GROUP DECISION MAKINGS FROM PARTIAL PREFERENCES

Ao Liu

Submitted in Partial Fullfillment of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Approved by:
Lirong Xia, Chair
Malik Magdon-Ismail
Alex Gittens
Rongjie Lai

Department of Computer Science
Rensselaer Polytechnic Institute
Troy, New York

[May 2023]
Submitted March 2023

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

It's my great honor to receive RPI-IBM AI Horizon (AIHN) scholarship. I could not imagine how difficult my Ph.D. journey would be without the support from this scholarship. The collaboration and communication with other researchers at AIHN brought me the opportunity to explore many areas that I had never thought about before joining AIHN.

I am very fortunate te to be advised by Prof. Lirong Xia. Lirong not only provided me with a lot of professional advice in research but also shared many beneficial experiences, including US culture, communication skills, management, etc. Beyond that, Lirong is a great research collaborator with extraordinary research tastes. In our collaboration, I built up a correct judgment of "what is good", which greatly benefits my career.

I appreciate the help from my collaborators, including Prof. Zhenming Liu, Prof. Pinyan Lu, Prof. Nengkun Yu, Prof. Eakta Jain, Prof. Vassilis Zikas, Prof. Yu-Xiang Wang, Prof. Yongzhi Cao, Prof. Hanpin Wang, Prof. Yuqing Kong, Prof. Rohit Vaish, Prof. Andrew Duchowski, Prof. Reynold Bailey, Prof. Kenneth Holmqvist, Prof. Sanjeev Koppal, Dr. Francesca Rossi, Dr. Pinyu Chen, Dr. Sijia Liu, Dr. Chuang Gan, Dr. Zhibing Zhao, Dr. Chao Liao, Dr. Yun Lu, Dr. Qiong Wu, Dr. Brendan John, Dr. Yan Zhu, Mohamed Hammad, Farhad Mohsin, Qishen Han, Sikai Ruan, Zhechen Li, Xiaoyu Chen, Lei Luo, Wufei Ma, Inwon Kang, Yutao Sun, Yanting Wang, Weijian Zeng, Gary Wang, and all others. I could not imagine how difficult my Ph.D. journey would become without your feedback and/or contributions to my research projects. I also appreciate the help and collaboration from other members of our research group, including Prof. Sujoy Sikdar, Dr. Jun Wang, Joshua Kavner, Dr. Chunheng Jiang, Yuchen Liang, and many others.

I greatly thank the help from my doctorate committee members, anonymous reviewers, and the staff in CS Department (especially Tracy Hoffman and Shannon Carrothers). Finally, I want to thank my parents, Prof. Chaitanya Ullal (my previous advisor in Department of Material Science and Engineering), and all my friends for their support in my life.

ABSTRACT

Group decision making is the situation that a group of agents makes collective choices over a set of alternatives. The input of group decision making is the partial preferences of agents, and its output is one (or more) candidates. This thesis focuses on a two-step (*Preference learning* and *rank aggregations*) group decision making framework. A preference learning method learns a statistical ranking model from the users' preferences, which might be partial rankings. In rank aggregations, the group decision is made according to the learned ranking model. Both preference learning and rank aggregations are highly challenging when the number of alternatives becomes large. This dissertation focuses on solving the following questions. How can we design efficient preference learning algorithms when the preferences are partial rankings? How can we design rank aggregation rules with privacy guarantees?

The first part of this dissertation focuses on preference learning. We propose two preference learning algorithms, both compatible with partial preferences. Our main theoretical contributions are the complexity analysis of the proposed preference learning algorithms, which guarantees that all our proposed preference learning algorithms can finish in polynomial (or sub-polynomial) time. The experiments confirm that our algorithms are robust, efficient, and practical.

Furthermore, this dissertation proposes a group of rank aggregation methods with privacy guarantees. We found *Differential privacy* (DP), a widely accepted notion of privacy, is unsuitable in many rank aggregation scenarios because it requires external noises. Thus, we proposed a novel privacy notion, smoothed DP, for rank aggregations. The smoothed DP notion can achieve a similar privacy guarantee with DP without requiring external noises. We theoretically proved and experimentally confirmed that most real-world elections are private under the smoothed DP notion.

Finally, we apply the private group decision making framework to a downstream application, improving the robustness of interpretation maps. Interpretation maps explain the reason of why deep neural networks output a certain classification and can be used in the application scenarios like objective detection, medical recommendation, and transfer learning. Inspired by rank aggregation, we proposed the first interpretation method with theoretically guaranteed robustness against ℓ_∞-norm attacks. The proposed method is not only $\sim$30% more robust but also surprisingly more accurate than state of the art according to an exper-

iment on real-world datasets.

CHAPTER 1
INTRODUCTION

Group decision making is the situation that a group of agents makes collective choices over a set of alternatives. Its input is the partial preferences of agents, and its output usually is one candidate, which serves as the aggregated opinion of the agents. Within our scope, all of the previous works about group decision makings has one of the following two limitations,

1. Lack of theoretical guarantees about the effectiveness (*e.g.,* [1], [2]).
2. Cannot be applied to general partial orders (*e.g.,* [3] is only compatible with ranked top-l orders, [4] with l-way orders, and [5] with choice data).

To avoid the above two limitations above, we proposed a framework for group decision makings from partial preference (Figure 1.1), whose two key steps are *Preference learning* and *rank aggregations*. A preference learning method learns a statistical model for (full) rankings from the users' (partial) preferences. In rank aggregations, the group decision is made according to the learned ranking model. Both preference learning and rank aggregations are highly challenging when the number of alternatives becomes large.

Privacy is another main concern in group decision makings. The agents usually do not want their preferences to be known by others. If there are no high-level requirements on the accuracy of group decisions, privacy can be guaranteed by introducing the standard notion of *differential privacy (DP)*, which requires adding external noises (or decreasing accuracy) to the group decision. However, in some group decision making scenarios (*e.g.,* presidential elections), adding external noises are prohibited. This application scenarios require a suitable privacy notion and a private group decision making approach without requiring external noise.

This dissertation focuses on solving the following questions.

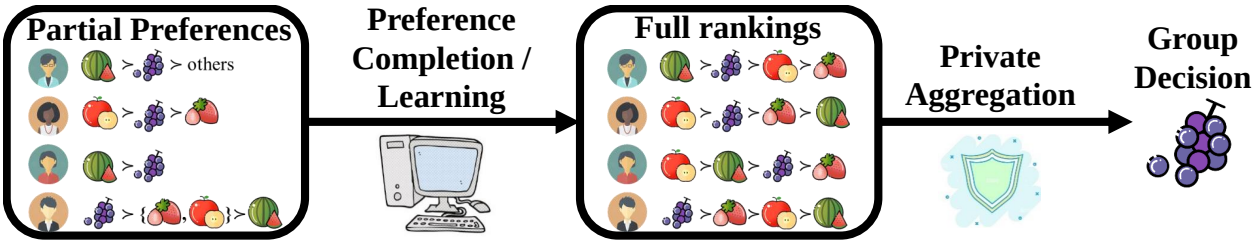

Figure 1.1: The framework of group decisions from partial preferences.

Can group decisions be made from general partial preferences with theoretical guarantees?

Can privacy be preserved in group decisions without adding external noises? Our answer to both open questions is "Yes".

1.1 Preference Learning from Partial Preferences

Preference learning is to learn a statistical ranking model from the data, and interpret the learned parameter as a ranking. The first part of this dissertation focuses on preference learning. We propose two preference learning algorithms, both compatible with partial preferences. In Chapter 2, we propose a non-parametric preference learning method, which is more suitable for the application scenarios when the number of agents is large. In Chapter 3, we propose a parametric preference learning method, which is more suitable for the application scenarios when the number of agents is small. Our main theoretical contributions are the complexity analysis of the proposed preference learning algorithms, which guarantees that all our proposed preference learning algorithms can finish in polynomial (or sub-polynomial) time. The experiments confirm that our algorithms are robust, efficient, and practical.

1.2 Smoothed Differential Privacy

In Chapter 4, we propose a group of rank aggregation methods with privacy guarantees. We found *Differential privacy* (DP), a widely accepted notion of privacy, is unsuitable in many rank aggregation scenarios because it requires external noises. Thus, we proposed a novel privacy notion, smoothed DP, for rank aggregations. The smoothed DP notion can achieve a similar privacy guarantee with DP without requiring external noises. We theoretically proved and experimentally confirmed that most real-world elections are private under the smoothed DP notion while non-private under DP.

1.3 Applications of Our Group Decision Making Framework

In Chapter 5, we apply the private group decision making framework to a downstream application, improving the robustness of interpretation maps. Interpretation maps explain why deep neural networks output a specific classification and can be used in the application scenarios like objective detection, medical recommendation, and transfer learning. Recent work [6] found that interpretation maps can be easily attacked. In other words, a delicately

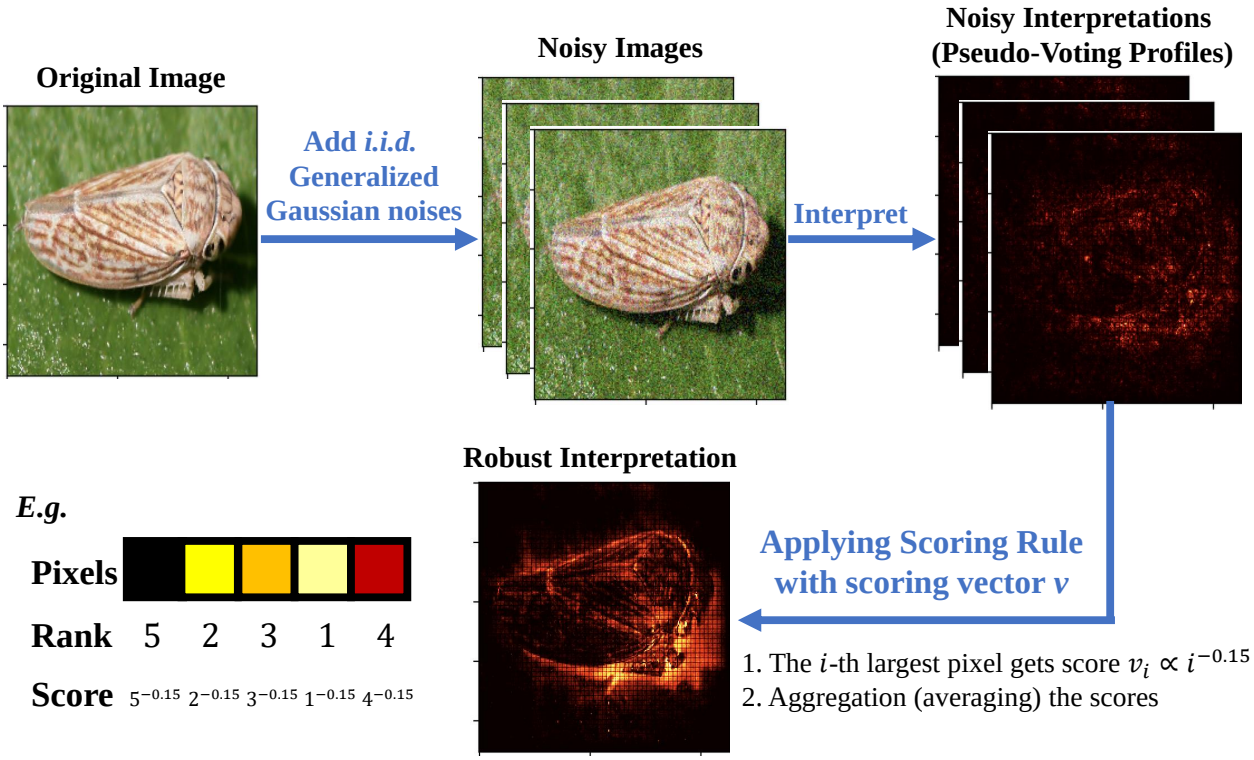

Figure 1.2: The framework to apply the proposed group decision framework to improve the robustness of interpretation maps.

designed small perturbation may significantly change the interpretation map without changing the classification result. Then, how to design interpretation methods with theoretically guaranteed robustness against attacks becomes an open and essential research question. Inspired by rank aggregation, we proposed an interpretation method, whose framework is shown in Figure 1.2, with theoretically guaranteed robustness against ℓ_d-norm attacks for arbitrary $d \in [1, \infty) \cup \{\infty\}$. At a high level, the first two steps in Figure 1.2 (adding noise and interpreting) focus on producing pseudo-voting profiles by adding noise to the images. The third step aggregates the pseudo-voting profiles by applying a scoring rule, which is a widely used rank aggregation method in real-world elections. The proposed method is not only $\sim$30% more robust but also surprisingly more accurate and more efficient than state-of-the-art according to the experiment on real-world datasets.

CHAPTER 2
NEAR-NEIGHBOR METHODS IN RANDOM PREFERENCE COMPLETION

2.1 Introduction

In a learning-to-rank problem, there is a set of agents (users) $\mathcal{X} = \{x_1, \dots x_n\}$ and a set of alternatives (items) $\mathcal{Y} = \{y_1, \dots y_m\}$. Each agent reveals her preferences over a subset of alternatives. The goal is to infer agents' preferences over all alternatives, including those that are not rated or ranked. This fundamental machine learning problem has many practical applications. For example, recommender systems use an agent' revealed preferences to discover other alternatives she might be interested in; product designers learn from consumers' past choices to estimate the demand curve of a new product; defenders can predict terrorists' preferences based on their past behavior; and political parties can evaluate campaign options based on voters' preferences. See [7] for a recent survey.

Rating vs. ranking. Agents' preferences can be represented by either a *rating* for each alternative (e.g., an integer rating in Netflix), or a *ranking* over the alternatives (i.e., complete ordering). Rating-based approaches have many known drawbacks [1], [2], including (i) agents often have different scales for ratings; and (ii) numeric values are often less robust than ranking-based approaches. In fact, rating data can always be converted to ranking data (e.g., y_1 is ranked higher than y_2 if y_1 has a higher rating) and thus ranking-based models and algorithms are more general. We focus on ranking data.

A common approach to infer an agent's preference is to first identify near neighbors of the agent in terms of the *Kendall-Tau* (KT) distance, then aggregate their rankings to produce a prediction. The KT distance is a metric that counts the number of pairwise disagreements between two ranking lists. This approach was proposed by [1], and their algorithm will be referred to as KT-kNN in this chapter. Many subsequent works are based on the following assumption [8]–[12].

Assumption 1. KT distance is a good measure of similarity between agents.

No theoretical justification for this assumption was known until recently. [2] proposed a latent utility model to justify the assumption. In their model, each agent or alternative is

Portions of this chapter have previously appeared as: A. Liu, Q. Wu, Z. Liu, and L. Xia."Near-neighbor methods in random preference completion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4336-4343.

associated with a latent feature. Alternative j's utility to agent i is controlled by a deterministic function of the similarity in their latent features. Under this model, consistency result is established for the KT-kNN algorithms.

However, this model assumes that agents' preferences are deterministic, which is unrealistic in many settings. For example, an agent can exhibit irrational behavior, or provide only a noisy version of her preferences. In fact, human preferences are often highly non-deterministic. Various statistical models have been built to model such randomness, pioneered by the Nobel Laureate McFadden [13] among many other researchers. Therefore, the following question remains open.

How can we learn an agent's random preferences from other agents' random preferences?

This question can be answered by designing algorithms for two closely-related problems: *(i) preference completion (PC):* given each agent's preferences over a subset of alternatives, the goal is to estimate its preference over all the alternatives. *(ii) near neighbors (NN):* given an agent x_i, the goal is to find agents $x_{i'}$ close to x_i in the latent space.

Standard techniques exist to use algorithms for the NN problem to solve a PC problem ([2], [1]; see also App. A.4). Therefore, we focus on the NN problem in this chapter.

Our Contributions. Our main conceptual contribution is the combination of a distance-based latent model and random preferences for learning to rank. To the best of our knowledge, while there is a large literature in each component, we are the first to consider both. See related work for more discussions.

Our model is called *distance-based random preference model*. Let the latent feature of agent i (alternative j) be x_i (y_j). Agent i's preferences are determined by a utility function $u(x_i, y_j) = \theta(x_i, y_j) + \epsilon_{i,j}$, where $\theta(x_i, y_j)$ is a deterministic monotonically decreasing distance-based function and $\epsilon_{i,j}$ is a zero mean independent random variable. Our model captures two pervasive characteristics of ranking datasets: *Ch1. Economically meaningful $\theta(\cdot, \cdot)$ function.* $u(x_i, y_j)$ is high in expectation when x_i and y_j are close. An agent is more likely to prefer alternatives with similar latent features to itself. *Ch2. Random preference model.* The function $u(x_i, y_j)$ contains a noise term $\epsilon_{i,j}$ to capture uncertainties in agents' behaviors.

Our technical contributions are two-fold. First, we prove that Assumption 1 does not hold anymore in our distance-based random preference model. More precisely, we prove that

the agents found by the KT-kNN algorithm [1] is far away from the given agents with high probability, even when $n, m \to \infty$.

Second, we design an "anchor-based" algorithm for finding an agent's near neighbors under random preferences. The algorithm is based on the following natural idea: if two agents i_1 and i_2 are close, then their KT distance to any other agent j (an anchor) should also be close. The algorithm proceeds by using the KT distance to other agents as an agent's feature, and measures the closeness between two agents by the L_1 distance of their features. We prove that asymptotically our algorithm identifies an agent's near neighbors with high probability when the latent space is 1-dimensional. Many techniques we developed can be generalized to high-dim settings.

Experiments on synthetic data verify our theoretical findings, and demonstrate that our algorithm is robust in high-dim spaces. Experiments on Netflix data shows that our anchor-based algorithm is superior to the KT-kNN algorithm and a standard collaborative filter (using the cosine-similarities to determine neighbors).

Related Work and Discussions. While using random utility models in learning-to-rank problems is not new [2], [12], [14]–[18], we are not aware of any that simultaneously achieves both *Ch1* and *Ch2*.

Random utility-based ranking algorithms [14], [12], [15] address *Ch2*, but the function $\theta(x_i, y_j)$ often does not have an explicit economics interpretation. For example, let $\Theta \in \mathbb{R}^{n \times m}$ be a matrix such that $\Theta_{i,j} = \theta(x_i, y_j)$. [12], [15] assume that Θ is low rank. But the low rank assumption does not have explicit economically interpretation.

While recent non-parametric models (e.g., [2]) allow one to use economically interpretable functions θ (addressing *Ch1*), they operate only under deterministic utility models.

Parametric preference learning has been extensively studied in machine learning, especially learning to rank [19]–[25]. These works are different with ours as it is often assumed that agents' preferences are generated from a parametric model.

2.2 Preliminaries

Distance-Based Random Preference Model. Let $\mathcal{X} = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ denote the set of agents and let $\mathcal{Y} = \{y_1, \ldots, y_m\} \subset \mathbb{R}^d$ denote the set of alternatives. We slightly abuse the notation and use x_i to refer to both agent i and her latent features. Each agent x_i has a ranking (preference list) $R_i = [y_{j_1} \succ \cdots \succ y_{j_m}]$ over $\mathcal{Y}$, where $\succ$ means "prefer to". We

observe only a subset of R_i for each $i \in [n]$.

Utility functions and the random utility model. Agent i's expected utility on alternative j is determined by a function $\theta(x_i, y_j)$. Throughout this chapter, we use $\theta(x_i, y_j) = \exp(-\|x_i - y_j\|_2)$, where $\|.\|_2$ is the ℓ_2-norm.

Agent i's ranking R_i is determined by the widely-used *Plackett-Luce model* [26], [27]. The realized utility of alternative y_j for agent i is generated by $u(x_i, y_j) \equiv \theta(x_i, y_j) + \epsilon_{i,j}$, where $\epsilon_{i,j}$ is a zero mean independent random variable that follows the Gumbel distribution. Then, agent i ranks the alternatives in decreasing order of their realized utilities. The density function of the Plackett-Luce model has a closed-form formula. Let $y_{j_1} \succ_i y_{j_2}$ represent that y_{j_1} is ahead of y_{j_2} in R_i and let $j_1, j_2, \dots j_m$ be a permutation of $[m]$. We have

$$\Pr\left[y_{j_1} \succ_i \cdots \succ_i y_{j_m}\right] = \prod_{t=1}^{m} \frac{\theta(x_i, y_{j_t})}{\sum_{t^*=t}^{m} \theta(x_i, y_{j_{t^*}})}. \tag{2.1}$$

The marginal distribution between alternatives j_1 and j_2 is $\Pr[y_{j_1} \succ_i y_{j_2}] = \frac{\theta(x_i, y_{j_1})}{\theta(x_i, y_{j_1}) + \theta(x_i, y_{j_2})}$.

