $s = v_1$ ). Let  $k$  be such that  $v_k = t$ . Let  $P$  be a simple path in  $G$  from  $s$  to  $t$ , and let  $E_P$  be the edges of the path. Then there exists a subset  $S \subset E_P$  such that:

1. For each  $a = 1, \dots, k-1$  there exists an edge  $e = (v_i, v_j) \in S$  with  $i \leq a$  and  $j \geq a+1$ .
2. If  $e_1 = (v_{i_1}, v_{j_1}), \dots, e_\ell = (v_{i_\ell}, v_{j_\ell})$  are the edges of  $S$  sorted such that  $i_1 < \dots < i_\ell$ , then also  $j_1 < \dots < j_\ell$ .
3.  $w(S) \geq w(P_0)$  where  $P_0$  is the shortest path in  $G$ .

**Proof of Claim.** The construction of  $S$  is visualized in Figure 2. Let  $e = (v_i, v_j) \in E$  and  $e' = (v_{i'}, v_{j'}) \in E$  be distinct edges in  $E$  with  $i < j$  and  $i' < j'$ . We say that  $e$  *covers*  $e'$  if  $i \leq i'$  and  $j \geq j'$ . We construct the set  $S$  as follows: Initially, let  $S$  be the edges of  $P$ . Should this just be the edges that go forward with respect to the indexing? While  $S$  contains distinct edges  $e, e'$  where  $e$  covers  $e'$ , we update  $S \leftarrow S \setminus \{e'\}$ . We claim that once no such update is possible, the set  $S$  satisfies the three requirements of the claim:

1. This clearly holds initially, since  $P$  is an  $s$ - $t$  path and must at some point cross from a vertex before or including  $v_a$  to a vertex after or including  $v_{a+1}$ . Moreover, by the requirement that a removed edge is always covered by another edge in  $S$ , the property is maintained after each update.
2. Assume for contradiction that this does not hold and let  $b$  be minimal such that  $j_{b+1} < j_b$ . Since also  $i_b < i_{b+1} < j_{b+1}$ , it follows that  $e = (i_b, j_b)$  covers  $e' = (i_{b+1}, j_{b+1})$ , a contradiction. Why is  $i_{b+1} < j_{b+1}$ ?

3. Denote by  $d_i = d(s, v_i)$ . By the triangle inequality, for  $i < j$ ,  $d(v_i, v_j) \geq d_j - d_i$ . It follows that

$$w(S) = \sum_{r=1}^{\ell} d(v_{i_r}, v_{j_r}) \geq \sum_{r=1}^{\ell} d_{j_r} - d_{i_r} \geq \sum_{r=1}^{k-1} (d_{r+1} - d_r) = d_k - d_1 = d(s, t) = w(P_0),$$

where the final inequality uses the first property of the set  $S$  in the claim. ◀

To finish the proof of the claim, let  $\mathcal{A}$  be the set of  $s$ - $t$  paths in  $G$  and define the map  $f : \mathcal{A} \rightarrow \mathcal{P}(E)$  such that for each path  $P \in \mathcal{A}$ ,  $f(P)$  is a subset of the edges of  $P$  satisfying the conditions in the claim. Define the family  $\mathcal{F} = \{f(P) \mid P \in \mathcal{A}\}$ . It follows from the second condition of the claim that to specify the edges of the set  $S$ , it suffices to specify the size  $\ell$  of  $S$  as well as the left endpoints  $v_{i_1}, \dots, v_{i_\ell}$  and right endpoints  $v_{j_1}, \dots, v_{j_\ell}$  of edges in  $S$ . Once this information is revealed, one can reconstruct  $S = \{(v_{i_1}, v_{j_1}), \dots, (v_{i_\ell}, v_{j_\ell})\}$ . It follows that  $|\mathcal{F}| \leq 2^n \cdot 2^n = 4^n$  where each factor accounts for choosing a subset of the set of nodes  $V$ . Let  $\mathcal{E}$  be the event that there exists an  $S \in \mathcal{F}$  such that  $\tilde{w}(S) \leq w(S) - 5\frac{n}{\varepsilon}$ . By Lemma 9 and a union bound,  $\Pr[\mathcal{E}] \leq 4^n e^{-2n} \leq e^{-n/2}$ . We claim that as long as  $\mathcal{E}$  does not hold, then  $\tilde{w}(P) \geq w(P_0) - 5\frac{n}{\varepsilon}$  for all  $P \in \mathcal{A}$ . To see this, fix  $P$  in  $\mathcal{A}$  and note that