Distributions of $\mathcal{X}$ and $\mathcal{Y}$. x_i and y_j are i.i.d. generated from distributions $\mathcal{D}_X$ and $\mathcal{D}_Y$. The supports of $\mathcal{D}_X$ and $\mathcal{D}_Y$ are a cube $\mathbf{B}(d)$ in $\mathbb{R}^d$, where $\mathbf{B}(d) = \{v \in \mathbb{R}^d : \|v\|_\infty \le c\}$, where c is a constant. We adopt the standard "near uniform" assumption for $\mathcal{D}_X$ and $\mathcal{D}_Y$ [28]–[31].

Definition 1. *Consider a continuous distribution $\mathcal{D}$ on $\mathbf{B}(d)$ with probability density function $f_{\mathcal{D}}(x)$. $\mathcal{D}$ is near-uniform if $\frac{\sup f_{\mathcal{D}}(x)}{\inf f_{\mathcal{D}}(x)}$ is bounded by a constant, where $x \in \mathbf{B}(d)$.*

Let f_X and f_Y be the PDFs of $\mathcal{D}_X$ and $\mathcal{D}_Y$ respectively. Define $c_X = \frac{\sup f_X(x)}{\inf f_X(x)}$ and $c_Y = \frac{\sup f_Y(x)}{\inf f_Y(x)}$.

Observation model. We observe only agent i's ranking over a subset $\mathcal{O}_i \subseteq [m]$ of alternatives. Each alternative j is in $\mathcal{O}_i$ independently with probability p. The $\mathcal{O}_i$'s are also independently generated across different agents. Let $R^{\mathcal{O}_i}$ be the ordered list over $\mathcal{O}_i \subseteq \mathcal{Y}$ that is consistent with R (i.e., $R^{\mathcal{O}_i}$ is the partial ranking of R over $\mathcal{O}_i$). For each agent i, we observe $R_i^{\mathcal{O}_i}$.

The near neighbor problem. Here, an algorithm needs to find near neighbors of an input agent. An algorithm is a $k(n, m)$-NN solver with parameter $\tau(n)$ if

- for any input agent i, the algorithm outputs k agents $i_1, i_2, \dots i_k$, and
- with overwhelming probability, $|x_i - x_{i_j}| \le \tau(n)$, where $\tau(n) = o(1)$.

We often write k-NN or kNN instead of $k(n,m)$-NN when k's dependencies on m and n are not critical.

Additional notations and examples. For an arbitrary ordered list R, we use it calligraphic form $\mathcal{R}$ to extract the rank of an alternative. For example, suppose y_j is the top-ranked alternative in R, then $\mathcal{R}(y_j) = 1$. Let $\mathbf{I}(v)$ be an indicator that sets to 1 if the argument v is true; if false, it sets to 0.

Let $|R|$ be the length of the list R. Let R_1 and R_2 be two ordered lists over the same set of alternatives. The normalized Kendall-Tau distance between R_1 and R_2 is

$$\mathrm{NK}(R_1, R_2) = \frac{1}{\binom{|R_1|}{2}} \sum_{j_1 \neq j_2 \in R_1} \mathbf{I}\Big(\big(\mathcal{R}_1(j_1) - \mathcal{R}_1(j_2)\big)\big(\mathcal{R}_2(j_1) - \mathcal{R}_2(j_2)\big) < 0\Big). \tag{2.2}$$

When R_1 and R_2 do not have the same support, the normalized KT distance is defined as $\mathrm{NK}(R_1^{\mathcal{O}}, R_2^{\mathcal{O}})$, where $\mathcal{O} = R_1 \cap R_2$.

To facilitate analysis, sometimes we need to introduce new agents outside $\mathcal{X}$. For a new agent with latent features x, let R_x denote its ranking over $\mathcal{Y}$ and let $R_x^{\mathcal{O}_x}$ denote the observed ranking.

Conditional probability and expectations. There are multiple levels of randomness for producing the rankings R_i's: (i) x_i and y_j are random and (ii) $u(x_i, y_j)$ consists of a random component (i.e., randomness from the Plackett-Luce model). Care must be taken when operating the conditional random variables defined in our process. For example,

- $\mathbb{E}[\mathrm{NK}(R_{i_1}, R_{i_2} \mid \mathcal{X}]$ refers to fixing the latent positions of the agents and taking expectations over $\mathcal{Y}$ and randomness from the Plackett-Luce model.
- $\mathbb{E}[\mathrm{NK}(R_{i_1}, R_{i_2}) \mid \mathcal{X}, \mathcal{Y}]$ refers to fixing the latent positions of both alternatives and agents and taking expectations over randomness from the Plackett-Luce model.

2.3 Inefficacy of KT-kNN

In this section, we will prove the inefficacy of KT-kNN algorithm by [1] (Algorithm 1) in our distanced-based random preference model. This implies that Assumption 1 does not hold in our model.

Recall that KT-kNN uses KT distances to find an agent's neighbors based on the intuition that when x_i and x_j are close, their "opinion" on alternatives' utilities should be

Algorithm 1: KT-kNN (it produces **incorrect** results)

Input: $\{R_j^{\mathcal{O}_j}\}_{j\in[n]}$, k, and an agent x_i.
Output: k neighbors near agent i in the latent space.
Find $j_1, j_2, \cdots, j_{n-1} (\in [n]/\{i\})$ such that $\mathrm{NK}(R_i^{\mathcal{O}_i}, R_{j_1}^{\mathcal{O}_{j_1}}) \leq \cdots \leq \mathrm{NK}(R_i^{\mathcal{O}_i}, R_{j_{n-1}}^{\mathcal{O}_{j_{n-1}}})$
Return $\mathcal{X}_{\text{KT-kNN}} \leftarrow \{j_1, \ldots j_k\}$

similar. The next theorem show that this intuition does not hold in our model, by proving that KT-kNN **does not return any near neighbors** for a large fraction of x_i.

Theorem 1. *Consider Algorithm* KT-kNN *under* distance-based random preference model *in which* $d = 1$ *and* $p = 1$. *Let* $\mathcal{D}_Y$ *and* $\mathcal{D}_X$ *be uniform distributions on* $[-1, 1]$. *For any constant* ϵ, *any* $x_i \in [-1 + \epsilon, -0.5] \cup [0.5, 1 - \epsilon]$, *and any* $k \leq n/\ln^5 n$, *we have*

$$\min_{x\in\text{KT-kNN}(\{R_j^{\mathcal{O}_j}\}_j, k, x_i)} ||x - x_i||_2 \geq \epsilon = \Omega(1). \tag{2.3}$$

with high probability. The probability comes from random $\mathcal{X}/\{x_i\}$, *random* $\mathcal{Y}$, *and random preferences.*

Remarks. Theorem 1 states that KT-kNN fails to work *even for the simple case where* $d = 1$ and $\mathcal{D}_X = \mathcal{D}_Y = \text{Uniform}([-1, 1])$. Eq. (2.3) is a strong result because trivial algorithms exist to find an agent x_j whose distance to x_i is $\Theta(1)$ (just picking up an arbitrary x_j). In addition, this result continues to hold for large populations (e.g., when $n, m \to \infty$ and $p = 1$), suggesting that the limitation of the KT-based approach roots at the structural properties of the NK function. In addition, *if we use* KT-kNN *to solve PC problem by applying standard techniques, it will also produce poor results* (see App. A.4 and Lemma 11 there).

Comparison to [2]. [2] proved that KT-kNN is effective under the *deterministic* utility model. This suggests that with the presence of uncertainties in the utility function (a more realistic assumption), the algorithmic structure of the NN problem is significantly altered.

Intuitions behind Theorem 1. The following example highlights the salient structures of KT-distances.

Example 1 (Near-neighbors in expectation)**.** Let $x_1 = 0$, $y_1 = -0.5$, and $y_2 = 1$. Let $x^* = \arg\min_x \mathbb{E}[\mathrm{NK}(R_x, R_1) \mid x_1, x, \mathcal{Y}]$ (e.g., where would we place an agent that minimizes its KT distance to x_1?). One would hope that when x^* and x_1 are close, R_{x^*} and R_1 is

close, but here $x^* = -0.5$. Specifically, let a be the probability x_1 prefers y_1 to y_2 (i.e., $a \equiv \frac{\theta(x_1,y_1)}{\theta(x_1,y_1)+\theta(x_1,y_2)}$) and let $b(x)$ be the probability x prefers y_1 to y_2. Recall that agents' support is $[-1, 1]$. We need to solve

$$x^* = \arg\min_x \mathbb{E}[\mathrm{NK}(R_x, R_1) \mid x_1, x, \mathcal{Y}] = \arg\min_x a(1 - b(x)) + (1 - a)b(x). \tag{2.4}$$

Here, we aim to minimize the weighted sum of $a \in [0, 1]$ and $(1 - a)$ via controlling $b(x)$. The optimal solution has a simple structure: when $a > (1 - a)$ (equivalently, $a > 0.5$), we need to set the weight associated with a as small as possible, which means setting $b(x)$ to the largest possible value. When $a < (1 - a)$, $b(x)$ needs to be minimized. Thus, the optimal solution uses the following threshold rules (assume $a \neq 0.5$ for simplicity).

$$x^* \in \begin{cases} (-1, y_1) & \text{if } a > 0.5 \\ (y_2, 1) & \text{if } a < 0.5 \end{cases}.$$

This minimizer is far from x_1. □

See also Example 5 in App. A.2.5 for another small and concrete example, in which KT-kNN produces poor output.

2.3.1 Proof Sketch of Theorem 1

Proof. We use intuitions from Example 1 to prove the theorem. Specifically, define

$$\mathcal{G}_i(x) \equiv \mathbb{E}[\mathrm{NK}(R_i, R_x) \mid x_i, x].$$

First, note that $\mathrm{NK}(R_i, R_x)$ concentrates at $\mathcal{G}_i(x)$ when m is sufficiently large. This comes from the concentration behavior of the NK function:

Lemma 1. *Let* $\mu = \mathbb{E}[\mathrm{NK}(R_i, R_j) \mid \mathcal{X}] = \mathcal{G}_i(x_j)$. *We have*

$$\Pr\left[|\mathrm{NK}(R_i, R_j) - \mu| \geq \delta\mu \mid \mathcal{X}\right] \leq 4m \exp\left(-\frac{\delta^2 m\mu}{6}\right).$$

See Appendix A.2.1 for the proof. The terms in NK are *not* independent terms so we cannot directly apply Chernoff bounds. Our proof uses the combinatorial structure of the NK function to decouple the dependencies among terms. The technique we develop can be

of independent interests.

Let $x^* = \arg\min_x \mathcal{G}_i(x)$. Below is our main lemma:

Lemma 2. *Let $\mathcal{D}_Y$ be uniform distribution on $[-1, 1]$. Let x_i be any agent in $[-1, -0.5]$. We have*

$$\arg\min_x \mathcal{G}_i(x) = -1.$$

Similarly, when $x_i \in [0.5, 1]$, $\arg\min_x \mathcal{G}_i(x) = 1$.

For any $x_i \in [-1, -0.5] \cup [0.5, 1]$, Lemma 1 and Lemma 2 give us:

$$\min_{i^*} \mathrm{NK}(R_{i^*}, R_i) \approx \min_{i^*} \mathcal{G}_i(x_{i^*}) \approx \mathcal{G}_i(-1),$$

where the first approximation comes from the concentration bound of NK and the second approximation comes from the fact that there must exist one agent close to -1 when the number of agent is large. Therefore, all the neighbors produced by KT-kNN are far from x_i (see Appendix A.2.4 for a rigorous analysis).

Proof of Lemma 2. By linearity of expectation, we have

$$\begin{aligned}\mathcal{G}_i(x) &= \frac{1}{\binom{m}{2}} \sum_{\ell_1 \neq \ell_2} \mathbb{E}\left[\mathrm{NK}\left(R_i^{\{y_{\ell_1}, y_{\ell_2}\}}, R_x^{\{y_{\ell_1}, y_{\ell_2}\}}\right) \middle| x, x_i\right] \\ &= \mathbb{E}\left[\mathrm{NK}\left(R_i^{\{y_{\ell_1}, y_{\ell_2}\}}, R_x^{\{y_{\ell_1}, y_{\ell_2}\}}\right) \middle| x, x_i\right].\end{aligned}$$

The last equality holds because y_j's are i.i.d. samples from $\mathcal{D}_Y$. Define

$$\begin{aligned}p_i(y_1, y_2) &\equiv \Pr[y_1 \succ_i y_2 \mid y_1, y_2, x_i] = \frac{\theta(x_i, y_1)}{\theta(x_i, y_1) + \theta(x_i, y_2)}. \\ p_x(y_1, y_2) &\equiv \Pr[y_1 \succ_x y_2 \mid y_1, y_2, x] = \frac{\theta(x, y_1)}{\theta(x, y_1) + \theta(x, y_2)}.\end{aligned}$$

When the context is clear, we shall refer to $p_i(y_1, y_2)$ and $p_x(y_1, y_2)$ as p_i and p_x, respectively. We have

$$\mathcal{G}_i(x) = \mathbb{E}_{y_1, y_2}\big[p_x(1 - p_i) + (1 - p_x)p_i \mid x_i, x\big].$$

One can see that $\mathcal{G}_i(x)$ is a smooth function (the first derivative exists). Our proof consists of three parts. *Part 1.* When $x \in (-1, x_i]$, $\partial \mathcal{G}_i(x)/\partial x > 0$, *Part 2.* When $x \in [x_i, -x_i]$, $\mathcal{G}_i(x) - \mathcal{G}_i(x_i) > 0$, and *Part 3.* When $x \in [-x_i, 1)$, $\partial \mathcal{G}_i(x)/\partial x > 0$.

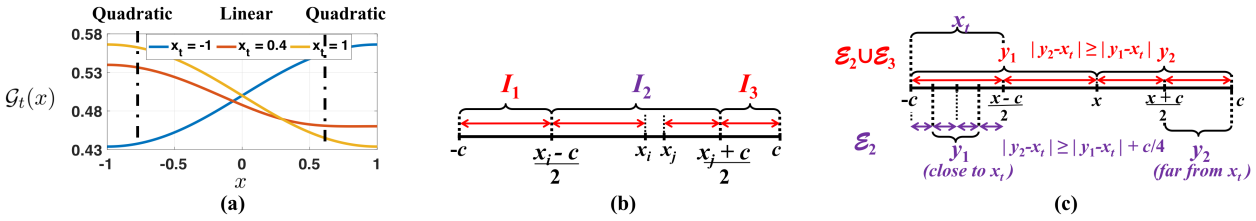

Figure 2.1: (a) Functions $\mathcal{G}_t(x)$ for $c = 1$, and $x_t = -1, 0.4$, and 1. Observations: *(i)* when $x_t = \pm 1$, the function $\mathcal{G}_t(x)$ is a monotone function; *(ii)* for all x_t, $\mathcal{G}_t(x)$ grows in sublinear manner when x is close to ± 1. (b) Definition of I_1, I_2 and I_3 used in the lower bound proof for Lemma 3 (Section 2.4). Agents in I_1 and I_3 are effective anchor agents. (c) Intuition of $\mathcal{E}_2$ and $\mathcal{E}_3$ in the proof of Lemma 4 (Section 2.4).

The proof for part 3 is similar to part 1. Proving part 2 is also simpler. Therefore, we focus only on the proof for part 1. Proof for part 2 and 3 is deferred to Appendix A.2.

We now show that when $x \in (-1, x_i]$, $\partial \mathcal{G}_i(x)/\partial x > 0$. We have (see Fact 2 in Appendix A.2.2):

$$\frac{\partial \mathcal{G}_i(x)}{\partial x} = \mathbb{E}\left[\Phi(y_1, y_2, x, x_i) \middle| x_i, x\right], \tag{2.5}$$

where

$$\begin{cases} \Phi(y_1, y_2, x, x_i) & \equiv \frac{e^{-\Delta_{1,2}^{(i)}} - 1}{e^{-\Delta_{1,2}^{(i)}} + 1} \cdot \frac{\text{sign}(y_1 - x) - \text{sign}(y_2 - x)}{4\cosh^2\left(\Delta_{1,2}^{(x)}/2\right)} \\ \Delta_{1,2}^{(i)} & \equiv |y_2 - x_i| - |y_1 - x_i| \\ \Delta_{1,2}^{(x)} & \equiv |y_2 - x| - |y_1 - x| \end{cases}.$$

Here, $\Delta_{1,2}^{(i)}$ ($\Delta_{1,2}^{(x)}$) measures whether x_i (x) is closer to y_2 or y_1. Similar to Example 1, they serve as important quantities determining the structure of $\partial \mathcal{G}_i(x)/\partial x$ (and therefore the optimal solution).

One can check that $\Phi(y_1, y_2, x, x_i) = \Phi(y_2, y_1, x, x_i)$. Therefore,

$$\frac{\partial \mathcal{G}(x)}{\partial x} = \mathbb{E}_{y_1, y_2}\left[\Phi(y_1, y_2, x, x_i) \mid x, x_i, (y_1 \leq y_2)\right].$$

Central to our analysis is carefully partitioning the event $y_1 \leq y_2$ into four disjoint (sub)-events. Under each event, the conditional expectation of Φ can be computed in a straightforward manner. Specifically, define

- $\mathcal{E}_1$: when $x \leq y_1 \leq y_2$ or $y_1 \leq y_2 \leq x$. Thus, $\Pr[\mathcal{E}_1 \mid y_1 \leq y_2] = \frac{x^2+1}{2}$.
- $\mathcal{E}_2$: when $y_1 < x$ and $y_2 \geq 1 + 2x_i$. Thus, $\Pr[\mathcal{E}_2 \mid y_1 \leq y_2] = -(x+1)x_i$.

- $\mathcal{E}_3$: when $y_1 < x$ and $x < y_2 < 2x_i - x$. Thus, $\Pr[\mathcal{E}_3 \mid y_1 \leq y_2] = (x+1)(x_i - x)$.
- $\mathcal{E}_4$: when $y_1 < x$ and $2x_i - x \leq y_2 < 1 + 2x_i$. Thus, $\Pr[\mathcal{E}_4 \mid y_1 \leq y_2] = \frac{(x+1)^2}{2}$.

Figure A.1(a) in Appendix A.1 visualizes the events to complement the analysis. We now interpret the meaning of these events.

Event $\mathcal{E}_1$. Event $\mathcal{E}_1$ represents the case in which y_1 and y_2 are on the same side of x. In this case, any movement of x without passing y_1, y_2 will not change the probability that $y_1 \succ_i y_2$ occurs (i.e., $\Pr[y_1 \succ_i y_2 \mid y_1, y_2, x, \mathcal{E}_1] = \Pr[y_1 \succ_i y_2 \mid y_1, y_2, x \pm \delta, \mathcal{E}_1]$ for any sufficiently small δ). Therefore, $\mathcal{E}_1$ does not impact $\mathbb{E}[\Phi]$ and $\mathbb{E}[\Phi \mid \mathcal{E}_1, x, x_i] = 0$.

Event $\mathcal{E}_2$. Under this event, one can check that $|x_i - y_1| < |x_i - y_2|$ (i.e., x_i is closer to y_1 than to y_2). In this case, when we increase the value of x (recall that $y_1 < x < x_i$), $\mathrm{NK}(R_i^{\{y_1,y_2\}}, R_x^{\{y_1,y_2\}})$ will increase. This is equivalent to $\mathbb{E}[\Phi \mid \mathcal{E}_2, x, x_i] > 0$.

Event $\mathcal{E}_3$. Under this event, one can check that $|x_i - y_1| > |x_i - y_2|$.In this case, when we increase the value of x, $\mathrm{NK}(R_i^{\{y_1,y_2\}}, R_x^{\{y_1,y_2\}})$ will decrease. This is equivalent to $\mathbb{E}[\Phi \mid x, x_i, \mathcal{E}_3] < 0$.

Event $\mathcal{E}_4$. Under this event $|x_i - y_1| - |x_i - y_2|$ is positive (we call this a positive event) with probability 0.5 and is negative (we call this a negative event) with probability 0.5. The first order term conditioned under the positive event cancels out that conditioned under the negative event. Therefore, $\mathbb{E}[\Phi \mid x, x_i, \mathcal{E}_4]$ will be a "small" term.

With the above intuition, we have

$$\begin{aligned}\mathbb{E}[\Phi \mid x, x_i] \approx\ & \mathbb{E}[\Phi \mid x, x_i, \mathcal{E}_2] \Pr[\mathcal{E}_2 \mid x, x_i] \\ & + \mathbb{E}[\Phi \mid x, x_i, \mathcal{E}_3] \Pr[\mathcal{E}_3 \mid x, x_i].\end{aligned}$$

We now relate $\mathcal{E}_2$ and $\mathcal{E}_3$ to Example 1. Event $\mathcal{E}_2$ corresponds to the setting where x_i is closer to y_1 than to y_2. As explained in Example 1, when we move x to the right, we *increase* the expected KT distance, whereas in $\mathcal{E}_3$ when we move x to the right, we *decrease* the expected KT distance. The crucial observation is that $\mathcal{E}_2$ is much more likely to happen than $\mathcal{E}_3$. The observation becomes clear with visualization in Fig. A.1b (i.e., the area for $\mathcal{E}_2$ is much larger than that for $\mathcal{E}_3$). Using these intuitions, we have (see Appendix A.2.2 for the full analysis):

Fact 1. *Using notations above, we have ,*

$$\mathbb{E}[\Phi \mid x, x_i] = \sum_{t=2,3,4} \mathbb{E}[\Phi \mid x, x_i, \mathcal{E}_t] \Pr[\mathcal{E}_t \mid x, x_i] > 0.$$

This completes the proof of Lemma 2. □

□

2.4 Anchor-Based Nearest Neighbor Algorithms

This section develops a new (high-dimensional) feature $\vec{F}_i$ for each agent i so that $|\vec{F}_i - \vec{F}_j|$ is small if and only if x_i and x_j are close. We present the positive results in the most general form in 1-dim latent space (i.e., we allow $\mathcal{D}_X$ and $\mathcal{D}_Y$ to be any near-uniform distribution, $p = o(1)$, and the latent space is $[-c, c]$ for any constant c).

Index convention. This section uses i, j, and t to index agents and ℓ (include ℓ_1, ℓ_2, etc.) to index alternatives.

Intuition of the design of $\vec{F}_i$. Our key idea is to leverage a third agent, namely an *anchor agent*, to determine the closeness of two agents x_i and x_j. Let x_t be a third agent. Compute $\mathrm{NK}(R_t, R_i)$ and $\mathrm{NK}(R_t, R_j)$. If x_t were chosen appropriately, then $\mathrm{NK}(R_t, R_i) \approx \mathrm{NK}(R_t, R_j)$ if and only if x_i and x_j are close. For example, if $x_t = -c$, then $\mathrm{NK}(R_t, R_i) \approx \mathcal{G}_t(x_i)$ and the function $\mathcal{G}_t(x)$ is a monotone function (see the blue curve in Fig. 2.1a). We have $\mathcal{G}_t(x_i) \approx \mathcal{G}_t(x_j)$ if and only if $x_i \approx x_j$.

We face two key technical challenges: *C1.* Not all agents can be served as effective anchor agents. For example, when $x_t = 0$, $\mathcal{G}_t(x) = \mathcal{G}_t(-x)$, which cannot separate x_i and x_j when $x_i = -x_j$. *C2.* the efficacy of the anchor agent is sensitive to x_i and x_j. For example, when $x_t = -c$ (see again Fig. 2.1a), the function $\mathcal{G}_t(x)$ grows in a sub-linear manner so it is less effective in detecting the closeness of x_i and x_j when they are both close to $-c$.

To address C1, we use *all possible* agents as anchor agents. To address C2, we need to develop new probabilistic techniques to analyze $\mathcal{G}_t(x)$.

Our features. Let $\vec{F}_i = (F_{i,1}, F_{i,2}, \cdots F_{i,n})$, where

$$F_{i,t} \equiv \mathbb{E}_{R_i, R_t, \mathcal{Y}}[\mathrm{NK}(R_i, R_t) \mid \mathcal{X}] = \mathcal{G}_t(x_i). \tag{2.6}$$

Then we use the L_1-distance between $\vec{F}_i$ and $\vec{F}_j$ to determine whether x_i and x_j are close.

Define

$$D(x_i, x_j) \equiv \frac{1}{n-2} \sum_{t \notin \{i,j\}} |F_{i,t} - F_{j,t}|. \tag{2.7}$$

The summation excludes i and j because x_i and x_j themselves cannot be used as anchor agents.

Replacing the features $\vec{F}$ by estimates. $F_{i,t}$ is not directly given so we use the empirical estimate as the plug-in estimator. Define $\hat{F}_{i,t} = \mathrm{NK}(R_i^{\mathcal{O}_i}, R_t^{\mathcal{O}_t})$ and our estimator is $\hat{D}(x_i, x_j) = \frac{1}{n-2} \sum_{t \notin \{i,j\}} |\hat{F}_{i,t} - \hat{F}_{j,t}|$ (see Algorithm 2).

Algorithm 2: Anchor-kNN

Input: $\{R_j^{\mathcal{O}_j}\}_{j \in [n]}$, k, and agent x_i.
Output: k neighbors near agent i.
Compute $\hat{F}_{i,j} = \mathrm{NK}(R_i^{\mathcal{O}_i}, R_j^{\mathcal{O}_j})$ for all $i, j \in [n]$.
Compute $\hat{D}(x_i, x_j) = \frac{1}{n-2} \sum_{t \neq i,j} |\hat{F}_{i,t} - \hat{F}_{j,t}|$.
Find $j_1, j_2, \cdots, j_{n-1}$ $(\in [n]/\{i\})$ such that
$\hat{D}(x_i, x_{j_1}) \leq \hat{D}(x_i, x_{j_2}) \cdots \leq \hat{D}(x_i, x_{j_{n-1}})$.
Return $\mathcal{X}_{\text{Anchor-kNN}} \leftarrow \{j_1, \cdots, j_k\}$.

Theorem 2. *Consider Algorithm* Anchor-kNN *under* distance-based random preference model. *Let $\mathcal{D}_X$ and $\mathcal{D}_Y$ be any near-uniform distribution on $[-c, c]$. Let $\tau(n)$ be an arbitrary quality parameter so that $\tau(n) = o(1) \wedge \tau(n) = \omega\left(n^{-\frac{1}{4}} \sqrt{\ln n}\right)$. For any $x_i \in [-c, c]$, any $m = \omega\left(\frac{\ln^3 n}{p^2 \cdot \tau^4(n)} \cdot \ln^2\left(\frac{\ln^3 n}{p^2 \cdot \tau^4(n)}\right)\right)$ and any $k = o(n \cdot \tau^2(n) \cdot \ln^{-1} n)$, we have*

$$\max_{x \in \text{Anchor-kNN}(\{R_j^{\mathcal{O}_j}\}_j, k, x_i)} ||x - x_i||_2 \leq \tau(n)$$

with high probability. The probability comes from random $\mathcal{X}/\{x_i\}$, random $\mathcal{Y}$, and random preferences.

Remark. $\tau(n)$ cannot be too small because our function $D(x_i, x_j)$ cannot measure the distance of two agents well if they are too close. m needs to grow when p (fewer samples) or $\tau(n)$ decreases (higher quality requirement), which is intuitive. k measures the number of numbers an algorithm can find so that their distance is within $\tau(n)$; larger k means the algorithm is more powerful.

Let $\mathbb{D}(x_i, x_j) = \mathbb{E}[D(x_i, x_j)]$. In the remainder of this section, we analyze the behavior of of $\mathbb{D}$. The function $\hat{D}$ concentrates at $\mathbb{D}$ and can be shown by using simple Chernoff bounds (see Appendix A.3.2 for a complete analysis).

Lemma 3. *For any near-uniform $\mathcal{D}_X, \mathcal{D}_Y$ on $[-c, c]$ and any two agents x_i, x_j, we have*

$$c_3(c) \cdot \left(\ln^{-1} n\right) \cdot |x_i - x_j|^2 \leq \mathbb{D}(x_i, x_j) \leq |x_i - x_j|,$$

where c_3 is a constant that depends only on c.

Upper bound proof for Lemma 3. The upper bound requires only a straightforward calculation. Recall that $p_i = p_i(y_1, y_2) = \Pr[y_1 \succ_i y_2 \mid y_1, y_2, x_i]$. We have

$$\begin{aligned} \mathbb{D}(x_i, x_j) &= \mathbb{E}_{x_t}\left[\left|\mathbb{E}_{y_1,y_2}[(p_i - p_j)(1 - 2p_t) \mid x_i, x_j, x_t]\right|\right] \\ &\leq \mathbb{E}_{y_1,y_2}\left[|p_i - p_j| \mid x_i, x_j\right] \leq |x_i - x_j|. \end{aligned} \tag{2.8}$$

The last inequality is shown in Fact 6 in App. A.3.1.

Lower bound bound proof for Lemma 3. Here we analyze only the case $|x_i - x_j| \leq 2c - 2\ln n$. When $|x_i - x_j| > 2c - 2\ln n$ (e.g., x_i and x_j are around the boundaries $-c$ and c, respectively), the result is trivial.

Wlog, assume that $x_i < x_j$. We partition $[-c, c]$ into three intervals and consider anchor agents in each of these intervals. Specifically, define (see also Figure 2.1(b))

$$I_1 \equiv \left[-c, \frac{x_i - c}{2}\right],\ I_2 \equiv \left(\frac{x_i - c}{2}, \frac{x_j + c}{2}\right) \,\&\, I_3 \equiv \left[\frac{x_j + c}{2}, c\right].$$

The agents in I_2 are "less effective anchors" (C2). We use trivial bound for terms in I_2 ($|F_{i,t} - F_{j,t}| \geq 0$). Focusing on I_1 and I_3, we have

$$\begin{aligned} &\mathbb{D}(x_i, x_j) \\ \geq\ &\frac{x_i + c}{4c \cdot c_X} \cdot \mathbb{E}_{x_t}\left[\left|\mathcal{G}_t(x_i) - \mathcal{G}_t(x_j)\right| \mid x_t \in I_1, x_i, x_j\right] + \\ &\frac{c - x_j}{4c \cdot c_X} \cdot \mathbb{E}_{x_t}\left[\left|\mathcal{G}_t(x_i) - \mathcal{G}_t(x_j)\right| \mid x_t \in I_3, x_i, x_j\right]. \end{aligned} \tag{2.9}$$

Note that $\left(\frac{x_i+c}{4c \cdot c_X} + \frac{c-x_j}{4c \cdot c_X}\right)$ is at least $\tilde{\Omega}(1)$. In the next Lemma, we will show that $\mathbb{E}_{x_t}\left[\left|\mathcal{G}_t(x_i) - \mathcal{G}_t(x_j)\right| \mid x_t \in I_1, x_i, x_j\right]$ is at least in the order of $|x_i - x_j|^2$. The analysis for the other term is similar. Below is the lemma we need (related to C2):

Table 2.1: Summary of results from Netflix dataset. KT coefficient is linearly related to KT distance and is in $[-1, 1]$.

	KT coefficient			Spearman Rho			Precision@5		
	CF	KT-kNN	Anchor-kNN	CF	KT-kNN	Anchor-kNN	CF	KT-kNN	Anchor-kNN
k =5	0.0531	0.0548	**0.0989**	0.0787	0.0811	**0.1462**	0.3286	0.3386	**0.4045**
k= 15	0.1199	0.1259	**0.1646**	0.1770	0.1860	**0.2423**	0.4291	0.4573	**0.4850**
k= 25	0.1403	0.1570	**0.1869**	0.2214	0.2307	**0.2742**	0.4718	0.4823	**0.5077**

Lemma 4. *For any near-uniform $\mathcal{D}_X, \mathcal{D}_Y$ on $[-c, c]$ such that $|x_i - x_j| \geq 2c - 2\ln n$ and $x_t \in I_1$, we have*

$$|\mathcal{G}_t(x_i) - \mathcal{G}_t(x_j)| \geq c_4(c) \cdot |x_i - x_j|^2,$$

where $c_4(c) = \frac{1-e^{-c/4}}{1+e^{-c/4}} \cdot \frac{1}{96c^2 \cdot \cosh^2(c) \cdot c_Y^2} \in (0, \frac{1}{2c})$.

Proof of Lemma 4. Note that $|x_i - x_j| \geq 2c - 2\ln n$ and $x_t \in I_1$ imply $x_j \geq x_i \geq 2x_t + c$. Because $|\mathcal{G}_t(x_i) - \mathcal{G}_t(x_j)| = \left|\int_{x_i}^{x_j} \frac{\partial \mathcal{G}_t(x)}{\partial x} dx\right|$, we aim to give a bound for $\frac{\partial \mathcal{G}_t(x)}{\partial x}$. Specifically,

$$\frac{\partial \mathcal{G}_t(x)}{\partial x} \geq 3c_4(c) \cdot (c^2 - x^2).$$

when $x \geq 2x_t + c$ (or equivalently, $x_t \leq \frac{x-c}{2}$). Re-cycle the definition of Φ, $\Delta_{1,2}^{(i)}$, and $\Delta_{1,2}^{(x)}$ used in the analysis of Theorem 1 (see also Appendix A.1 for the notation summary) so that $\frac{\partial \mathcal{G}(x)}{\partial x} = \mathbb{E}_{y_1 < y_2}[\Phi(y_1, y_2, x, x_t) \mid x, x_t]$

We partition the positions of $\{y_1, y_2\}$ into events (see also Fig. 2.1c for a visualization):

- $\mathcal{E}_1$: when $x \leq y_1 \leq y_2$ or $y_1 \leq y_2 \leq x$.
- $\mathcal{E}_2$: when $y_1 \in [\frac{x-7c}{8}, \frac{3x-5c}{8}]$ and $y_2 \geq \frac{x+c}{2}$. We have $\Pr[\mathcal{E}_2 \mid y_1 < y_2] \geq \frac{c^2-x^2}{16c^2c_Y^2}$.
- $\mathcal{E}_3$: when $(y_1, y_2) \notin \mathcal{E}_1 \cup \mathcal{E}_2$. As explained below, we do not need to explicitly calculate $\Pr[\mathcal{E}_3 \mid y_1 \leq y_2]$.

We now explain the intuition associated with these events.

Event $\mathcal{E}_1$. Because y_1 and y_2 are on the same side of x, we have $\mathbb{E}[\Phi \mid \mathcal{E}_1, x, x_t] = 0$ (see also Lemma 2).

Event $\mathcal{E}_2$ and event $\mathcal{E}_3$. When $(y_1, y_2) \in \mathcal{E}_2 \cup \mathcal{E}_3$, we have $|y_1 - x_t| \leq |y_2 - x_t|$ (x_t is closer to y_1 than to y_2). This is because *(i)* $|y_2 - x_t| \geq x - x_t$ when $y_2 \geq x$ and $x \in I_1$, and *(ii)*

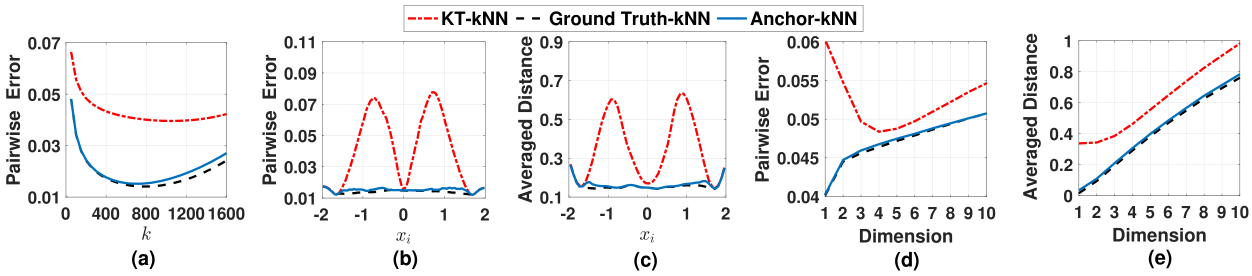

Figure 2.2: (a) Average pairwise probability prediction error (or pairwise error, see Appendix. A.5.1 for the definition) for different $k \in [101, 1601]$. (b) Pairwise error when $k = 751$ (optimal k for ground-truth kNN). (c) The average latent distance between x_i and its predicted 751 near neighbors. (d) Validation for high dimension ($1 \sim 10$ dimensional latent space). Let $k = 73 \approx \ln^2 n$ and measure the pairwise errors when $k = 73$. (e) The average latent distance when $k = 73$.

$|y_1 - x_t| \le \max\{x_t - c, x - x_t\} \le x - x_t$ since $y_1 \le x$ and $x \ge 2x_t + c$. This conclusion is trivial when the positions of the points are visualized (Figure 2.1(c)).

In these events, an increment in x will result in an increment in $\mathcal{G}_t(x)$. Therefore $\mathbb{E}[\Phi \mid \mathcal{E}_2 \cup \mathcal{E}_3, x, x_t] \ge 0$.

Event $\mathcal{E}_2$. Knowing Φ can be arbitrarily close to 0 when $\mathcal{E}_2 \cup \mathcal{E}_3$ happens. Here, we need also identify an event so that Φ is at least a positive constant. Event $\mathcal{E}_2$ serves for this purpose because it ensures x to be far from the mid-point of y_1 and y_2, so $\mathcal{G}(x)$ behaves like a linear function in this region (and thus its derivative behaves like a constant).

Intuition on dependencies on $|x_i - x_j|^2$. We now explain why $|\mathcal{G}_t(x_i) - \mathcal{G}_t(x_j)|$ depends on $|x_i - x_j|^2$ instead of $|x_i - x_j|$. Only if $(x_i, x_j) \notin \mathcal{E}_1$, Φ will be non-zero. This requires $y_1 \le x \le y_2$. Recall we only interests to $x_i \le x \le x_j$. When x_i and x_j get too close to $-c$, $\Pr[y_1 < x < y_2 \mid x] = \Theta(x_i + c) \approx \Theta(|x_i - x_j|)$. When we carry out integration over $\partial \mathcal{G}_t(x)/\partial x$, this term will lead to an additional factor of $|x_i - x_j|$ (see also Figure 2.1(a) for simulated $\mathcal{G}_t(x)$).

Using the above intuition, we have

$$\begin{aligned}\mathbb{E}_{y_1<y_2}[\Phi \mid x, x_t] &\ge \mathbb{E}_{y_1<y_2}[\Phi \mid \mathcal{E}_2, x, x_t] \cdot \Pr_{y_1<y_2}[\mathcal{E}_2 \mid x, x_t] \\ &= \Omega(1) \cdot \frac{c^2 - x^2}{16c^2 c_Y^2}.\end{aligned}$$

The last inequality uses the fact that $\mathbb{E}_{y_1 \le y_2}[\Phi \mid \mathcal{E}_2, x, x_i] = \Omega(1)$ (Fact 7 in Appendix A.3).

By carrying out an integration over $\partial\mathcal{G}_t(x)$, we get $|\mathcal{G}_t(x_i) - \mathcal{G}_t(x_j)| \geq c_4(c) \cdot |x_i - x_j|^2$. $\square$

Lemma 4 and its similar result for I_3 suffice to establish the lower bound.

2.5 Experiments

Our experiments aim to *(i)* confirm the behaviors of KT-kNN and Anchor-kNN for finite sample size when $d = 1$, *(ii)* understand the behavior of Anchor-kNN in high-dim settings, and *(iii)* validate the practicality of our algorithm over real-world datasets.

Details of all experiments are in Appendix A.5.1. Note that this chapter mostly focuses on *theoretical* results. Extensive evaluations on real-world data are a promising direction for future work.

1-dim synthetic data. In this experiment, we compare the efficacy of KT-kNN, Anchor-kNN, and Ground-Truth-kNN. Ground-Truth-kNN assumes an oracle access to an input's (ground-truth) k-nearest neighbors so this is an optimal kNN. Figure 2.2(a) plots the performance of Anchor-kNN,KT-kNN and Ground-truth-kNN in completing the preferences for different k. Figure 2.2(b) plots the completion/prediction error for different x_i using an optimal k (chosen by using cross-validations for Ground-Truth-kNN). Figure 2.2(c) shows the average latent distance between the return set and x_i. The experiments confirm that *(i)* Anchor-kNN is consistently better, *(ii)* KT-kNN does not return near neighbors when $x_i \approx \pm\frac{c}{2}$, and *(iii)* when $x_i \approx \pm\frac{c}{2}$, KT-kNN is the poorest at completing preferences. Appendix A.5.1 provides additional experiments for different settings on m.

High-dim synthetic data. We repeat the experiments for $d = 1, \cdots, 10$. Figure 2.2(d) is the prediction error and Figure 2.2(e) is the average distance. Anchor-kNN continues to outperform KT-kNN, suggesting that our algorithm works for $d > 1$. The performance of all algorithms deteriorate for large d because of the curse-of-dimensionality problem in neighbor-based algorithms [32].