$$\tilde{w}(P) \geq \tilde{w}(f(P)) \geq w(f(P)) - 5\frac{n}{\varepsilon} \geq w(P_0) - 5\frac{n}{\varepsilon},$$

where the first inequality uses that edges in  $\tilde{G}$  have non-negative edge weights and  $f(P) \subset P$ , the second inequality uses that we assumed  $\mathcal{E}^c$ , and the third inequality uses the third property in the claim. Thus

$$d_{\tilde{G}}(s, t) = \min_{P \in \mathcal{A}} \tilde{w}(P) \geq w(P_0) - 5\frac{n}{\varepsilon} = d_G(s, t) - 5\frac{n}{\varepsilon}. \quad (6)$$

Now, Equations (5) and (6) both hold except with error probability  $e^{-n/2} + e^{-n}$ . Combining them and union bounding over all  $s, t$ , we obtain the desired result. ◀

### B.3 Private Minimum Weight Perfect Matching

We now give new bounds for releasing the edges of a perfect matching. Given a public graph topology  $G = (V, E)$  together with weights  $\mathbf{W} \in \mathbb{R}_+^E$ , we want to find a subset  $M \subseteq E$  where each vertex  $v$  is adjacent to exactly one edge in  $M$  and minimizes the weight of  $w(M)$ . We denote the set of perfect matchings on some graph  $G$  as  $\mathcal{P}(G)$ . With some small modification, the previous technique in Theorem 5 also holds for privately releasing a minimum-weight perfect matching and we will prove the following Theorem 11

► **Theorem 11 (Lower Bound MPM).** *There exists a graph topology  $G = (V, E)$  and distribution of weights  $\mathcal{D}_\omega$  on weights of  $E$  such that for any  $(\varepsilon, \delta)$ -DP protocol  $\mathcal{B}$  that outputs an approximate minimum-weight perfect matching (mpm)  $T$  under the  $\ell_1$  neighboring relationship, if the weights  $\mathbf{W}$  of the graph are sampled according to  $\mathcal{D}_\omega$ , then with probability at least  $1/2$ , we have*

$$w(T) - w(T^*) \geq \Omega((n/\varepsilon) \cdot \min(\log(1/\delta), \log n))$$

where  $T^*$  is the optimal mpm in  $(G, \mathbf{W})$ .

The idea is again to encode a vector  $\mathbf{X} = (X_1, \dots, X_n) \in [n]^n$  into a perfect matching, this time on a weighted complete bipartite graph  $G$ . Each vertex  $v_i$  on the right side represents the values in  $[n]$  and will be mapped to the coordinates represented by the left part. Assuming that no two coordinates have the same value, it is clear that a perfect matching allows to reconstruct  $\mathbf{X}$ , but if they have, the previous construction does not work any more. Then, two coordinates  $X_i = X_j$  would be encoded by edges to the same vertex where only one can be part of the minimum-weight perfect matching. To fix this, the crucial observation is that with high probability, only a constant fraction of the coordinates will collide with some other value. Therefore, we cut the dataset  $\mathbf{X}' = (X_1, \dots, X_{n'})$  for some  $n' = \alpha n \in \Theta(n)$  for some  $0 < \alpha < 1$  and only encode those coordinates that don't collide with any other coordinate which is roughly  $\alpha n - \alpha^2 n$ , still a constant fraction of the dataset with high probability. We need the following standard lemma about balls and bins.

► **Lemma 12** (Constant fraction of collisions). *Let  $X_1, \dots, X_d$  be i.i.d. uniformly distributed in  $[n]$  with  $d = \alpha n$  for some fixed  $0 < \alpha \leq 1$ . Then, with probability at least  $1 - \exp(-\Omega(\alpha^3 n))$ , the number of indices  $i$  for which  $\exists j \neq i$  such that  $X_i = X_j$  is  $\mathcal{O}(\alpha^2 n)$ .*

**Proof.** Define  $\tilde{X}_i = \mathbf{1}\{\exists j \neq i : X_j = X_i\}$  and let  $Y = \sum_{i=1}^d \tilde{X}_i$ . By a union bound,  $\mathbb{E}[\tilde{X}_i] \leq \sum_{j \neq i} \Pr[X_j = X_i] = (d-1)/n$ , so that  $\mathbb{E}[Y] \leq d(d-1)/n = \alpha^2 n - \alpha \leq \alpha^2 n$ . Since changing a single  $X_i$  affects at most two of the  $\tilde{X}_i$ 's,  $Y$  is 2-Lipschitz. By McDiarmid's inequality [19], for any  $t > 0$ ,  $\Pr[Y \geq \mathbb{E}[Y] + t] \leq \exp(-2t^2/(4d))$ . Setting  $t = \mathbb{E}[Y]$  gives  $\Pr[Y \geq 2\mathbb{E}[Y]] \leq \exp(-\mathbb{E}[Y]^2/(2d)) = \exp(-\Omega(\alpha^3 n))$ . Thus, with probability at least  $1 - \exp(-\Omega(\alpha^3 n))$ , we have  $Y \leq 2\mathbb{E}[Y] = \mathcal{O}(\alpha^2 n)$ . ◀

In other words, inside a small constant fraction  $\alpha n$  of the dataset  $\mathbf{X}$ , only a constant fraction of  $\alpha^2 n$  has collisions with high probability. Therefore, we can get a collision-free subset of the dataset of size roughly  $\alpha n - \alpha^2 n \in \Theta(n)$  which is enough for our previous technique to work in this setting. We refer again to Figure 1 for a visualization.

► **Definition 13** (Encoding a perfect matching). **Encoding.** *Let  $\text{ENC}_R : [n]^d \rightarrow \mathcal{G}_\omega$  be the encoding function that takes some vector  $\mathbf{X} \in [n]^d$  and returns a weighted graph  $G = (V, E, \mathbf{W})$ . The parameter  $R \in \mathbb{R}$  is used to control the edge weights. Create a complete bipartite graph  $K_{n,n}$  with vertex set  $V = \{l_1, \dots, l_n, r_1, \dots, r_n\}$  and for all  $i, j \in [n]$ , set the weight of all edges  $e = \{l_i, r_j\}$  to be  $w_e = R$  if  $l_i \neq X_i$  and  $R$  elsewhere.*

**Decoding.** *Let  $\text{DEC}_G : \mathcal{M} \rightarrow [n]^d$  be a function parameterized by a weighted graph  $G \in \mathcal{G}_\omega$  with vertices  $V = \{l_1, \dots, l_n, r_1, \dots, r_n\}$ , together with an perfect matching  $M \in \mathcal{P}(G)$ . Return a vector  $\mathbf{X} = (X_1, \dots, X_d)$  where  $X_i = j$  if  $\{l_i, r_j\} \in M$ .*

By the same argument as in Lemma 7, we have for  $\mathbf{X}, \mathbf{X}' \in [n]^d$  and  $\mathbf{X} \sim_H \mathbf{X}'$ :  $\|\text{ENC}_R - \text{ENC}_R\|_1 \leq 2R$  and  $\text{ENCDEC}_R(\mathcal{A}, \mathbf{X}) = \mathbf{X}$ . Hence, also the following holds:

► **Lemma 14.** *Let  $\mathbf{X} \in [n]^d$  be some vector, and  $\mathcal{M}$  be any mechanism that takes a weighted graph  $G \in \mathcal{G}_\omega$ , outputs a perfect matching  $T \in \mathcal{P}(G)$  that preserves  $(\varepsilon, \delta)$ -DP under the  $\ell_1$  neighboring relationship and fix some  $R \in \mathbb{R}^+$ . Then releasing  $\mathcal{M}(\text{ENC}(\mathbf{X}, R))$  satisfies  $(2R\varepsilon, 2Re^{2R\varepsilon}\delta)$ -DP.*

**Proof Sketch of Theorem 11.** The proof is essentially the same as in Theorem 5, but intuitively, we only try to break a constant fraction  $\alpha n$  of the dataset rather than the complete one. This is already enough to derive the same contradiction as before. Therefore, fix some  $0 < \alpha \leq 1$  and draw  $\mathbf{X} \in_R [n]^d$ .