Real dataset. We examine the performance of Anchor-kNN using the standard Netflix dataset [33], [34]. The baselines are KT-kNN and a standard collaborative filtering algorithm using cosine-similarity [35], [36]. Table 2.1 presents the results. Anchor-kNN consistently outperforms KT-kNN and collaborative filtering. Our experiments suggests that our distance-based random preference model and Anchor-kNN seem to be quite practical.

2.6 Conclusion

This chapter introduced a natural learning-to-rank model and showed that under this model a widely used KT-distance based kNN algorithm failed to find similar agents (users), challenging the assumptions made in many prior preference completion algorithms. To fix the problem, we introduced a new anchor-based algorithm Anchor-kNN that uses all the agents' ranking data to determine the closeness of two agents. Our approach is in sharp contrast to most existing feature engineering methods. We provided a rigorous analysis for Anchor-kNN for 1-dim latent space, and performed experiments on both synthetic and real datasets. Our experiments showed that Anchor-kNN works in high dim space and promises to outperform other widely used techniques.

CHAPTER 3
LEARNING PLACKETT-LUCE MIXTURES FROM PARTIAL PREFERENCES

3.1 Introduction

Rank aggregation has found applications in many areas, such as meta-search [37], information retrieval [7], and recommender systems [38]. In the rank aggregation problem, we want to find an aggregated ranking from the rank data of a given population. One popular approach is *learning to rank*, which is to learn a statistical ranking model from the data, and interpret the learned parameter as a ranking.

Plackett-Luce model (PL) [26], [27] is one of the most widely-used models. In PL, each alternative is assigned a positive value. The greater this value is, the more likely its corresponding alternative is ranked at a higher position. In recent years, mixtures of Plackett-Luce models have drawn attention for heterogeneous data. Mixture models provide better fitness to data [17], and can be used for clustering. A mixture model assumes that there are multiple clusters in the population, and each of the clusters can be learned using a component model. In the case of mixtures of Plackett-Luce models, each component is a Plackett-Luce model, while the quality of a certain alternative is usually different across different components.

Algorithms to learn a single Plackett-Luce model and its mixtures from *linear orders* (full rankings) have been well investigated. However, many real-world datasets are composed of *partial orders* over a subset of alternatives [39], [40], e.g. the Netflix prize data. Some of the previously proposed algorithms work only for special structures of partial orders or general partial orders with ignored comparisons or dependence between pairwise comparisons. However, modern datasets can be unstructured or even implicit, e.g., preferences implied by comparing time users spent on news articles or whether a user finished watching a movie [24]. None of the existing algorithms is able to learn PL and its mixtures with full utilization of general partial orders with no restrictions. The only known approach to tackle general partial orders without ignoring information in literature is to sample linear extensions from partial orders under a given model (Mallows' model) and learn the model from full rankings [39].

Portions of this chapter have previously appeared as: A.Liu, Z Zhao, C. Liao, P. Lu, and L. Xia."Learning plackett-luce mixture from partial preference," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019 pp. 4328-4335.

This chapter addresses the following problem: *How can we efficiently learn the Plackett-Luce model and its mixtures from partial orders?*

Our Contributions. We propose an EM framework to learn mixtures of Plackett-Luce models from partial orders. The E step consists of two parts: given the current estimate, (i) sampling multiple linear extensions from each partial order, and (ii) computing the membership of each sampled linear order (full ranking). The M step can adapt any learning algorithm for the single Plackett-Luce model. In this chapter, we use the LSR algorithm by [25] as the subroutine.

We address the technical challenge of sampling an linear extension of a partial order under PL in part (i) by proposing the following two samplers.

Sampler 1: GRIM-MCMC generalizes the GRIM-based sampling algorithm by [39], which was for Mallows' model, to PL. We further prove that GRIM-MCMC is efficient for a large class of preferences called *layered-structure preferences* in Theorem 5.

Sampler 2: Gibbs generalizes the sampler by [41] to deal with partial orders, and can be naturally generalized to learn mixtures of Random Utility Models [42], which subsume PL.

We compare the performances of the proposed sampling algorithms for PL and its mixtures using experiments on synthetic data and show that Gibbs sampler is better. Our algorithm uses 75% information at the cost of mere 6% MSE compared to the best algorithm for PL mixtures for linear orders. Experiments on real-world data show that the likelihood of test dataset increases when partial orders provide more information; or the number of components in PL mixtures increases within a certain range.

Related Work. We are not aware of any previous algorithm that learns a mixture of Plackett-Luce models from general partial orders without discarding information. A vast number of algorithms have been proposed to learn the Plackett-Luce model [4], [19], [25], [24], [43]–[45] and its mixtures [16], [46]–[48, 49]. Some of these algorithms were designed for linear orders (full rankings) [19], [45], [16], or some special structures of partial orders [4], [25], [24], [44], [48]–[50], e.g. top-l rankings. Very few algorithms were designed for general partial orders, but some ignore dependencies between pairwise comparisons with overlapping alternatives [47], and some discard comparison information in order for the algorithm to be consistent [43].

[39] proposed an algorithm to learn the Mallows model and its mixtures from general partial orders by sampling linear extensions. We focus on a very different but even more

popular model: Plackett-Luce model, as well as its mixtures. Efficiently sampling linear extensions from arbitrary distributions is an open problem. There have been research on generating linear extensions according to various distributions (*e.g.*, uniform distribution, Mallows model, *etc.*). Repeated insertion methods (RIM) and MCMC are two groups of methods widely used in those works.

We propose two algorithms to sample linear extensions from a partial order under the Plackett-Luce model: one inspired by [39] while the other based on the random utility nature of Plackett-Luce model. To our best knowledge, this is the first work that exploits all information from general partial orders by sampling linear extensions under the Plackett-Luce model, as well as other random utility models (Gibbs).

3.2 Preliminaries

Let $\mathcal{C} = \{c_1, \cdots, c_m\}$ denote a set of m alternatives. Let $\mathcal{L}(\mathcal{C})$ denote the set of linear orders, which are transitive, antisymmetric and total binary relations, over $\mathcal{C}$. A linear order R is often denoted by $c_{i_1} \succ c_{i_2} \succ \cdots \succ c_{i_m}$, which means that c_{i_1} is ranked at the top, c_{i_2} is ranked at the second, ..., c_{i_m} is ranked at the bottom. Let $\{1, \cdots, n\}$ denote n agents. Each agent j is assumed to have a strict complete preference over $\mathcal{C}$, represented by a linear order R_j. However, only partial preferences over a subset of alternatives $\mathcal{ES}_j$ is observed, represented by a *partial order*[1], which consists of irreflexive, transitive, and asymmetric binary relations. We call $\mathcal{ES}_j$ the *effective alternative set* of agent j and let $m_j^* = |\mathcal{ES}_j|$. We describe a partial order by a set of pairwise comparisons $V_j = \{c_{i_1} \succ c_{i'_1}, \cdots, c_{i_q} \succ c_{i'_q} | 1 \leq i_1, i'_1, \cdots, i_q, i'_q \leq m\}$. Due to transitivity, such description of a partial order is not unique. A unique representation of a partial order V_j is the transitive closure, denoted by $\texttt{tc}(V_j)$, which consists of all pairwise comparisons in V_j considering transitivity. An example of a partial order and its transitive closure is shown in Figure 3.1. Let $P = (V_1, \cdots, V_n)$ denote the data (also called a *preference profile*), where V_j is agent j's partial preference. Let $\mathcal{P}(\mathcal{C})$ denote the set of all partial orders.

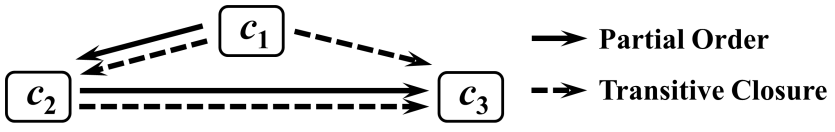

Figure 3.1: Example of a partial order and its transitive closure.

[1]Throughout this chapter, all partial orders are strict.

Let $\mathrm{Ext}(V_j)$ be the set of all *linear extensions* of V_j, which consists of all linear orders consistent with V_j. A linear extension bridges a partial order (observed data) and a linear order. For example, $\mathrm{Ext}(\emptyset)$ is the set of all permutations on $\mathcal{C}$. In this chapter, we assume each agent's preference is generated from the Plackett-Luce model, defined as follows.

Definition 2 (Plackett-Luce model)**.** *The sample space is $\mathcal{L}(\mathcal{C})^n$. The parameter space is $\Theta = \{\vec{\theta} = (\theta_1, \cdots, \theta_m) \mid \forall i, \theta_i \in (0,1), \sum_{i=1}^m \theta_i = 1\}$. Given parameter $\vec{\theta} \in \Theta$, the probability of any linear order $R = c_{i_1} \succ, \cdots, \succ c_{i_m}$ is*

$$\Pr_{PL}(R|\vec{\theta}) = \frac{\theta_{i_1}}{\sum_{p\geq 1}\theta_{i_p}} \times \frac{\theta_{i_2}}{\sum_{p\geq 2}\theta_{i_p}} \times \cdots \times \frac{\theta_{i_{m-1}}}{\theta_{i_{m-1}} + \theta_{i_m}}. \tag{3.1}$$

The probability of any partial order V is the sum of all probability of its linear extensions, i.e. $\Pr_{\mathrm{PL}}(V|\vec{\theta}) = \sum_{R\in\mathrm{Ext}(V)} \Pr(R|\vec{\theta})$. Another probability of interest is the probability of linear order R conditioned on a partial order V, where $R \in \mathrm{Ext}(V)$. By the definition of conditional probability, when $R \in \mathrm{Ext}(V)$, we have

$$\Pr{}_{\mathrm{PL}}(R|\vec{\theta}, V) = \frac{\Pr_{\mathrm{PL}}(R|\vec{\theta})}{\Pr_{\mathrm{PL}}(V|\vec{\theta})}. \tag{3.2}$$

The Plackett-Luce model has the random utility interpretation, where each alternative c_i is associated with a Gumbel distribution centered at $\ln\theta_i$, whose cumulative distribution function is $e^{-\theta_i \cdot e^{-x}}$ [51]. A linear order is generated by sampling a utility for each alternative from its corresponding utility distribution and ranking all the sampled utilities in descending order. For any $\vec{\theta} \in \Theta$, we further define $\eta_{\vec{\theta}} = \frac{\theta_{\max}}{\theta_{\min}}$, where $\theta_{\max} = \max_{i\in[m]}\theta_i$ and $\theta_{\min} = \min_{i\in[m]}\theta_i$.

Following [52] and [4], we make the following assumption on the data so that there exists a maximum likelihood estimation.

Assumption 1. *In every possible partition of the alternatives into two nonempty subsets, some individual in the second set beats some individual in the first set at least once.*

We define the mixture of k Plackett-Luce models (k-PL), which subsumes the Plackett-Luce model by letting $k = 1$, as follows.

Definition 3 (k-PL)**.** *Given integers $m \geq 2$ and $k \geq 1$, the k-mixture Plackett-Luce model is defined as follows. The sample space is $\mathcal{L}(\mathcal{C})^n$. The parameter space has two parts. The*

first part is the mixing coefficients $\vec{\alpha} = (\alpha_1, \cdots, \alpha_k)$*, where for all* $1 \leq r \leq k$*,* $\alpha_r \geq 0$ *and* $\sum_{r=1}^{k} \alpha_r = 1$ *The second part is* $\left(\vec{\theta}^{(1)}, \cdots, \vec{\theta}^{(k)}\right)$*, where* $\vec{\theta}^{(r)} \in \Theta$ *is the parameter of the r-th Plackett-Luce component. The probability of any linear order* R *is* $\Pr_{k\text{-}PL}(R|\vec{\theta}) = \sum_{r=1}^{k} \alpha_r \Pr_{PL}(R|\vec{\theta}^{(r)})$.

Similar to the (single) Plackett-Luce model, the probability of any partial order V is $\Pr_{k\text{-PL}}(V|\vec{\theta}) = \sum_{R \in \text{Ext}(V)} \Pr_{k\text{-PL}}(R|\vec{\theta})$. We note that this probability is equivalent to $\Pr_{k\text{-PL}}(V|\vec{\theta}) = \sum_{r=1}^{k} \alpha_r \Pr_{\text{PL}}(V|\vec{\theta}^{(r)})$.

3.3 An EM Framework for Learning PL Mixtures

We propose an EM framework where the E-step has two parts: (1) generating N linear extensions from each partial order, and (2) computing the expected membership of each linear order. In the next section, we will introduce the two sampling algorithms for part (1): GRIM-MCMC (Algorithm 5) and Gibbs Sampling (Algorithm 6).

Algorithm 3: EM Algorithm for k-PL with GRIM sampler and Gibbs sampler

Input: Profile $P = (V_1, \cdots, V_n)$, the number of components k, number of linear extensions per partial order N, the number of iterations T.
Output: For all $1 \leq r \leq k$, $\alpha_r^{(T)}$ and $\theta^{(r,T)}$
Initialization: Randomly generate $\vec{\theta}^{(1,0)}, \cdots, \vec{\theta}^{(k,0)}$, and mixing coefficients $\alpha_1^{(0)}, \cdots, \alpha_k^{(0)}$.
for $t = 1, \cdots, T$ **do**
    E-step:
    Sample N linear extensions for each partial order using e.g. Algorithm 5 (GRIM-MCMC) or Algorithm 6 (Gibbs);
    $\forall 1 \leq j \leq n, 1 \leq r \leq k$, compute $w_{jr}^{(t)}$ using (3.3).
    M-step:
    For all $1 \leq r \leq k$, update $\alpha_r^{(t)} = \frac{\sum_j w_{jr}^{(t)}}{n}$ and compute $\vec{\theta}^{(r)}$ using the LSR algorithm by [25].
end

Formally, let z_{jr} denote the membership indicator where $z_{jr} = 1$ means the linear order R_j belongs to the rth component. Let w_{jr} denote the weight of ranking R_j in rth component. For all j, we have $\sum_r w_{jr} = 1$. Given last iteration estimate $(\vec{\alpha}^{(t-1)}, \vec{\theta}^{(1,t-1)}, \ldots, \vec{\theta}^{(k,t-1)})$, each

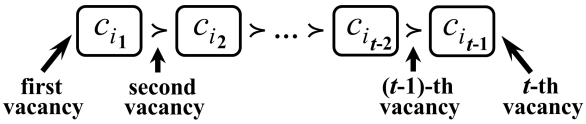

Figure 3.2: Definition and labeling of vacancies.

linear order R_j is clustered to each component by the following equation

$$w_{jr}^{(t)} = \Pr(z_{jr} = 1 | R_j, \vec{\theta}^{(r,t-1)}) = \frac{\Pr(R_j | \vec{\theta}^{(r,t-1)}) \cdot \alpha_r^{(t-1)}}{\sum_{s=1}^{k} \Pr(R_j | \vec{\theta}^{(s,t-1)}) \cdot \alpha_s^{(t-1)}}. \tag{3.3}$$

In the case of $k = 1$, the membership computation is redundant.

In the M-step, we update the mixing coefficients by $\alpha_r^{(t)} = \frac{\sum_j w_{jr}^{(t)}}{n}$ and the parameters of all components $(\vec{\theta}^{(1)}, \cdots, \vec{\theta}^{(k)})$ using the LSR algorithm proposed by [25]. The algorithm is formally shown as Algorithm 3.

3.4 Sampling Linear Extensions According to PL

In the section, we introduce two efficient algorithms to sample linear extensions of a partial order under PL, which are used in step 6 of Algorithm 3.

3.4.1 The GRIM-MCMC Sampler for Plackett-Luce

We extend the GRIM-based algorithm by [39] to Plackett-Luce model. Repeated insertion model (RIM) [53] is a polynomial-time tool applicable to exactly sample from Plackett-Luce model (given an empty partial order). Then, we generalize the Plackett-Luce RIM to an approximate sampler for linear extensions given an arbitrary partial order.

GRIM for Plackett-Luce Model. We first present the PL RIM sampler as the building block of GRIM. RIM samples a linear order by inserting alternatives one by one to the temporary ranking until the ranking is complete, where the order of insertions is arbitrary. W.l.o.g., we use the order from c_1 to c_m throughout this section. Let $R^{(t)} = [c_{r_1}^{(t-1)} \succ \cdots \succ c_{r_t}^{(t-1)}]$ denote the temporary ranking over alternatives $c_1, \cdots, c_{t-1}$. Let the i-th vacancy of this ranking as the i-th possible position for c_t to be inserted (shown in Figure 3.2), we define *conditional insertion probabilities* for PL RIM as:

$$\Pr_{\text{RIM}}\left(R^{(t)} \middle| R^{(t-1)}\right) \equiv \mathbb{1}_{R^{(t)} \in \text{Ext}\left(R^{(t-1)}\right)} \cdot \frac{\Pr_{\text{PL}}\left(R^{(t)} \middle| (\theta_1, \cdots, \theta_t)\right)}{\Pr_{\text{PL}}\left(R^{(t-1)} \middle| (\theta_1, \cdots, \theta_{t-1})\right)}. \tag{3.4}$$

The condition in indicator function $\mathbb{1}_{R^{(t)} \in \mathrm{Ext}\left(R^{(t-1)}\right)}$ guarantees c_t is inserted to one of the vacancies of $R^{(t-1)}$. Due to the space limit, we only present the generalized version of RIM in Algorithm 4, which will reduce to RIM if letting input V be $\emptyset$.

Theorem 3. *PL RIM is an exact sampler for Plackett-Luce model that runs in $O(m^2)$ time.*

Proof. We first prove the correctness of PL RIM. Slightly abusing the notation, we let $R^{(t)}$ be the temporary ranking over $c_1, \cdots, c_t$ following the same order as full ranking R. It's easy to check that $R^{(t)} \in \mathrm{Ext}\left(R^{(t-1)}\right)$ and $R = R^{(m)}$. Thus, for Plackett-Luce RIM defined above, we have, $\Pr_{\mathrm{RIM}}\left(R|\vec{\theta}\right) = \Pr_{\mathrm{RIM}}\left(R^{(m)}|R^{(m-1)}\right) \times \cdots \times \Pr_{\mathrm{RIM}}\left(R^{(2)}|R^{(1)}\right) \times \Pr_{\mathrm{RIM}}\left(R^{(1)}\right) = \Pr_{\mathrm{PL}}\left(R|\vec{\theta}\right)$.

We now analyze the complexity of calculating all insertion probabilities at the t-th insertion step. Instead of directly calculating (3.4) from (3.1), we use the following iterative formula to calculate $\Pr_{\mathrm{RIM}}\left(R_i^{(t)}\middle|R^{(t-1)}\right)$ from $\Pr_{\mathrm{RIM}}\left(R_{i-1}^{(t)}\middle|R^{(t-1)}\right)$:

$$\frac{\Pr_{\mathrm{RIM}}\left(R_i^{(t)}\middle|R^{(t-1)}\right)}{\Pr_{\mathrm{RIM}}\left(R_{i-1}^{(t)}\middle|R^{(t-1)}\right)} = \frac{\sum_{i'=i-1}^{t-1} \theta_{r_{i'}^{(t-1)}}}{\theta_t + \sum_{i'=i}^{t-1} \theta_{r_{i'}^{(t-1)}}} = \frac{A_i^{(t-1)}}{B_i^{(t-1)}}.$$

It is noteworthy that $A_i^{(t-1)}$ and $B_i^{(t-1)}$ can be calculated through the iterative formulas $A_i^{(t-1)} = A_{i+1}^{(t-1)} + \theta_{r_{i-1}}^{(t-1)}$ and $B_i = A_{i+1}^{(t-1)} + \theta_t$. The 2-level iteration process gives a $O(t)$ approach to calculate all needed probabilities and Theorem 3 follows by combining all m insertion steps. □

Algorithm 4: Plackett-Luce GRIM

1 **Input:** Partial order V, Plackett-Luce parameters $\vec{\theta} = (\theta_1, \cdots, \theta_m)$ (θ_i corresponds to alternative c_i).
2 **Initialization:** Set $R^{(1)} = [c_1]$ as initialized ranking.
3 **for** $t = 2$; $t \leq m$ **do**
4 Insert alternative c_t to $R^{(t-1)} = [c_{r_1}^{(t-1)} \succ c_{r_2}^{(t-1)} \succ \ldots \succ c_{r_{t-1}}^{(t-1)}]$'s i^{th} vacancy ($1 \leq i \leq t$) with probability calculated by (3.5).
5 **end**
6 **return** $R^{(m)}$.