Problem	PN	Error Upper Bound	Error Lower Bound
MST	$\varepsilon$ -DP	$\mathcal{O}(n^2 \cdot \log n)^{\text{C,D}}$	$\Omega(n^2 \cdot \log n)^{\text{B}}$
	$(\varepsilon, \delta)$ -DP	$\mathcal{O}(n^{3/2} \cdot \sqrt{\log(n)} \cdot \sqrt{\log(1/\delta)})^{\text{C,D}}$	$\Omega(n^{3/2} \cdot \log(n))^{\text{C}}$
SP	$\varepsilon$ -DP	$\mathcal{O}(n^2 \cdot \log(n))^{\text{Theorem 15, o}}$	$\Omega(n)^{\text{A}}$
	$(\varepsilon, \delta)$ -DP	$\mathcal{O}(n^2 \cdot \sqrt{\log(n)} \cdot \sqrt{\log(1/\delta)})^{\text{A}}$	$\Omega(n)^{\text{A}}$
MWPM	$\varepsilon$ -DP	$\mathcal{O}(n^2 \cdot \log(n))^{\text{Theorem 15, o}}$	$\Omega(n^2 \cdot \log n)^{\text{Theorem 16}}$
	$(\varepsilon, \delta)$ -DP	$\mathcal{O}(n^2 \cdot \sqrt{\log(n)} \cdot \sqrt{\log(1/\delta)})^{\text{A}}$	$\Omega(n \cdot \min(\log(1/\delta), \log(n)))^{\text{Theorem 11}}$

■ **Table 2** Completing the landscape of private graph optimization problems for the  $\ell_\infty$  neighboring relationship. All bounds have a multiplicative factor  $(1/\varepsilon)$ , omitted for the sake of clarity. For those marked with (o) an efficient implementation is not known.

**References:** A) Sealfon [2016], B) Hladík and Tětek [2025], C) Pagh et al. [2024], D) Pinot et al. [2018]

As before, by Lemma 8, for some fixed  $R \in \mathbb{R}^+$ , according to Lemma 7 and  $\mathbf{X} \sim_H \mathbf{X}'$ , we have for all possible outputs  $\mathbf{Y} \in [n]^n$ :

$$\Pr[\text{ENCDEC}_R(\mathcal{B}, \mathbf{X}) = \mathbf{Y}] \leq e^{2R\varepsilon} \Pr[\text{ENCDEC}_R(\mathcal{B}, \mathbf{X}') = \mathbf{Y}] + 2Re^{2R\varepsilon}\delta$$

Now let  $\tilde{X} = (X_1, \dots, X_z)$  where  $z \leq \lceil \alpha n \rceil$  be some  $\alpha$  fraction of  $\mathbf{X}$  that has no collision and encode only this part into a graph  $G$ . By Lemma 12, we know that  $\tilde{X}$  has roughly  $\Theta(n)$  with very high probability. We can restore the original (larger) vector by just putting ones into the positions that have not been encoded. Finally, we can again get a contradiction by the same inequalities as in Theorem 5. We just need to replace the accuracy by  $\alpha^2 n - \alpha \in \Theta(n)$  instead of  $n - 1$  and adjust the constants accordingly. ◀

The function is parameterized by  $R \in \mathbb{R}$ , controlling the edge weights. A visualization of this construction is depicted in Figure 1.

## B.4 Results for the $\ell_\infty$ neighborhood

For completeness, we will now provide bounds for the  $\ell_\infty$  neighboring relationship. You can find a status in Table 2. We present new upper bounds obtained by invoking the exponential mechanism on the set of possible perfect matchings and shortest path trees. Unfortunately, it is unclear how to sample from them efficiently. Furthermore, we will use a similar lower bound technique as Hladík and Tětek to show tight bounds for privately releasing a perfect matching.

### B.4.1 Upper bounds for $\varepsilon$ -DP

We provide the following upper bound for *minimum-weight perfect matching* and *shortest paths* with a simple application of the exponential mechanism that gives  $\varepsilon$ -DP. Unfortunately, contrary to minimum spanning trees [11], we do not know how to sample from this distribution efficiently.