Now we are ready to propose PL GRIM, which samples linear extensions of any given partial order, where in each step we only insert the current alternative to vacancies consistent

with the partial order. We define the GRIM insertion probability by

$$\Pr_{\text{GRIM}}\left(R_i^{(t)}\middle|\, R^{(t-1)}\right) \equiv \frac{\mathbb{1}_{R_i^{(t)}\in \text{Ext}(V)}}{Z_{V,R^{(t-1)},\vec{\theta}}} \cdot \frac{\Pr_{\text{PL}}\left(R_i^{(t)}\middle|\,(\theta_1,\cdots,\theta_t)\right)}{\Pr_{\text{PL}}\left(R^{(t-1)}\middle|\,(\theta_1,\cdots,\theta_{t-1})\right)}, \tag{3.5}$$

where

$$Z_{V,R^{(t-1)},\vec{\theta}} = \sum_{i=1}^{t}\left(\mathbb{1}_{R_i^{(t)}\in \text{Ext}(V)} \cdot \frac{\Pr_{\text{PL}}\left(R_i^{(t)}\middle|\,(\theta_1,\cdots,\theta_t)\right)}{\Pr_{\text{PL}}\left(R^{(t-1)}\middle|\,(\theta_1,\cdots,\theta_{t-1})\right)}\right) \tag{3.6}$$

is the normalization factor. Formally, GRIM procedure is shown in Algorithm 4 (note again its fourth-step used (3.5) and thus used the information from V).

Theorem 4. *PL GRIM (Algorithm 4) runs in $O(m^2)$ time, and the probability of outputting full ranking R is $\Pr_{GRIM}\left(R\middle|\,\vec{\theta},V\right) = \frac{\mathbb{1}_{R\in Ext(V)}}{\prod_{t=1}^{m-1} Z_{V,R^{(t)},\vec{\theta}}} \cdot \Pr_{PL}\left(R\middle|\,\vec{\theta}\right)$.*

Table 3.1: Example of GRIM (probability columns show conditional probabilities on given rankings).

Insert c_1, c_2		Insert c_3	
Ranking	Pr	Ranking	Pr
$c_1 \succ c_2$	$\frac{1}{1+2} = \frac{1}{3}$	$c_1 \succ c_3 \succ c_2$	3/5
		$c_1 \succ c_2 \succ c_3$	2/5
$c_2 \succ c_1$	$\frac{2}{1+2} = \frac{2}{3}$	$c_2 \succ c_1 \succ c_3$	1

The distribution of PL GRIM is not always exactly the distribution in (3.2) (See Example 2).

Example 2. *Consider the Plackett-Luce model over $\mathcal{C} = \{c_1, c_2, c_3\}$, partial order $V = \{[c_1 \succ c_3]\}$ and parameter $\vec{\theta} = (1/6, 2/6, 3/6)$, the process of GRIM is shown in Table 3.1. The comparison between GRIM and Placket-Luce model is shown in Table 3.2.*

Table 3.2: Compare the distributions between GRIM and PL.

	$c_1 \succ c_3 \succ c_2$	$c_1 \succ c_2 \succ c_3$	$c_2 \succ c_1 \succ c_3$
$\Pr_{\text{GRIM}}$	1/5	2/15	2/3
$\Pr_{\text{PL}}$	2/5	4/15	1/3

Even though GRIM is not a precise sampler for Plackett-Luce model, it provides us a good proposal distribution on linear extensions for Metropolis–Hastings algorithm [54],

[55], which is an MCMC algorithm that tunes a proposal distribution to the target distribution. It works as follows. For each state β we first generate a candidate next state γ from a proposal distribution $\hat{\mathcal{P}}(\cdot)$ (slightly abusing this notation). Then, let the next state to be γ with probability $\min\left\{1, \frac{\mathcal{P}(\gamma)\hat{\mathcal{P}}(\beta|\gamma)}{\mathcal{P}(\beta)\hat{\mathcal{P}}(\gamma|\beta)}\right\}$, otherwise the next state remains β, where $\mathcal{P}(\cdot)$ is the (probability of) target distribution.

GRIM-MCMC. In this section, we present the Markov Chain $\boldsymbol{\mathcal{M}}_{\text{PL}}$ used to tune PL GRIM in Algorithm 5. In each step of $\boldsymbol{\mathcal{M}}_{\text{PL}}$, we first use Algorithm 4 to generate a full ranking γ, which is a linear extension of preference V. Then, let $\beta = \gamma$ with probability $\min\left\{\frac{\Pr_{\text{PL}}(\gamma|\vec{\theta})}{\Pr_{\text{PL}}(\beta|\vec{\theta})} \cdot \frac{\Pr_{\text{GRIM}}(\beta|\vec{\theta})}{\Pr_{\text{GRIM}}(\gamma|\vec{\theta})}, 1\right\}$ or remain the next ranking as the current ranking β otherwise.

Algorithm 5: Tuned Plackett-Luce GRIM by Markov Chain $\boldsymbol{\mathcal{M}}_{\text{PL}}$

1 **Input:** Plackett-Luce parameters $\vec{\theta} = (\theta_1, \cdots, \theta_m)$, partial order V_j and number of iteration N.
2 **Initialization:** Generate an linear extension β over alternative set $\mathcal{ES}_j$ by Algorithm 4.
3 **for** $n = 1$; $n \leq N$ **do**
4 Generate another linear extension γ using parameters $\vec{\theta}$ over alternative set $\mathcal{ES}_j$ using Algorithm 4.
5 With probability $\min\left\{\frac{\Pr_{\text{PL}}(\gamma|\vec{\theta})}{\Pr_{\text{PL}}(\beta|\vec{\theta})} \cdot \frac{\Pr_{\text{GRIM}}(\beta|\vec{\theta})}{\Pr_{\text{GRIM}}(\gamma|\vec{\theta})}, 1\right\}$, let $\beta = \gamma$.
6 **end**
7 Insert all remaining alternatives in $\mathcal{C} \setminus \mathcal{ES}_i$ to β using Algorithm 4.
8 **return** R.

Before analyzing the mixing time of $\boldsymbol{\mathcal{M}}_{\text{PL}}$, we first show the probability $\frac{\Pr_{\text{PL}}(\gamma|\vec{\theta})}{\Pr_{\text{PL}}(\beta|\vec{\theta})} \cdot \frac{\Pr_{\text{GRIM}}(\beta|\vec{\theta})}{\Pr_{\text{GRIM}}(\gamma|\vec{\theta})}$ used in step 5 of Algorithm 4 can be easily computed after GRIM sampling in the following Proposition.

Proposition 1. $\frac{\Pr_{PL}(\gamma|\vec{\theta})}{\Pr_{PL}(\beta|\vec{\theta})} \cdot \frac{\Pr_{GRIM}(\beta|\vec{\theta})}{\Pr_{GRIM}(\gamma|\vec{\theta})} = \prod_{t=1}^{m-1} \frac{Z_{V,\gamma^{(t)},\vec{\theta}}}{Z_{V,\beta^{(t)},\vec{\theta}}}$.

Proof. By Theorem 4, we have,

$$\begin{aligned}
&\frac{\Pr_{\mathrm{PL}}(\gamma)\cdot \Pr_{\mathrm{GRIM}}(\beta)}{\Pr_{\mathrm{PL}}(\beta)\cdot \Pr_{\mathrm{GRIM}}(\gamma)}\\
=&\frac{\frac{\mathbb{1}_{\gamma\in \mathrm{Ext}(V)}}{Z_V}\cdot \Pr_{\mathrm{PL}}(\gamma|\vec{\theta},V)}{\frac{\mathbb{1}_{\beta\in \mathrm{Ext}(V)}}{Z_V}\cdot \Pr_{\mathrm{PL}}(\beta|\vec{\theta},V)}\cdot\frac{\frac{\mathbb{1}_{\beta\in \mathrm{Ext}(V)}}{\prod_{t=1}^{m-1} Z_{V,\beta^{(t)},\vec{\theta}}}\cdot \Pr_{\mathrm{PL}}(\beta|\vec{\theta},V)}{\frac{\mathbb{1}_{\gamma\in \mathrm{Ext}(V)}}{\prod_{t=1}^{m-1} Z_{V,\gamma^{(t)},\vec{\theta}}}\cdot \Pr_{\mathrm{PL}}(\gamma|\vec{\theta},V)}\\
=&\prod_{t=1}^{m-1}\frac{Z_{V,\gamma^{(t)},\vec{\theta}}}{Z_{V,\beta^{(t)},\vec{\theta}}},
\end{aligned}$$

where all normalization factors, $Z_{V,\beta^{(t)},\vec{\theta}}$ and $Z_{V,\gamma^{(t)},\vec{\theta}}$ are already calculated in GRIM process. □

We now prove that the mixing time of $\boldsymbol{\mathcal{M}}_{\mathrm{PL}}$ is exponentially upper bounded for a large class of profiles, called layered-structure profiles, defined as follows. Intuitively, alternatives in such a partial order can be partitioned into multiple layers, such that each layer contains alternatives among which no comparisons are available, and there is a linear order over layers such that alternatives in a high-ranked layer is always preferred to alternatives in a low-ranked layer.

Definition 4 (Layered-Structure Preferences)**.** *We call agent j's preference is layered-structured if and only if there exist non-overlapping sets of alternatives $\mathcal{C}_1,\cdots,\mathcal{C}_l$ such that:*
1° *For any $l'\in[l]$ and any two alternatives c_{i_1}, $c_{i_2}\in\mathcal{C}_{l'}$, no preference between c_{i_1}, c_{i_2} are given by agent j.*
2° *For any two alternatives $c_{i_1}\in\mathcal{C}_{l_1}$ and $c_{i_2}\in\mathcal{C}_{l_2}$, $c_{i_1}\succ_j c_{i_2}$ if and only if $l_1<l_2$.*
3° $\cup_{l'\in[l]}\mathcal{C}_{l'}=\mathcal{ES}_j$.

Theorem 5. *For layered-structured preferences, the mixing time of Markov Chain $\boldsymbol{\mathcal{M}}_{PL}$ is $O(\eta^{m^{*2}}\ln\epsilon^{-1})$, where $\eta=\frac{\max_{i\in[m]}\theta_i}{\min_{i\in[m]}\theta_i}$ and $m^*=|\mathcal{ES}_i|$.*

Proof. Since the proposal state γ is independent to current state β, we have the following Lemma:

Lemma 5. *[Example 2 of [56]] The mixing time of $\boldsymbol{\mathcal{M}}_{PL}$ is $O\left(s_{\max}\ln\epsilon^{-1}\right)$, where $s_{\max}=\max_{\gamma\in Ext(V_i)}\frac{\Pr_{PL}(\gamma|\vec{\theta})}{\Pr_{GRIM}(\gamma|\vec{\theta})}$.*

Armed with Lemma 5, we only need to provide an upper bound of $s_{\max}$ (shown in Lemma 6).

Lemma 6. *For any layer-structured preference V, $s_{\max} = \max_{\gamma \in Ext(V_i)} \frac{\Pr_{PL}(\gamma|\vec{\theta})}{\Pr_{GRIM}(\gamma|\vec{\theta})} \leq \eta^{m^{*2}}$.*

Proof. We use $m_{l'}$ to denote the number of alternatives in $\mathcal{C}_{l'}$ and we define

$$s_{\min} = \min_{\gamma \in \mathrm{Ext}(V_i)} \frac{\Pr_{\mathrm{PL}}(\gamma|\vec{\theta})}{\Pr_{\mathrm{GRIM}}(\gamma|\vec{\theta})},$$

which is similar to $s_{\min}$. Since $s_{\max}$ and $s_{\min}$ are defined as the maximum (minimum) ratio between the probabilities of GRIM and Plackett-Luce, we have $s_{\max} \geq 1 \geq s_{\min}$. According to Proposition 1, we have,

$$s_{\max} \leq \frac{s_{\max}}{s_{\min}} = \max_{\gamma,\beta \in \mathrm{Ext}(V)} \left(\prod_{t=1}^{m-1} \frac{Z_{V,\gamma^{(t)},\vec{\theta}}}{Z_{V,\beta^{(t)},\vec{\theta}}} \right). \tag{3.7}$$

Inequality (3.7) reduce the problem to bounding GRIM normalization factors. Next, we will focus on layer-structure preferences and show the their normalization factors are bounded as:

$$\frac{Z_{V,\gamma^{(t-1)},\vec{\theta}}}{Z_{V,\beta^{(t-1)},\vec{\theta}}} \leq \begin{cases} 1 & t < m_1 \\ \eta^t & t \geq m_1 \end{cases}, \tag{3.8}$$

where $\gamma, \beta \in \mathrm{Ext}(V)$ are two extensions of V.

W.l.o.g., we let the insertion order in GRIM follow the same order of $\mathcal{C}_1 \to \cdots \to \mathcal{C}_l$. Recalling the justification of RIM algorithm, we know GRIM is a perfect sampler on $\mathcal{C}_1$ since all alternatives in $\mathcal{C}_1$ can be inserted to any vacancies and the case of $t < m_1$ follows.

Recalling the definition of $Z_{V,\gamma^{(t-1)},\vec{\theta}}$, alternative $c_t \in C_{l'}$'s normalization factor $Z_{V,\gamma^{(t-1)},\vec{\theta}}$ is $\sum_{i=1}^{t} \mathbb{1}_{\gamma_i^{(t)} \in \mathrm{Ext}\left(\gamma^{(t-1)}\right)} \cdot \frac{\Pr_{\mathrm{PL}}\left(\gamma_i^{(t)} \middle| (\theta_{1,1},\cdots,\theta_{1,m_1},\cdots,\theta_{l',1},\cdots,\theta_{l',t-(m_1+\cdots+m_{l'-1})})\right)}{\Pr_{\mathrm{PL}}\left(\gamma_i^{(t)} \middle| (\theta_{1,1},\cdots,\theta_{1,m_1},\cdots,\theta_{l',1},\cdots,\theta_{l',t-(m_1+\cdots+m_{l'-1})-1})\right)}$, where $\theta_{i,j}$ represent the j-th inserted alternative in C_i. Since all alternatives in $C_{l'}$ are strictly less preferred to alternatives in $C_1, \cdots, C_{l'-1}$, only the last $t - \sum_{i \in [l'-1]} m_i$ vacancies are allowed by linear

extension condition, and we have,

$$
\begin{aligned}
&Z_{V,\gamma^{(t-1)},\vec{\theta}} \\
&= \sum_{i=m_1}^{t} \frac{\Pr_{\mathrm{PL}}\left(\gamma_i^{(t)} \middle| (\theta_{1,1},\cdots,\theta_{1,m_1},\cdots,\theta_{l',1},\cdots,\theta_{l',t-(m_1+\cdots+m_{l'-1})})\right)}{\Pr_{\mathrm{PL}}\left(\gamma_i^{(t)} \middle| (\theta_{1,1},\cdots,\theta_{1,m_1},\cdots,\theta_{l',1},\cdots,\theta_{l',t-(m_1+\cdots+m_{l'-1})-1})\right)} \\
&= \left(\prod_{j=1}^{\sum_{i\in[l'-1]} m_i} \frac{\theta_{\gamma_j}}{\theta_t + \sum_{i'=j}^{t-1}\theta_{\gamma_{i'}}}\right) \cdot \left(\prod_{j=1+\sum_{i\in[l'-1]} m_i}^{t-1} \frac{\theta_{\gamma_j}}{\sum_{i'=j}^{t-1}\theta_{\gamma_{i'}}}\right),
\end{aligned}
$$

where γ_j denote the $j-$th ranked alternative in $\gamma^{(t-1)}$. Doing the same deviation to ranking $\beta^{(t-1)}$ and combine it with the result of γ, for any alternative $c_t \in C_{l'}$ we have,

$$
\begin{aligned}
&\frac{Z_{V,\gamma^{(t-1)},\vec{\theta}}}{Z_{V,\beta^{(t-1)},\vec{\theta}}} \\
&= \left(\prod_{j=1}^{\sum_{i\in[l'-1]} m_i} \frac{\theta_t + \sum_{i'=j}^{t-1}\theta_{\beta_{i'}}}{\theta_t + \sum_{i'=j}^{t-1}\theta_{\gamma_{i'}}}\right) \cdot \left(\prod_{j=1+\sum_{i\in[l'-1]} m_i}^{t-1} \frac{\sum_{i'=j}^{t-1}\theta_{\beta_{i'}}}{\sum_{i'=j}^{t-1}\theta_{\gamma_{i'}}}\right) \\
&\leq \left(\prod_{j=1}^{\sum_{i\in[l'-1]} m_i} \frac{\theta_{\max}}{\theta_{\min}}\right) \cdot \left(\prod_{j=1+\sum_{i\in[l'-1]} m_i}^{t-1} \frac{\theta_{\max}}{\theta_{\min}}\right) \leq \eta^t.
\end{aligned}
$$

Lemma 6 follows by combining (3.8) and (3.7). □

Theorem 5 follows by applying Lemma 6 to Lemma 5. □

3.4.2 The Gibbs Sampler for Plackett-Luce Model

We first introduce one exact Plackett-Luce sampler for full rankings using random utility interpretation (Theorem 6). It is noteworthy that the complexity of this sampler is $O(m)$ (or equivalently, the complexity of one Gibbs iteration in Algorithm 6 is $O(m^*)$).

Theorem 6. *[51, Theorem 5] For sample space $\mathcal{S} = \mathcal{L}(\mathcal{C})$ and parameters $\vec{\theta} = (\theta_1, \cdots, \theta_m)$, Plackett-Luce sampling is equivalent to sample from Gumbel distributions whose cumulative distribution functions (CDFs) are $e^{-\theta_i \cdot e^{-x}}$.*

Our Gibbs Plackett-Luce sampler given arbitrary partial orders (Algorithm 6) is based on the PL sampler introduced above and the Gibbs sampler by [41]. We let $l_U = 1$ (or $l_L = 0$)

Algorithm 6: Gibbs Plackett-Luce Sampler

Input: Plackett-Luce parameter $\vec{\theta} = (\theta_1, \cdots, \theta_m)$, partial order V_j and number of iterations N.
Initialization: Arbitrarily set an $1 \times m$ array of utilities $u = (u_1, \cdots, u_m)$ and set of inserted alternatives $S = \emptyset$.
for $n = 1$; $n \leq N$ **do**
  for $c_t \in \mathcal{ES}_j$ **do**
    Calculate $l_U = \min_{i \in U_i} e^{-\theta_t \cdot e^{-u_i}}$, where $U_i = \{i \mid (c_i \in S) \wedge [c_i \succ_j c_t] \in V_j\}$.
    Calculate $l_L = \max_{i \in L_i} e^{-\theta_t \cdot e^{-u_i}}$, where $L_i = \{i \mid (c_i \in S) \wedge [c_t \succ_j c_i] \in V_j\}$.
    Uniformly generate a random number r from interval $(\max\{0, l_L\}, \min\{1, l_U\})$.
    Let $u_t = -\ln(-\ln r + \ln \theta_t)$ and add c_t to S.
  end
end
Sample the remaining alternatives' utilities according to Theorem 6.
Sort the utilities $(u_1, \cdots, u_m)$ to get the linear order R.
return R.

when $U_i = \emptyset$ (or $L_i = \emptyset$). We note that the sampler shown in Algorithm 6 is applicable to any Random Utility Models with given CDF on utility distributions.

Remark for Gibbs Sampler Bounding the mixing time of Gibbs sampler for arbitrary (truncated) distribution is still an open question. Moreover, the mixing time of Algorithm 6 is even harder to bound because it focuses on the distribution over rankings instead of utilities. Nevertheless, Gibbs sampler might be a fast algorithm in experiments (or, averaged case) because it takes the advantage of lower complexity per iteration.

3.5 Experiments

We illustrate the performances of our algorithms on both synthetic data and real-world data. We note that there is no existing algorithm for learning Plackett-Luce model and its mixtures from general partial orders without discarding any information. All experiments with recorded runtime were run on an Ubuntu Linux server with Intel Xeon E5 v3 CPUs each clocked at 3.50 GHz.

3.5.1 Experiments on Synthetic Data

Learning the Plackett-Luce Model from Partial Orders. We first generated linear orders from the Plackett-Luce model over 10 alternatives. The ground truth parameters $\vec{\theta} = (\theta_1, \ldots, \theta_m)$ were generated uniformly at random and normalized s.t. $\sum_{i=1}^{m} \theta_i = 1$.

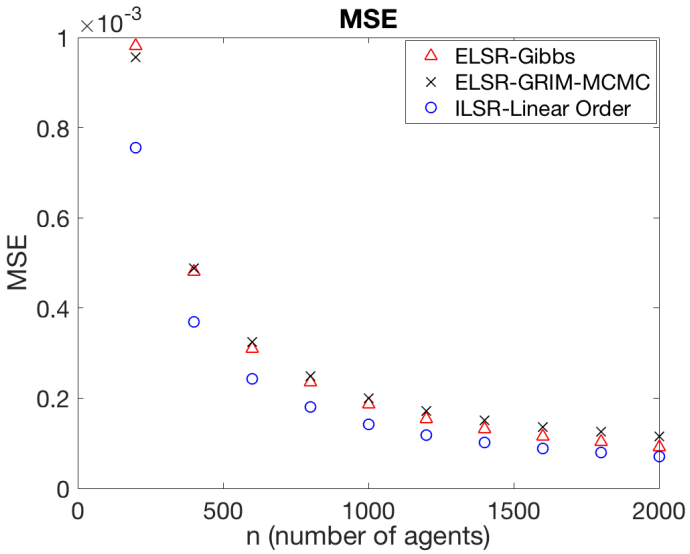

Figure 3.3: MSE of the proposed EM algorithms for a single Plackett-Luce model.

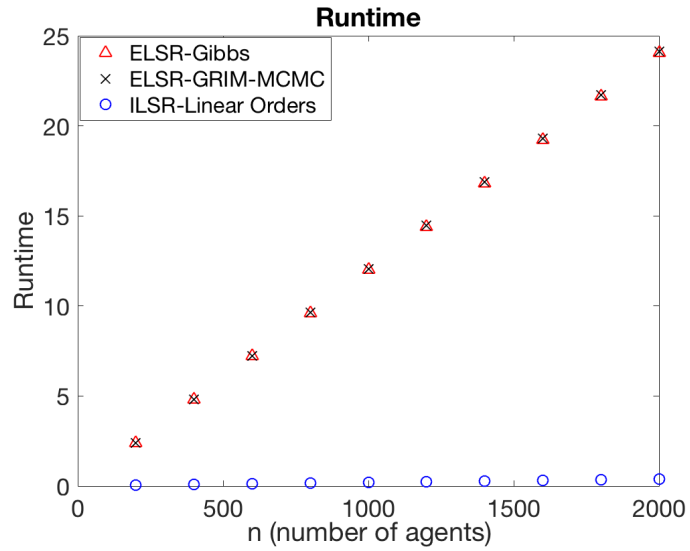

Figure 3.4: Runtime (in seconds) of the proposed EM algorithms for a single Plackett-Luce model. Note ILSR-Linear Orders is not applicable to arbitrary partial preferences.

The linear orders were sampled by Definition 2. Then we sampled partial orders in the following way. We fixed $p = 0.5$. Given any linear order, we sampled from all $\frac{m(m-1)}{2}$ pairwise comparisons by keeping each pairwise comparison with probability p independently. The sampled partial order is then converted to its transitive closure. In average, below 75% of all pairwise comparisons were sampled. We ran our proposed EM algorithms with GRIM-MCMC and Gibbs samplers respectively, denoted by "ELSR-GRIM-MCMC" and "ELSR-

Gibbs". Two linear extensions were sampled for each partial order. We use five EM iterations for each algorithm. Values were averaged over 2000 trials. To show the statistical efficiency of our algorithms, we also learned from linear orders (which are assumed to be unobservable) using the ILSR algorithm (denoted by "ILSR-Linear Order") by [25], which is the iterative LSR. Results are shown in Figures 3.3 and 3.4. **Observations.** Both statistical efficiency and computational efficiency of the algorithm with Gibbs sampler are slightly better than that with GRIM sampler. The statistical efficiency of both our algorithms are close to ILSR, which learns from supposedly unobservable linear orders.

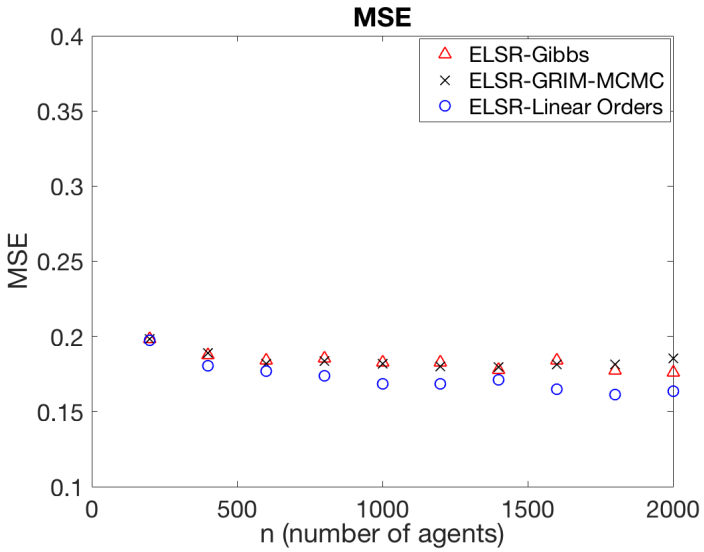

Figure 3.5: MSE of the proposed EM algorithms for mixtures of 3 Plackett-Luce models.

Learning Mixtures of Plackett-Luce Models. The setup is similar to the previous experiments. We generated the ground truth parameters for 3-PL over 6 alternatives by generating the mixing coefficients and the parameter of each component uniformly at random and normalizing s.t. $\sum_{r=1}^{k} \alpha_r = 1$ and for all r, $\sum_{i=1}^{m} \theta_i^{(r)} = 1$. Then we sampled partial orders using exactly the same way with a fixed $p = 0.5$. On average, below 65% of all pairwise comparisons were sampled due to fewer alternatives. 6 linear extensions were generated from each partial order. We use five EM iterations for each algorithm. Values were averaged over 2000 trials. **Observations.** From Figures 3.5 and 3.6 we observe that our algorithm with Gibbs sampler is more computationally efficient than that with GRIM-MCMC sampler. Again, both algorithms have similar statistical efficiency "ELSR-Linear Order", which learns from supposedly unobservable linear orders.

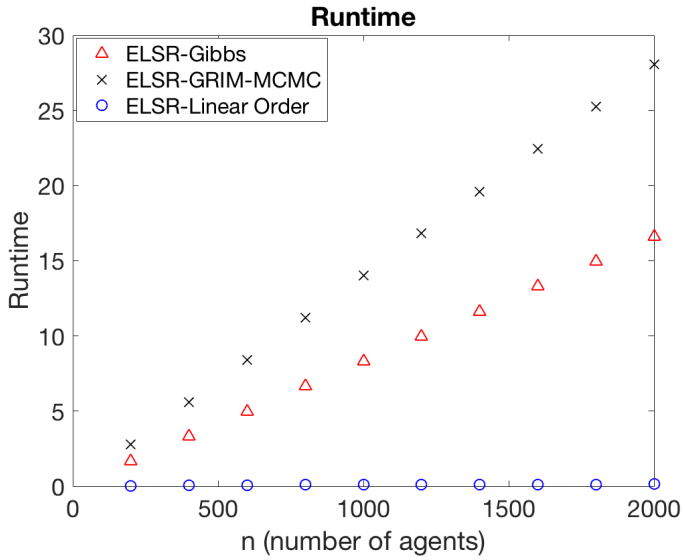

Figure 3.6: Runtime (in seconds) of the proposed EM algorithms for mixtures of 3 Plackett-Luce models.

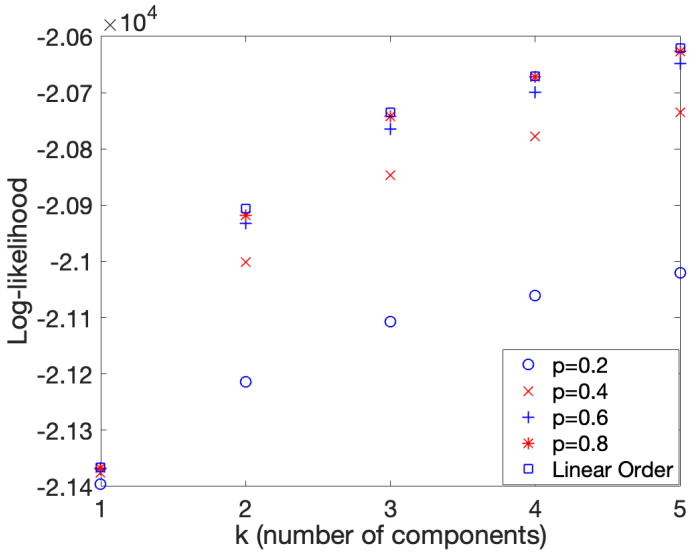

Figure 3.7: Log-likelihood of sushi test data as k increases when a k-PL is learned from the training data. Values were averaged over 100 trials.

3.5.2 Experiments on Real-World Data

The sushi data from Preflib [57] consists of 5000 linear orders of 10 types of sushi. We randomly split this dataset into training data (3500 linear orders) and test data (1500 partial orders). To generate partial orders, we sample pairwise comparisons with $p = 0.2, 0.4, 0.6, 0.8, 1$ and learn a k-PL from the partial orders using the proposed E-LSR algorithm with Gibbs sampler, where k ranges from 1 to 5. We measure the fitness of a k-PL using the likelihood

of the test data given the learned parameter. Results are shown in Figure 3.7.

Observations. We observe that when p increases, the learning outcome fits the test data better. For $k \leq 5$, k-PL fits the test data better as k increases.

3.6 Conclusions and Future Work

We propose an EM-based framework with two samplers for learning Plackett-Luce model and its mixtures from partial orders. We demonstrate the efficiency of our algorithms with experiments on both synthetic datasets and real-world data and conclude that Gibbs sampler outperforms GRIM-MCMC. For future work we plan to explore faster algorithms to sample linear extensions from partial orders or other approaches to learn mixtures of Plackett-Luce models, as well as other mixture models.

CHAPTER 4
SMOOTHED DIFFERENTIAL PRIVACY

4.1 Introduction

Differential privacy (DP), a *de facto* measure of privacy in academia and industry, is often achieved by adding external noises to published information [58]. However, external noises are procedurally or practically unacceptable in many real-world applications. For example, presidential elections often require a deterministic rule to be used [59]. In such cases, though, *internal noise* often exists, as shown in the following example.

Example 3 (**Election with Internal Noise).** *Due to COVID-19, many voters in the 2020 US presidential election chose to submit their votes by mail. Unfortunately, it was estimated that the US postal service might have lost up to 300,000 mail-in ballots (*0.2% *of all votes) [60]. For the purpose of illustration, suppose these votes are distributed uniformly at random, and the histogram of votes is announced after the election day.*

A critical public concern about elections is: should publishing the histogram of votes be viewed as a significant threat to privacy? Notice that with internal noise such as in Example 3, the (sampling) histogram mechanism[2] can be viewed as a randomized mechanism, formally called *sampling-histogram* in this chapter. The same question can be asked about publishing the winner under a deterministic voting rule, where the input is sampled from the real votes.

If we apply DP to answer this question, we would then conclude that publishing the histogram *can* poses a significant threat to privacy, as the privacy parameter $\delta \approx 1$ (See Section 4.2 for the formal definition) in the following worst-case scenario: all except one vote are for the Republican candidate, and there is one vote for the Democratic candidate. Notice that $\delta \approx 1$ is much worse than the threshold for private mechanisms, $\delta = o(1/n)$ (n is the number of agents). Moreover, using the adversary's utility as the measure of privacy loss (see Section 4.2 for the formal definition), in this (worst) case, the privacy loss is large (≈ 1, see Section 4.2 for the formal definition of utility), which means the adversary can make accurate predictions about every agent's preferences.

Portions of this chapter have previously appeared as: A. Liu, Y. Wang, and L. Xia. "Smoothed differential privacy." 2021, *arXiv:2107.01559*.

[2]Throughout this chapter, the sampling process is always treated as one step inside of the mechanism.

However, DP does not tell us whether publishing the histogram poses a significant threat to privacy *in general*. In particular, the worst-case scenario described in the previous paragraph never happened even approximately in the modern history of US presidential elections. In fact, no candidates get more than 70% of the votes since 1920 [61], when the progressive party dissolved.

It turns out the privacy loss (measured by the adversary's utility) may not be as high as measured by DP. To see this, suppose 0.2% of the votes were randomly lost in the presidential elections of each year since 1920 (in light of Example 3), we present the adversary's utility of predicting the unknown votes in Figure 4.1. It can be seen that the adversary has very limited utility (at the order of 10^{-32} to 10^{-8}, which is always smaller than the threshold of private mechanisms n^{-1}), meaning that the adversary cannot learn much from the published histogram of votes. We also observe an interesting decreasing trend in δ, which implies that the elections become more private in more recent years. This is primarily due to the growth of voting population, which is exponentially related to the adversary's utility (Theorem 8). In Appendix B.1, we show that the elections are still private when only 0.01% of votes got lost.

As another example, for *deep neural networks* (DNNs), even adding slight noise can lead to dramatic decreases in the prediction accuracy, especially when predicting underrepresented classes [62]. Internal noises also widely exist in machine learning, for example, in the standard practice of cross-validation as well as in training (e.g., batch-sampling when training DNNs).

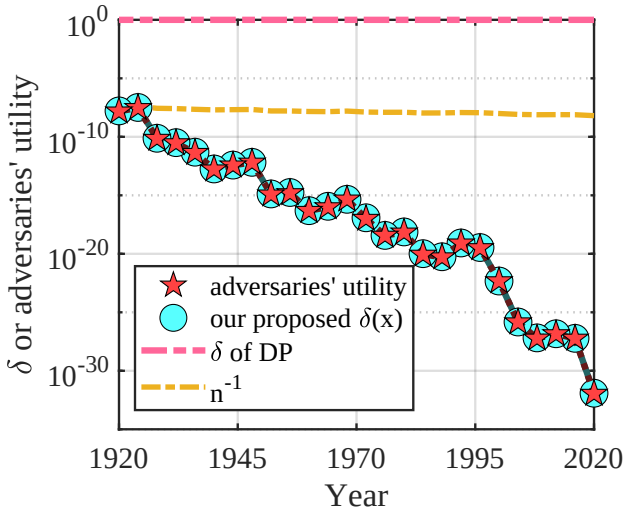

Figure 4.1: The privacy loss in US presidential elections. The lower δ is, the more private the election is. The rigid definition of adversaries' utility can be found in Equation (B.4) of Appendix B.4.

As shown in these examples, the worst-case privacy according to DP might be too loose to serve as a practical measure for evaluating and comparing mechanisms without external noises in real-world applications. This motivates us to ask the following question.

How can we measure privacy for mechanisms without external noise under realistic models?

The choice of model is critical and highly challenging. A model based on worst-case analysis (such as in DP) provides upper bounds on privacy loss, but as we have seen in Figure 4.1, in some situations, such upper bounds are too loose to be informative in practice. This is similar to the runtime analysis of an algorithm—an algorithm with exponential worst-case runtime, such as the simplex algorithm, can be faster than some algorithms with polynomial runtime in practice.

Average-case analysis is a natural choice of the model, but since "*all models are wrong*" [63], any privacy measure designed for a certain distribution over data may not work well for other distributions. Moreover, ideally, the new measure should satisfy the desirable properties that played a central role behind the success of DP, including *composability* and *robustness to post-processing*. These properties make it easier for the mechanism designers to figure out the privacy level of mechanisms. Unfortunately, we are not aware of a measure based on average-case analysis that has these properties.

We believe that the *smoothed analysis* [64] provides a promising framework for addressing this question. Smoothed analysis is an extension and combination of worst-case and average-case analyses that inherits the advantages of both. It measures the expected performance of algorithms under slight random perturbations of worst-case inputs. Compared with the average-case analysis, the assumptions of the smoothed analysis are much more natural. Compared with the worst-case analysis, the smoothed analysis can better describe the real-world performance of algorithms. For example, it successfully explained why the simplex algorithm is faster than some polynomial algorithms in practice.

Our Contributions. The main merit of this chapter is a new notion of privacy for mechanisms without external noises, called *smoothed differential privacy* (*smoothed DP* for short), which applies smoothed analysis to the privacy parameter $\delta(x)$ (Definition 6) as a function of the database x. In our model, the "ground truth" distribution of agents is from a set of distributions Π over data points, on top of which the nature adds random noises. Formally, the "smoothed" $\delta(x)$ is defined as

$$\delta_{\text{SDP}} \triangleq \max_{\vec{\pi}} \left(\mathbb{E}_{x\sim\vec{\pi}} \left[\delta(x)\right] \right), \tag{4.1}$$

where $x \sim \vec{\pi} = (\pi_1, \cdots, \pi_n) \in \Pi^n$ means that for every $1 \leq i \leq n$, the i-th entry in the database independently follows the distribution π_i. We note that Π is a parameter for smoothed analysis, not for the mechanisms $\mathcal{M}$. At a high level, we compare the smoothed DP notion and other DP(-like) notions in Table 4.1.

Table 4.1: The comparison between smoothed DP and other privacy notions.

<table>
<tr><th>Notions</th><th>database x</th><th>output $\mathcal{S}$</th><th>adjacent database x'</th></tr>
<tr><td>Smoothed DP (this chapter)</td><td>smoothed analysis</td><td rowspan="2">worst-case analysis</td><td rowspan="5">worst-case analysis</td></tr>
<tr><td>DP [65]</td><td rowspan="5">worst-case analysis</td></tr>
<tr><td>Gaussian DP or f-DP [66]</td></tr>
<tr><td>Rényi DP [67]</td><td>average-case analysis (weighted by Rényi divergence or sub-Gaussian divergence)</td></tr>
<tr><td>Concentrated DP [68], [69]</td></tr>
<tr><td>Bayesian DP [70]</td><td rowspan="3">worst-case analysis</td><td>less informative adversary</td></tr>
<tr><td>Random DP [71]</td><td rowspan="2">less informative adversary</td></tr>
<tr><td>Distributional DP [72]</td><td>worst-case analysis</td></tr>
</table>

Theoretically, we prove that smoothed DP satisfies many desirable properties, includ-

ing two properties also satisfied by the standard DP: *robustness to post-processing* (Proposition 2) and *composability* (Proposition 3). In addition, we prove two unique properties for smoothed DP, called *pre-processing* (Proposition 4) and *distribution reduction* (Proposition 5),which makes it easier for the mechanism designer to figure out the privacy level when the set of distributions Π is hard to estimate. Using smoothed DP, we found that many discrete mechanisms without external noise (and with small internal noise) are significantly more private than those guaranteed by DP. For example, the sampling-histogram mechanism in Example 3 has an exponentially small δ_{SDP} (Theorem 8), which implies that the mechanism protects voters' privacy in elections—and this is in accordance with the observation on US election data in Figure 4.1. We also note that the sampling-histogram mechanism is widely used in machine learning (e.g., the SGD in quantized DNNs). In comparison, smoothed DP implies a similar privacy level as the standard DP in many continuous mechanisms. We proved that smoothed DP and the standard DP have the same privacy level for the widely-used sampling-average mechanism when the inputs are continuous (Theorem 9).

Experimentally, we numerically evaluate the privacy level of the sampling-histogram mechanism using US presidential election data. Simulation results show an exponentially small δ_{SDP}, which is in accordance with our Theorem 8. Our second experiment shows that a one-step *stochastic gradient descent* (SGD) in quantized DNNs [73], [74] also has an exponentially small δ_{SDP}. This result implies that SGD with gradient quantization can already be private in practice, without adding any external noise. In comparison, the standard DP notion always requires extra (external) noise to make the network private at the cost of a significant reduction in accuracy.

Related Work and Discussions. There is a large body of literature on the theory and practice of DP and its extensions. We believe that the smoothed DP introduced in this chapter is novel. To the best of our knowledge, none of the literature has proposed any DP-like notions for the mechanism without external noises. A notable exception is distributional DP [72], which considers a less informative adversary to provide a privacy measure for deterministic mechanisms. However, since distribution DP does not require any randomness in the mechanism, its privacy guarantee is much weaker than other DP-like notions. Rényi DP [67], Gaussian DP [66] and Concentrated DP [68], [69] target to provide tighter privacy bounds for the adaptive mechanisms. Those three notions generalized the (ϵ, δ) measure of distance between distributions to other divergence measures. Bayesian DP [70] tries to

provide an "affordable" measure of privacy that requires less external noise than DP. With similar objectives, Bun and Steinke [75] add noises according to the average sensitivity instead of the worst-case sensitivity required by DP. However, external noises are required in [75] and [70]. Random DP [71] combines the high-level idea of distributional DP and Bayesian DP, which considers randomness in both the database x and its neighboring database x'.