► **Theorem 15** (Exponential mechanism). *Assuming an  $\ell_\infty$  neighboring relationship, there exists an algorithm  $\mathcal{B}$  that is  $\varepsilon$ -DP and releases a matching (shortest path)  $M$  with additive error with high probability is*

$$w(M) - w(M^*) \leq \mathcal{O}((n \log n)/\varepsilon)$$

where  $M^*$  is the minimum weight perfect matching (shortest paths tree) in  $G$ .

**Proof sketch.** For matching, we can use the exponential mechanism on the set of all possible perfect matchings, which is  $\mathcal{O}(n^n)$ . As the sensitivity of two distinct matchings, can be up to  $\Delta = n/2$ , we sample each matching  $M \in \mathcal{M}(G) \propto \exp(\varepsilon/2\Delta) = \exp(\varepsilon/n)$ , which is  $\varepsilon$ -DP. By the utility of the exponential mechanism, with probability  $\exp(-\beta)$ , we have  $w(M) - w(M^*) \leq 2\Delta/\varepsilon(\ln(n^n) + \beta) \in \mathcal{O}((n/\varepsilon) \cdot \log(n))$ .

As there might be  $\mathcal{O}(n^n)$  many edge-disjoint paths from some vertex  $s$  to  $v$ , we can (instead of just outputting the edges on the  $s$ - $t$  path) apply the exponential mechanism on the set of spanning trees in  $G$ , which size is of the same order of magnitude. The utility function is just the sum over all the same argument as above applies again.  $\blacktriangleleft$

### B.4.2 A packing lower bound for $\varepsilon$ -DP

We now show tight error bounds under  $\varepsilon$ -DP for both the  $\ell_\infty$  neighboring relationships using a similar argument as proposed by Hladík and Tětek. This proof works also for  $\ell_1$  and matches the bound we gave in Theorem 11, and we will state both cases simultaneously.

The key observation is that similar to the private MST problem, we can again (greedily) construct an exponential-sized set of dissimilar matchings where each pair has a large hamming distance of  $\Theta(n)$ . We will show that the additive errors of  $\Theta(\frac{n}{\varepsilon} \log n)$  and  $\Theta(\frac{n^2}{\varepsilon} \log n)$  are asymptotically optimal under the  $\ell_1$  and  $\ell_\infty$  neighboring relationship respectively. For a given graph  $G = (V, E, \mathbf{W})$ , we shortly denote the optimal perfect matching as  $M_{\mathbf{W}}^*$ .

► **Theorem 16** (Lower bound  $\varepsilon$ -DP). *Let  $G$  be the complete bipartite graph  $K_{n,n}$  with  $2n$  vertices. Given a set of matching  $S \subseteq \mathcal{P}(G)$  such that  $M_1, M_2 \in S$  with  $d_H(M_1, M_2) > d$  for some  $d > 0$ . For some algorithm  $\mathcal{M}$  that returns a minimum-weight perfect matching of  $G$  under  $\varepsilon$ -DP, then there exists weights  $\mathbf{W}$ :*

$$\frac{1}{\sqrt{|S|}} \geq \begin{cases} \Pr[w(\mathcal{M}_G(\mathbf{W})) > w(M_G^*) + \frac{1}{128}(\frac{n}{\varepsilon} \log(n) - 8)] & (\text{with respect to } \ell_1) \\ \Pr[w(\mathcal{M}_G(\mathbf{W})) > w(M_G^*) + \frac{1}{64}(\frac{n^2}{\varepsilon} \log(n) - 8n)] & (\text{with respect to } \ell_\infty) \end{cases}$$

This lower bound also holds for the relaxation, where we don't require a matching to be perfect. We denote with  $\mathcal{L}_{\mathbf{W}}^G := \{M \in \mathcal{P}(G) \mid w(M) \leq w(M_{\mathbf{W}}^*) + \mu\}$  the set of perfect matchings with an additive error at most some  $\mu \in \mathbb{R}_+$ . We will prove it via a packing lower bound:

► **Theorem 17** ([27, 11] Packing argument for Matchings). *Fix an unweighted graph  $G = (V, E)$  and an arbitrary neighboring relation  $\sim$  on  $R^E$ . We are given a collection of weights  $\mathcal{W} \subseteq \mathbb{R}^E$  that all are at a distance at most  $r \in \mathbb{N}$  induced by  $\sim$  from some fixed weight vector  $\mathbf{W}_0 \in \mathbb{R}^E$ . For a chosen parameter  $\mu$  such that all sets in  $\mathcal{L}_{\mathbf{W}}^G$  are distinct and any  $\varepsilon$ -differentially private mechanism  $\mathcal{M}_G : \mathbb{R}^E \rightarrow \mathcal{M}(G)$ , there exists weights  $\mathbf{W} \in \mathcal{W}$ , such that  $\Pr[\mathcal{M}_G(\mathbf{W}) \in \mathcal{L}_{\mathbf{W}}^G] = \Pr[w(M_G(\mathbf{W})) \leq w(M_{\mathbf{W}}^*) + \mu] \leq \frac{\exp(r\varepsilon)}{|\mathcal{W}|}$ .*

While [11] used a complete graph for the reduction, the complete bipartite graph we also used in Theorem 11 is simpler to work with.

**Proof of Theorem 16.** Let  $G = (V, E)$  be the complete bipartite graph  $K_{n,n}$  with  $2n$  vertices and let  $S = \{S_1, \dots\} \subseteq \mathcal{M}(G)$  such that each pair is at least  $d$  apart in hamming distance. For each  $S_i \in S$ , we create a corresponding weight function  $\mathbf{W}_i$  where we set the weight of an edge to some value 0, if it is part of this matching and  $R \in \mathbb{R}_+$  otherwise. If we fix some arbitrary  $\mathbf{W}_0$ , we note that each of the other  $W \in \mathcal{W}$  have  $\|\mathbf{W} - \mathbf{W}_0\|_1 \leq \lceil 2R_1 n \rceil$  (for

$\ell_1$ ) and  $\|\mathbf{W} - \mathbf{W}_0\|_\infty \leq \lceil R_\infty \rceil$  (for  $\ell_\infty$ ) for two  $R_1, R_\infty \in \mathbb{R}_+$ . If we set  $\mu = Rn/2$ , we can show that all  $\mathcal{L}_{\mathbf{W}}^G$  are disjoint. Now we can apply Theorem 17. Assuming the  $\ell_1$  neighboring relationship, we have that for every algorithm  $\mathcal{M}$  that returns a perfect matching in  $G$ , there exists weights  $\mathbf{W} \in \mathcal{W}$ , such that

$$\Pr \left[ w(\mathcal{M}_G(\mathbf{W})) \leq w(\mathbf{M}_{\mathbf{W}}^*) + \frac{Rd}{2} \right] \leq \begin{cases} \frac{\exp((2Rn+1)\varepsilon)}{|S|} & \text{under } \ell_1 \\ \frac{\exp((R+1)\varepsilon)}{|S|} & \text{under } \ell_\infty \end{cases}. \quad (7)$$

Now we can use the same constants as used by [11] and set  $R := 1/4 \cdot \log|S|/(\varepsilon n) - 1/(2n)$  for  $\ell_1$  and  $R := 1/(2\varepsilon) \log|S| - 1$  for  $\ell_\infty$  which follows by solving the above numerator for  $\sqrt{|S|}$ . The claim follows by Lemma 19, which allows us to set  $d = n/4$  and  $|S| = 2^{n/4 \log n}$ .  $\blacktriangleleft$

► **Lemma 18** (Volume of a  $d$ -ball around perfect matching). *For the complete bipartite graph  $G$ , some  $T \in \mathcal{P}(G)$  and some real  $d > 0$ , we have  $|\{M' \in \mathcal{P}(G) | d_H(M, M') \leq d\}| \leq n^d$*

**Proof.** Let  $G$  be the  $K_{n,n}$  with bipartitions  $L$  and  $R$  and together with a perfect matching  $M$ . We can give a simple combinatorial upper bound by observing that we first pick some subset  $S \subseteq L$  of size  $d$  and then we get at most  $d!$  many ways to remap the vertices in  $S$ .

$$|\{M' \in \mathcal{P}(G) | d_H(M, M') \leq d\}| \leq \binom{n}{d} d! = \frac{n!}{(n-d)!} \leq n^d$$