The smoothed analysis [76] is a widely-accepted analysis tool in mathematical programming, machine learning [77]–[79], computational social choice [80]–[83], and other topics [84]–[86]. Lastly, we note that smoothed DP is very different from the smooth sensitivity framework [87]. The latter is an algorithmic tool that smooths the changes of the noise level across neighboring databases (and achieves the standard DP), while we use smoothing as a theoretical tool to analyze the intrinsic privacy properties of non-randomized algorithms in practice.

Appendix B.2 discusses the related works in the field of quantized neural networks.

4.2 Differential Privacy and Its Interpretations

In this chapter, we use n to denote the number of records (entries) in a database $x \in \mathcal{X}^n$, where $\mathcal{X}$ denotes all possible values for a single entry. n also represents the number of agents when one individual can only contribute one record. Let $||x - x'||_1$ denote the number of different records (the ℓ_1 distance) between database x and x'. We say that two databases are *neighboring*, if they contain no more than one different entry.

Definition 5 (**Differential privacy**). *Let $\mathcal{M}$ denote a randomized algorithm and $\mathcal{S}$ be a subset of the image space of $\mathcal{M}$. $\mathcal{M}$ is said to be (ϵ, δ)-differentially private for some $\epsilon > 0, \delta > 0$, if for any $\mathcal{S}$ and any pair of inputs x and x' such that $||x - x'||_1 \leq 1$,*

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{M}(x') \in \mathcal{S}] + \delta, \tag{4.2}$$

Notice that the randomness comes from the mechanism $\mathcal{M}$ in the worst case x.

DP guarantees immunity to many kinds of attacks (e.g., *linkage attacks* [88] and *reconstruction attacks* [58]). Take *reconstruction attacks* for example, the adversary has access to a subset of the database (such information may come from public databases, social media, etc.). In an extreme situation, an adversary knows all but one agent's records. To protect the data of every single agent, DP uses $\delta = o(1/n)$ as a common requirement of private

mechanisms [58, p. 18]. To meet this requirement, a private mechanism (under DP notion) usually[3] need to include external noises (e.g., Gaussian noise, Laplacian noise, and the noise, which usually is discrete, in exponential mechanisms). Next, we recall two common interpretations/justifications on how DP helps protect privacy even in the extreme situation of reconstruction attacks.

Justification 1: DP guarantees the prediction of adversaries cannot be too accurate [90], [91]. Assume that the adversary knows all entries except the i-th. Let x_{-i} denote the database x with its i-th entry removed. With the information provided by the output $\mathcal{M}(x)$, the adversary can infer the missing entry by testing the following two hypotheses:

$\mathcal{H}_0$: The missing entry is X (or equivalently, the database is $x = x_{-i} \cup \{X\}$).

$\mathcal{H}_1$: The missing entry is X' (or equivalently, the database is $x' = x_{-i} \cup \{X'\}$).

Suppose that after observing the output of $\mathcal{M}$, the adversary uses a rejection region rule for hypothesis testing[4], where $\mathcal{H}_0$ is rejected if and only if the output is in the rejection region $\mathcal{S}$. For any fixed $\mathcal{S}$, the decision rule can be wrong in two possible ways, false positive (Type I error) and false negative (Type II error). Thus, the Type I error rate is $\text{Error}_{\text{I}}(x) = \Pr[\mathcal{M}(x) \in \mathcal{S}]$. Similarly, the Type II error rate is $\text{Error}_{\text{II}}(x') = \Pr[\mathcal{M}(x') \notin \mathcal{S}] = 1 - \Pr[\mathcal{M}(x') \in \mathcal{S}]$. According to the definition of DP, for any neighboring x and x', the adversary always has

$$e^{\epsilon} \cdot \text{Error}_{\text{I}}(x) + \text{Error}_{\text{II}}(x') \geq 1 - \delta \quad \text{and} \quad e^{\epsilon} \cdot \text{Error}_{\text{II}}(x') + \text{Error}_{\text{I}}(x) \geq 1 - \delta, \tag{4.3}$$

which implies that $\text{Error}_{\text{I}}(x)$ and $\text{Error}_{\text{II}}(x')$ cannot be small at the same time. When ϵ and δ are both small, both Error_{I} and Error_{II} becomes close to 0.5 (the error rates of random guess), which means that the adversary cannot get much information from the output of $\mathcal{M}$.

Justification 2: With probability at least 1-2δ, $\mathcal{M}$ is insensitive to the change of one record [58].

In more detail, (ϵ, δ)-DP guarantees the distribution of $\mathcal{M}$'s output will not change signifi-

[3]Some differentially private mechanisms such as "stability-based query release" appear to not require external noise with high probability if the local sensitivity is 0 for the input database and for all other databases that differ by at most $\ln(1/\delta)/\epsilon$ data points [89] . Note that in our case (such as the motivating example), the local sensitivity is not 0. Also, the "stability-based" methods still need randomization (though not adding noise).

[4]The adversary can use any decision rule, and the rejection region rule is adopted just for example.

cantly when one record changes. Here, "change" corresponds to add, remove or replace one record of the database. Mathematically, [58, p. 18] showed that given any pair of neighboring databases x and x',

$$\Pr_{a\sim\mathcal{M}(x)}\left[e^{-\epsilon} \leq \frac{\Pr[\mathcal{M}(x)=a]}{\Pr[\mathcal{M}(x')=a]} \leq e^{\epsilon}\right] \geq 1-2\delta. \tag{4.4}$$

where the probability [5] is taken over a (the output of $\mathcal{M}$). The above inequality shows that the change of one record cannot make an output significantly more likely or significantly less likely (with at least $1-2\delta$ probability). Dwork [58, p. 25] also claimed that the above formula guarantees that the adversary cannot learn too much information about any single record of the database through observing the output of $\mathcal{M}$.

4.3 Smoothed Differential Privacy

Recall that DP is based on the worst-case analysis over all possible databases. However, as shown in Exampe 3 and Figure 4.1, the worst-case nature of DP sometimes leads to an overly pessimistic measure of privacy loss, which may bring unnecessary external noise in the hope of improved privacy but at the cost of accuracy. In this section, we introduce smoothed DP, which applies the smoothed analysis to the privacy parameter $\delta(x)$, and prove its desirable properties. Due to the space constraint, all proofs of this section can be found in Appendix B.5.

4.3.1 The Database-Wise Privacy Parameter

We first introduce the database-wise parameter $\delta_{\epsilon,\mathcal{M}}(x)$, which measures the privacy leakage of mechanism $\mathcal{M}$ when its input is x.

Definition 6 (**Database-wise privacy parameter $\boldsymbol{\delta_{\epsilon,\mathcal{M}}(x)}$**). *Let $\mathcal{M}:\mathcal{X}^n\to\mathcal{A}$ denote a randomized mechanism. Given any database $x\in\mathcal{X}^n$ and any $\epsilon>0$, define the database-wise privacy parameter as:*

$$\delta_{\epsilon,\mathcal{M}}(x) \triangleq \max\Big(0,\ \max_{x':||x-x'||_1\leq 1}\big(d_{\epsilon,\mathcal{M}}(x,x')\big),\ \max_{x':||x-x'||_1\leq 1}\big(d_{\epsilon,\mathcal{M}}(x',x)\big)\Big), \tag{4.5}$$

where $d_{\epsilon,\mathcal{M}}(x,x') = \max_{\mathcal{S}}\left(\Pr\left[\mathcal{M}(x)\in\mathcal{S}\right] - e^{\epsilon}\cdot\Pr\left[\mathcal{M}(x')\in\mathcal{S}\right]\right)$.

[5] The probability notion used in [58] is different from the standard definition of probability. See Appendix B.3 for formal descriptions.

In words, $\delta_{\epsilon,\mathcal{M}}(x)$ is the minimum δ values, such that the (ϵ,δ)-DP requirement on $\mathcal{M}$ (Inequality (4.2)) holds at neighboring pairs (x,x') and (x',x). In Lemma 13, we will show that $d_{\epsilon,\mathcal{M}}(x,x')$ measures the utility of the adversary.

DP as the worst-case analysis of $\delta_{\epsilon,\mathcal{M}}(x)$. In the next theorem, we show that the privacy measure based on the worst-case analysis of $\delta_{\epsilon,\mathcal{M}}$ is equivalent to the standard DP.

Theorem 7 (**DP in $\delta_{\epsilon,\mathcal{M}}(x)$**). *Mechanism $\mathcal{M}:\mathcal{X}^n\to\mathcal{A}$ is (ϵ,δ)-differentially private if and only if,*

$$\max_{x\in\mathcal{X}^n}\Big(\delta_{\epsilon,\mathcal{M}}(x)\Big)\le\delta.$$

4.3.2 Formal Definition of Smoothed DP

With the database-wise privacy parameter $\delta_{\epsilon,\mathcal{M}}(x)$ defined in the last subsection, we now formally define smoothed DP, where the worst-case "ground truth" distribution of every agent is allowed to be any distribution from a set Π, on top of which Nature adds random noises to generate the database. We would like to note again that Π is a parameter for smoothed analysis, not for the mechanisms $\mathcal{M}$.

Definition 7 (**Smoothed DP**). *Let Π be a set of distributions over $\mathcal{X}$. We say $\mathcal{M}:\mathcal{X}^n\to\mathcal{A}$ is (ϵ,δ,Π)-smoothed differentially private if,*

$$\max_{\vec{\pi}\in\Pi^n}\left(\mathbb{E}_{x\sim\vec{\pi}}\left[\delta_{\epsilon,\mathcal{M}}(x)\right]\right)\le\delta, \tag{4.6}$$

where $x\sim\vec{\pi}=(\pi_1,\cdots,\pi_n)$ means that for every $i\in\{1,\cdots,n\}$, the i-th entry in the database follows π_i.

Like DP, smoothed DP bounds privacy leakage (in an arguably more realistic setting), via the following two justifications that are similar to the two common justifications of DP in Section 4.2.

Justification 1: Smoothed DP guarantees the prediction of the adversary cannot be too accurate. Mathematically, a (ϵ,δ,Π)-smoothed DP mechanism $\mathcal{M}$ guarantees

$$\begin{aligned}
&e^{\epsilon}\cdot\max_{\vec{\pi}\in\Pi^n}\Big(\mathop{\mathbb{E}}_{x\sim\vec{\pi}}\left[\mathrm{Error}_{\mathrm{I}}(x)\right]\Big)+\max_{\vec{\pi}\in\Pi^n}\Big(\mathop{\mathbb{E}}_{x\sim\vec{\pi}}\left[\mathrm{Error}_{\mathrm{II}}(x')\right]\Big)\ge 1-\delta\quad\text{and}\\
&e^{\epsilon}\cdot\max_{\vec{\pi}\in\Pi^n}\Big(\mathop{\mathbb{E}}_{x\sim\vec{\pi}}\left[\mathrm{Error}_{\mathrm{II}}(x')\right]\Big)+\max_{\vec{\pi}\in\Pi^n}\Big(\mathop{\mathbb{E}}_{x\sim\vec{\pi}}\left[\mathrm{Error}_{\mathrm{I}}(x)\right]\Big)\ge 1-\delta
\end{aligned} \tag{4.7}$$

The proof follows after bounding Type I and Type II errors when the input is x by $\delta_{\epsilon,\mathcal{M}}$. That is, for a fixed database x, it is not hard to verify

$$\begin{aligned} e^{\epsilon} \cdot \mathrm{Error}_{\mathrm{I}}(x) + \mathrm{Error}_{\mathrm{II}}(x') &\geq 1 - \max_{x':||x-x'||_1 \leq 1} \left(d_{\epsilon,\mathcal{M}}(x, x')\right) \quad \text{and} \\ e^{\epsilon} \cdot \mathrm{Error}_{\mathrm{II}}(x') + \mathrm{Error}_{\mathrm{I}}(x) &\geq 1 - \max_{x':||x-x'||_1 \leq 1} \left(d_{\epsilon,\mathcal{M}}(x', x)\right) \end{aligned} \tag{4.8}$$

Then, by the definition $\delta_{\epsilon,\mathcal{M}}(x)$, we have,

$$\begin{aligned} e^{\epsilon} \cdot \mathrm{Error}_{\mathrm{I}}(x) + \mathrm{Error}_{\mathrm{II}}(x') &\geq 1 - \delta_{\epsilon,\mathcal{M}}(x) \quad \text{and} \\ e^{\epsilon} \cdot \mathrm{Error}_{\mathrm{II}}(x') + \mathrm{Error}_{\mathrm{I}}(x) &\geq 1 - \delta_{\epsilon,\mathcal{M}}(x), \end{aligned} \tag{4.9}$$

which means that $\mathrm{Error}_{\mathrm{I}}$ and $\mathrm{Error}_{\mathrm{II}}$ cannot be small at the same time when the database is x. It follows that the smoothed DP, which is a smoothed analysis of $\delta_{\epsilon,\mathcal{M}}$, can bound the smoothed Type I and Type II errors.

Justification 2: A smoothed DP $\mathcal{M}$ is insensitive to the change of one record with at least 1-2δ probability. Mathematically, a (ϵ, δ, Π)-smoothed DP mechanism $\mathcal{M}$ guarantees

$$\max_{\vec{\pi} \in \Pi^n} \left(\mathop{\mathbb{E}}_{x \sim \vec{\pi}} \left[\Pr_{a \sim \mathcal{M}(x)} \left[e^{-\epsilon} \leq \frac{\Pr[\mathcal{M}(x) = a]}{\Pr[\mathcal{M}(x') = a]} \leq e^{\epsilon} \right] \right] \right) \geq 1 - 2\delta \tag{4.10}$$

The proof is, again, done through analyzing $\delta_{\epsilon,\mathcal{M}}(x)$. More precisely, given any mechanism $\mathcal{M}$, any $\epsilon \in \mathbb{R}_+$ and any pair of neighboring databases x, x', we have

$$\Pr_{a \sim \mathcal{M}(x)} \left[e^{-\epsilon} \leq \frac{\Pr[\mathcal{M}(x) = a]}{\Pr[\mathcal{M}(x') = a]} \leq e^{\epsilon} \right] \geq 1 - 2\delta_{\epsilon,\mathcal{M}}(x). \tag{4.11}$$

The formal claim can be found in Lemma 12 in Appendix B.3.

As smoothed DP replaces the worst-case analysis with smoothed analysis, we also view $\delta = o(1/n)$ as a requirement for private mechanisms for smoothed DP. In addition to the two justifications above, Appendix B.4 justifies the notion of smoothed DP under the view of "adversarial utilities". At a high-level, δ parameter of smoothed DP tightly bounds the adversary's utility in Bayesian predictions under realistic settings.

4.4 Properties of Smoothed DP

First, we present four properties of smoothed DP and discuss how they can help mechanism designers figure out the smoothed DP level of mechanisms. We first present the robustness to *post-processing* property, which says no function can make a mechanism less private without adding extra knowledge about the database. The post-processing property of smoothed DP can be used to upper bound the privacy level of many mechanisms. With it, we know private data preprocessing can guarantee the privacy of the whole mechanism. Then, the rest of the mechanisms can be arbitrarily designed. The proof of all four properties of the smoothed DP can be found in Appendix B.6.

Proposition 2 (**Post-processing**). *Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{A}$ be a (ϵ, δ, Π)-smoothed DP mechanism. For any $f : \mathcal{A} \to \mathcal{A}'$ (which can also be randomized), $f \circ \mathcal{M} : \mathcal{X}^n \to \mathcal{A}'$ is also (ϵ, δ, Π)-smoothed DP.*

Then, we introduce the composition theorem for the smoothed DP, which bounds the smoothed DP property of databases when two or more mechanisms publish their outputs about the same database.

Proposition 3 (**Composition**). *Let $\mathcal{M}_i : \mathcal{X}^n \to \mathcal{A}_i$ be an $(\epsilon_i, \delta_i, \Pi)$-smoothed DP mechanism for any $i \in [k]$. Define $\mathcal{M}_{[k]} : \mathcal{X}^n \to \prod_{i=1}^k \mathcal{A}_i$ as $\mathcal{M}_{[k]}(x) = \big(\mathcal{M}_1(x), \cdots, \mathcal{M}_k(x)\big)$. Then, $\mathcal{M}_{[k]}$ is $\Big(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i, \Pi\Big)$-smoothed DP.*

In practice, Π might be hard to accurately characterize. The next proposition introduces the pre-processing property of smoothed DP, which says the distribution of data can be replaced by the distribution of features (extracted using any deterministic function). For example, in deep learning, the distribution of data can be replaced by the distribution of gradients, which is usually much easier to estimate in real-world training processes.

More technically, the pre-processing property guarantees that any deterministic way of data preprocessing is not harmful to privacy. To simplify notation, we let $f(\pi)$ be the distribution of $f(X)$ where $X \sim \pi$. For any set of distributions $\Pi = \{\pi_1, \cdots, \pi_m\}$, we let $f(\Pi) = \{f(\pi_1), \cdots, f(\pi_m)\}$.

Proposition 4 (**Pre-processing for deterministic functions**). *Let $f : \mathcal{X}^n \to \tilde{\mathcal{X}}^n$ be a deterministic function and $\mathcal{M} : \tilde{\mathcal{X}}^n \to \mathcal{A}$ be a randomized mechanism. Then, $\mathcal{M} \circ f : \mathcal{X}^n \to \mathcal{A}$ is (ϵ, δ, Π)-smoothed DP if $\mathcal{M}$ is $\big(\epsilon, \delta, f(\Pi)\big)$-smoothed DP.*

The following proposition shows that any two sets of distributions with the same convex hull have the same privacy level under smoothed DP. With this, the mechanism designers can ignore all inner points and only consider the vertices of the convex hull when calculating the privacy level of mechanisms. Let $\mathrm{CH}(\Pi)$ denote the convex hull of Π.

Proposition 5 (**Distribution reduction**). *Given any $\epsilon, \delta \in \mathbb{R}_+$ and any Π_1 and Π_2 such that $\mathrm{CH}(\Pi_1) = \mathrm{CH}(\Pi_2)$, an anonymous mechanism $\mathcal{M}$ is $(\epsilon, \delta, \Pi_1)$-smoothed DP if and only if $\mathcal{M}$ is $(\epsilon, \delta, \Pi_2)$-smoothed DP.*

4.5 Smoothed DP as a Measure of Privacy

In this section, we use smoothed DP to measure the privacy of some commonly-used mechanisms, where the randomness is intrinsic and unavoidable (as opposed to external noises such as Gaussian or Laplacian noises). Our analysis focuses on two widely-used algorithms where the intrinsic randomness comes from sampling (without replacement). We also compare the privacy levels of smoothed DP with DP. All missing proofs of this section can be found in Appendix B.7.

4.5.1 Discrete Mechanisms Are More Private Than What DP Predicts

In this subsection, we study the smoothed DP property of (discrete) sampling-histogram mechanism (SHM), which is widely used as a pre-possessing step in many real-world applications like the training of DNNs. As smoothed DP satisfies *post-processing* (Proposition 2) and *pre-processing* (Proposition 3), the smoothed DP property of SHM can upper bound the smoothed DP of many mechanisms used in practice, which are based on SHM.

SHM first sample T data from the database and then output the histogram of the T samples. Formally, we define the sampling-histogram mechanism in Algorithm 7. Note that we require all data in the database to be chosen from a finite set $\mathcal{X}$.

Algorithm 7: Sampling-histogram mechanism $\mathcal{M}_H$

1 **Inputs:** A finite set $\mathcal{X}$, the number of samples $T \in \mathbb{Z}_+$ and a database $x = \{X_1, \cdots, X_n\}$ where $X_i \in \mathcal{X}$ for all $i \in [n]$

2 Randomly sample T data from x without replacement. The sampled data are denoted by $X_{j_1}, \cdots, X_{j_T}$.

3 **Output:** The histogram $\mathrm{hist}(X_{j_1}, \cdots, X_{j_T})$

Smoothed DP of mechanisms based on SHM. The smoothed DP of SHM can be used to upper bound the smoothed DP of the following three groups of mechanisms/algorithms.

The first group consists of deterministic voting rules, as presented in the motivating example in Introduction. The sampling procedure in SHM mimics the votes that got lost.

The second group consists of machine learning algorithms based on randomly-sampled training data, such as cross-validation. The (random) selection of the training data corresponds to SHM. Notice that many training algorithms are essentially based on the histogram of the training data (instead of the ordering of data points). Therefore, overall the training procedure can be viewed as SHM plus a post-processing function (the learning algorithm). Consequently, the smoothed DP of SHM can be used to upper bound the smoothed DP of such procedures.

The third group consists of SGD of DNNs with gradient quantization [92], [73], where the gradients are rounded to 8-bit in order to accelerate the training and inference of DNNs. The smoothed DP of SHM can be used to bound the privacy leakage in each SGD step of the DNN, where a batch (subset of the training set) is firstly sampled and the gradient is the average of the gradients of the sampled data.