$\blacktriangleleft$

► **Lemma 19** (Large set of dissimilar matchings). *Let  $G$  be the complete bipartite graph  $G$  with  $n > 2$  vertices on each side. Then we can construct a set  $S \subseteq \mathcal{P}(G)$  of size  $2^{n/4 \log n}$ , such that  $d_H(M_1, M_2) \geq n/4$  for all distinct  $M_1, M_2 \in S$ .*

**Proof.** For some  $n > 2$ , let  $G$  be the complete bipartite graph  $K_{n,n}$ . By a similar greedy argument as in [11], we can generate a set  $S \subseteq \mathcal{P}(G)$ , such that for all distinct  $M_1, M_2 \in S$ , we have  $d_H(M_1, M_2) > d$  and  $|S| \geq \frac{n!}{n^d}$  for some real  $0 < d \leq n$ . Initialize a set  $S := \emptyset$  and let  $R = \mathcal{P}(G)$  be the set of matchings that might still be added to  $S$ . Repeat the following processes until  $R$  is nonempty: Pick some  $M \in \mathcal{P}(G)$  and add it into  $S$ . Then, remove all at most  $n^d$  matchings from  $R$  that are closer than  $d$  in hamming distance to  $M$  (Lemma 19) and repeat. When this process stops we have  $|S| \geq \frac{n!}{n^d}$ . Now set  $d = n/4$  and we get  $|S| \geq \frac{n!}{n^d} \geq n^{n/2-d} = 2^{n/4 \log n} = \Theta(e^{n \log n})$  where we used the trivial lower bound of  $n! \geq n^{n/2}$  which holds for all  $n \geq 1$ .  $\blacktriangleleft$

## Previous Work.

Besides the model of *edge-weight* differential privacy [26], other studied models are edge-level [10] and node-level [16] privacy. We refer to [21, 18] for more background information on releasing various graph statistics under differential privacy constraints.

The canonical *input-perturbation* approach, depicted in Algorithm 2, is to release a private synthetic graph where noise is being added to each of the edges, where adding noise from a Laplace distribution gives  $\varepsilon$ -DP and  $(\varepsilon, \delta)$ -DP is obtained by using noise from a Gaussian distribution [26]). One way to upper-bound the total error is by union-bounding the maximum error for each single edge together with some observations on the relation

---

■ **Algorithm 2** : *Releasing private synthetic graph [26]*

---

```

1: Procedure INPUTPERTURBATION
   Input: Graph  $G = (V, E, \mathbf{W})$ , noise distribution  $\mathcal{D}$ 
2:   For  $e \in E$ , draw  $X_e \sim \mathcal{D}$ 
3:    $\hat{w}(e) = w(e) + X_e$ 
4:   Return  $\tilde{G} = (V, E, \tilde{\mathbf{W}})$ 
5: end Procedure

```

---

between the edges of a solution in the real and the synthetic graph. We will briefly survey the state of the problems individually.

**Minimum spanning trees.** Hladík and Tětek recently showed tight bounds using a packing argument that holds for  $\varepsilon$ -DP. Under the  $\ell_\infty$  neighboring relationship, releasing a full synthetic graph becomes too costly as we have to scale the noise with the number of edges. Therefore, Pinot introduced PAMST, a private version of the Prim-Jarník [25, 14] algorithm that gives asymptotically better utility. A private version of Kruskal’s algorithm has implicitly been used in the context of synthetic data generation [20].

Surprisingly, Pagh et al. recently showed that *input-perturbation* is more powerful than initially suspected. They showed that privacy will be amplified if we only release the *output* of an MST algorithm computed on a synthetic graph that is not fully private by itself. They also showed a tighter lower bound for  $\ell_\infty$ , leaving a small gap of  $\Omega(\sqrt{\log n} \cdot \sqrt{\log(1/\delta)})$ .

**Minimum-weight perfect matchings.** Known results [12, 5] used a slightly different model than used in this work. We are unaware of any other works than [26] for our setting.

**Shortest paths.** Releasing a shortest  $s - t$  path has mostly been studied from the perspective of privately releasing all pairwise shortest paths distances [8, 9, 2]. Again, we are unaware of works other than [26] that use the setting in this paper.