DP vs. Smoothed DP for SHM. We are ready to present the main theorem of this chapter, which indicates that SHM is very private under some mild assumptions. We say distribution π is *strictly positive*, if there exists a positive constant c such that $\pi(X) \geq c$ for any X in the support of π. A set of distributions Π is *strictly positive* is there exists a positive constant c such that every $\pi \in \Pi$ if strictly positive (by c). The strictly positive assumption is often considered mild in *elections* [80] and *discrete machine learning* [93], even though it may not hold for every step of SGD.

Theorem 8 (**DP vs. Smoothed DP for Sampling Histogram Mechanism $\mathcal{M}_H$**). *For any $\mathcal{M}_H$, given any strictly positive set of distribution Π, any finite set $\mathcal{X}$, and any $n, T \in \mathbb{Z}_+$, we have:*
(i) **(Smoothed DP)** *$\mathcal{M}_H$ is $\Big(\epsilon, \exp\big(-\Theta(\frac{(n-T)^2}{n})\big), \Pi\Big)$-smoothed DP for any $\epsilon \geq \ln\big(\frac{2n}{n-T}\big)$.*[6]
(ii) **(Tightness of smoothed DP bound)** *For any $\epsilon > 0$, there does not exist $\delta = \exp(-\omega(n))$ such that $\mathcal{M}_H$ is (ϵ, δ, Π)-smoothed DP.*
(iii) **(DP)** *For any $\epsilon > 0$, there does not exist $0 \leq \delta < \frac{T}{n}$ such that $\mathcal{M}_H$ is (ϵ, δ)-DP.*

This theorem says the privacy leakage is exponentially small under real-world application scenarios. In comparison, DP cares too much about the extremely rare cases and

[6]This exponential upper bound of δ is also affected by ϵ and m, where m is the cardinality of $\mathcal{X}$. In general, a smaller ϵ or a larger m results in a larger δ. See Appendix B.7.1 for detailed discussions.

predicts a constant privacy leakage. Also, note that our theorem allows T to be at the same order of n. For example, when setting $T = 90\% \times n$, SHM is $(3, \exp(-\Theta(n)), \Pi)$-smoothed DP, which is an acceptable privacy threshold in many real-world applications. Appendix B.8 proves similar bounds for the SHM with replacement.

4.5.2 Continuous Mechanisms Are Similar to What DP Predicts

In this section, we show that the sampling mechanisms with continuous support are still not privacy-preserving under smoothed DP. The "gap" between discrete and continuous SHM comes from the fact that SHM becomes less private when $|\mathcal{X}|$ increases. See footnote 4 for detailed explanations.

Algorithm 8: Continuous sampling average $\mathcal{M}_A$

1 **Inputs:** The number of samples T and a database $x = \{X_1, \cdots, X_n\}$ where $X_i \in [0, 1]$ for all $i \in [n]$

2 Randomly sample T data from x without replacement. The sampled data are denoted as $X_{j_1}, \cdots, X_{j_T}$.

3 **Output:** The average $\bar{x} = \frac{1}{T} \sum_{j \in [T]} X_{i_j}$

Our result indicates that the neural networks without quantized parameters are not private without external noise (i.e., the Gaussian or Laplacian noise). We use the sampling-average (Algorithm 8) algorithm as the standard algorithm for continuous mechanisms. Because sampling-average can be treated as SHM plus an average step, sampling-average is non-private also means SHM with continuous support is also non-private according to the post-processing property of smoothed DP.

Theorem 9 (**Smoothed DP for continuous sampling average**). *For any continuous sampling average algorithm $\mathcal{M}_A$, given any set of strictly positive*[7] *distribution Π over $[0, 1]$, any $T, n \in \mathbb{Z}_+$ and any $\epsilon \geq 0$, there does not exist $0 \leq \delta < \frac{T}{n}$ such that $\mathcal{M}_A$ is (ϵ, δ, Π)-smoothed DP.*

4.6 Experiments

Smoothed DP in elections. We use a similar setting as the motivating example, where 0.2% of the votes are randomly lost. We numerically calculate the δ parameter of

[7]Distribution π is strictly positive by c if $p_\pi(x) \geq c$ for any x in the support of π, where p_π is the PDF of π.

smoothed DP. Here, the set of distributions Π includes the distribution of all 57 congressional districts of the 2020 presidential election. Using the *distribution reduction* property of smoothed DP (Proposition 5), we can remove all distributions in Π except DC and NE-2[8], which are the vertices for the convex hull of Π. Figure 4.2 (left) shows that the smoothed δ parameter is exponentially small in n when $\epsilon = 7$, which matches our Theorem 8. We find that δ is also exponentially small when $\epsilon = 0.5, 1$ or 2, which indicates that the sampling-histogram mechanism is more private than DP's predictions under our settings. Also, see Appendix B.9.2 for the experiments with different settings on Π. All experiments of this section are implemented in MATLAB 2021a and tested on a Windows 10 Desktop with an Intel Core i7-8700 CPU and 32GB RAM.

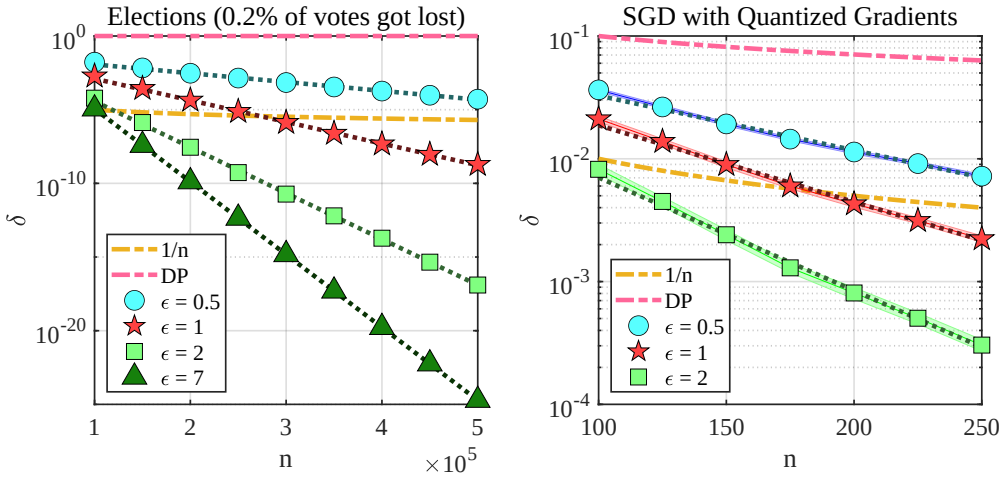

Figure 4.2: Left: DP and smoothed DP of 2020 US presidential election when 0.2% of votes got lost. Right: DP and smoothed DP of 1-step SGD with 8-bit gradient quantization when the set of distributions is Π. In both plots, the vertical axes are in log-scale and the pink dashed line presents the δ parameter of DP with whatever (finite) ϵ. The dot lines are the exponential fittings of smoothed δ parameters. The left plot is an accurate calculation of δ. The shaded area shows the 99% confidence interval of the right plot.

SGD with 8-bit gradient quantization. According to the pre-processing property of smoothed DP, the smoothed DP of (discrete) sampling average mechanism upper bounds the smoothed DP of SGD (for one step). In 8-bit neural networks for computer vision tasks, the gradient usually follows Gaussian distributions [73]. We thus let the set of distributions $\Pi = \{\mathcal{N}_{\text{8-bit}}(0, 0.12^2), \mathcal{N}_{\text{8-bit}}(0.2, 0.12^2)\}$, where $\mathcal{N}_{\text{8-bit}}(\mu, \sigma^2)$ denotes the 8-bit quantized

[8]DC refers to Washington, D.C., and NE-2 refers to Nebraska's 2nd congressional district.

Gaussian distribution (See Appendix B.9 for its formal definition). The standard variation, 0.12, is the same as the standard variation of gradients in a ResNet-18 network trained on CIFAR-10 dataset [73]. We use the standard setting of batch size $T = \sqrt{n}$. Figure 4.2 (right) shows that the smoothed δ parameter is exponentially small in n for the SGD with 8-bit gradient quantization. The probabilities are estimated through 10^6 independent trails. This result implies that the neural networks trained through quantized gradients can be private without adding external noises. Also, see Appendix B.9.2 for the experiments under another three different settings on Π.

4.7 Conclusions and Future Work

We propose a novel notion to measure the privacy leakage of mechanisms without external noises under realistic settings. One promising next step is to apply our smoothed DP notion to the entire training process of quantized DNNs. Is the quantized DNN private without external noise? If not, what level of external noises needs to be added, and how should we add noises in an optimal way? More generally, we believe that our work has the potential of making many algorithms private without requiring too much external noise.

CHAPTER 5
CERTIFIABLY ROBUST INTERPRETATION VIA RÉNYI DIFFERENTIAL PRIVACY

5.1 Introduction

Convolutional neural networks (CNNs) have demonstrated success in various computer vision applications, such as image classification [94], [95], object detection [96], semantic segmentation [97] and video recognition [98]. Spurred by the promising applications of CNNs, understanding why they make the correct prediction has become essential. The interpretation map explains which part of the input image plays a more important role in the prediction of CNNs. However, it has recently been shown that many interpretation maps, e.g., Simple Gradient [99], Integrated Gradient [100], DeepLIFT [101] and GradCam [102] are vulnerable to imperceptible input perturbations [6], [103]. In other words, slight perturbations to an input image could cause a significant discrepancy in its coupled interpretation map while keeping the predicted label unchanged (see an illustrative example in Figure 5.1).

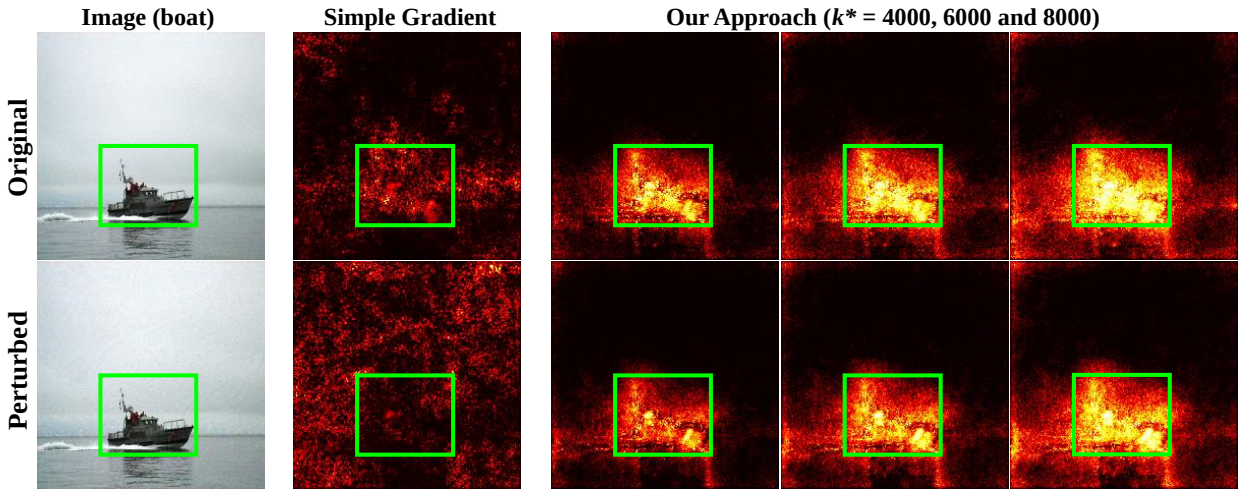

Figure 5.1: An example of the vulnerability of *simple gradient*. The green boxes annotate the position of the labeled object. Our approach offers stronger robustness against interpretation attacks. Remarkably, our interpretation also more accurately matches the position of the object.

The robustness of the interpretation maps is essential. Successful attacks can create

Portions of this chapter have previously appeared as: A. Liu, X. Chen, S. Liu, L. Xia, and C. Gan."Certifiably robust interpretation via Rényi differential privacy," *Artif. Intell.* vol. 313, Dec. 2022, Art. no. 103787.

confusion between the model interpreter and the classifier, which would further ruin the trustworthiness of systems that use the interpretations in down-stream actions, e.g., medical recommendation [104], source code captioning [105], and transfer learning [106]. In this chapter, we aim to provide a framework to generate certifiably robust interpretation maps against ℓ_d-norm attacks, where $d \in [1, \infty]$[9]. Our framework does not require the defender to know the type of attack, as long as it is an ℓ_d-norm attack. Our framework provides a stronger robustness guarantee if the exact attack type is known.

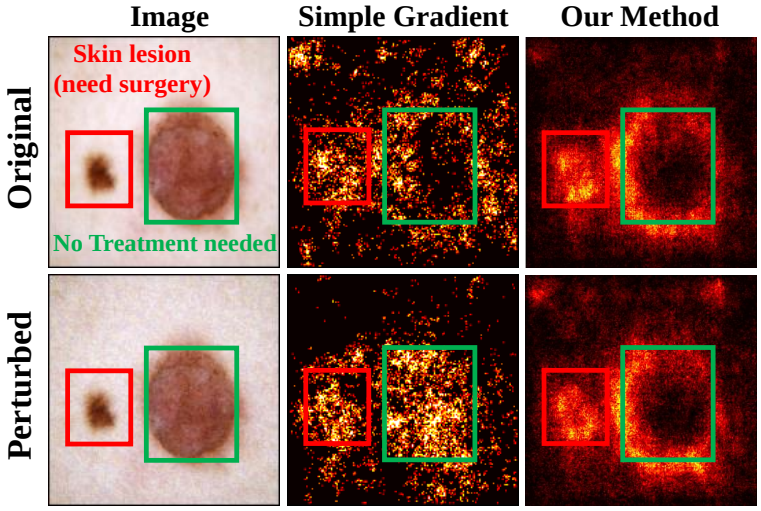

Figure 5.2: A motivating example of interpretation robustness.

A Motivating Example. To better motivate and strengthen our claim, we present an experiment on real-world medical images: *skin lesion diagnosis* (see Figure 5.2, [107]). In this task, the ML classifier predicts a type of skin lesion, and an interpretation map can help doctors localize the most interpretable region responsible for the current prediction (e.g., the region highlighted by the red box in Figure 5.2). However, an adversary (that crafts imperceptible input perturbations) could fool the interpretation map (for incorrect identification of region marked by the green box in Figure 5.2) under the same classification label. If doctors perform a medical diagnosis based on the fooled interpretation map, the doctor will waste his/her time treating the incorrect region. Figure 5.2 shows that our proposed robust interpretation can effectively defend such an adversary.

Differential privacy (DP) has recently been introduced as a tool to improve the robustness of machine learning algorithms. DP-based robust machine learning algorithms can provide theoretical robustness guarantees against adversarial attacks [108]. *Rényi differential*

[9]In all discussions of this chapter, we use $[1, \infty]$ to represent $[1, \infty) \cup \{\infty\}$.

privacy (RDP) is a generalization of the standard notion of differential privacy. It has been proved that analyzing the robustness of CNNs using RDP can provide a stronger theoretical guarantee than standard DP [109]. However, the following question remains open.

How can we protect the interpretation map from ℓ_d-norm attacks for arbitrary $d \in [1, \infty]$?

This is the question we will address in this chapter.

Our main theoretical contribution is the Rényi-Robust-Smooth framework (Algorithm 9) that provides a smooth tradeoff between interpretation robustness and computational efficiency. Here, the robustness of interpretation is measured by the resistance to the top-k attribution[10] change of an interpretation map when facing interpretation attacks. To prove the robustness of our framework, we first propose the new notion of Rényi robustness (Definition 10), which is directly connected with RDP and other robustness notions. Then, we show that the interpretation robustness can be guaranteed by Rényi robustness (Section 5.4). Our RDP-based analysis can provide a significantly tighter robustness bound in comparison with the recently proposed interpretation method, Sparsified SmoothGrad [110] (Figure 5.6 left). Finally, we extend the proposed method to classification tasks (Algorithm 10), and show that the extended algorithm has certifiable robustness against ℓ_∞-norm attacks for classifications tasks (Figure 5.9).

Experimentally, our Rényi-Robust-Smooth can provide $\sim$12% improvement on robustness in comparison with state-of-the-art (Figure 5.6 left). Surprisingly, our improvement of interpretation robustness does not come at the cost of accuracy. We found that the accuracy of Rényi-Robust-Smooth out-performs the recently-proposed approach of Sparsified SmoothGrad (Figure 5.6 right). Our method works surprisingly well when computational resources are highly constrained. The top-k attributions of our approach are twice more robust than the existing approaches when the computational resources are highly constrained (Figure 5.8).

Related Works. Many algorithms have been proposed recently to interpret the predictions of CNNs (e.g., Simple Gradient [99], Integrated Gradient [100], DeepLIFT [101], CAM [111] and GradCam [102]). The idea of adding random noise is a popular tool to improve the performance of interpretation maps. Following this idea, SmoothGrad [112], Smooth Grad-

[10]Top-k attribution refers to the top-k important pixels (top-k largest components) in the interpretation maps.

CAM++ [113], among others, have been proposed to improve the performance of Simple Gradient, Grad-CAM, and other interpretation methods. However, none of the above-mentioned works can provide certified robustness to interpretation maps. We are only aware of one paper studying this problem [110], where the authors proposed an interpretation algorithm to provide certifiably robustness against only ℓ_2-norm attack.

Adversarial training is one popular tool to improve the robustness of CNNs against adversarial attacks [114]–[117]. The method of adversarial training has also been applied to the interpretation algorithm recently [118]. However, there is no theoretical guarantee to the robustness of adversarial training against any norm-based attacks. Theoretical guarantees usually are also missing in other popular tools [119], [120] to improve the robustness of CNNs against adversarial attacks.

To the best of our knowledge, we are the first to apply RDP to provide certified robustness of interpretation maps. The connection between DP and robustness was first revealed by [121]. [108] were the first to introduce a DP-based algorithm to improve the prediction robustness of CNNs using the DP-robustness conclusion. RDP [67] is a popular generalization of the standard notion of DP. RDP often can provide tighter privacy bound than standard DP in many applications [109]. [122] uses a voting-based RDP approach to improve the privacy level of neural networks. The tool of RDP is also used to improve the robustness of classifiers[123]. However, [123]'s (and its improvement [124]'s) approach can only defend ℓ_1-norm or ℓ_2-norm attacks for classifications tasks, and usually cannot effectively defense ℓ_d-norm attacks or ℓ_∞-norm attacks (even for classification tasks) [125].

5.2 Preliminaries

Interpretation Attacks. In this chapter, we focus on the interpretations of CNNs for image classification, where the input is an image $\boldsymbol{x}$, and the output is a label in $\mathcal{C}$. An interpretation of this CNN explains why CNN makes this decision by showing the importance of the features in the classification. An *interpretation algorithm* $\boldsymbol{g} : \mathbb{R}^n \times \mathcal{C} \to \mathbb{R}^n$ maps the (image, label) pairs to an n-dimensional interpretation map[11]. The output of $\boldsymbol{g}$ consists of pixel-level attribution scores, which reflect the impact of each pixel on making the prediction. Because interpretation attacks will not change in the prediction of CNNs, we sometimes omit the label part of the input when the context is clear. In an interpretation attack, the adversary

[11]In this chapter, we treat both the input $\boldsymbol{x}$ and interpretation map $\boldsymbol{m}$ as vectors of length n. The dimension of the output can be different from that of the input. We use $\mathbb{R}^n$ to simplify the notation.

replaces $\boldsymbol{x}$ with its perturbed version $\tilde{\boldsymbol{x}}$ while keeping the predicted label unchanged. We assume that the perturbation is constrained by ℓ_d-norm $||\boldsymbol{x}-\tilde{\boldsymbol{x}}||_d \triangleq \left(\sum_{i=1}^{n} |\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i|^d\right)^{1/d} \leq L$. When $d = \infty$, the constraint becomes $\max_i |\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i| \leq L$.

Measure of Interpretation Robustness. Interpretation maps are usually applied to identify the important features of CNNs in many tasks like object detection [126]. Those tasks usually care more about the top-k pixels of the interpretation maps. Accordingly, we adopt the same robustness measure as [6], [127]–[129]. Here, the robustness is measured by the overlapping ratio between the top-k components of $\boldsymbol{g}(\boldsymbol{x}, C)$ and the top-k components of $\boldsymbol{g}(\tilde{\boldsymbol{x}}, C)$. Here, we use $V_k(\boldsymbol{g}(\boldsymbol{x}, C), \boldsymbol{g}(\tilde{\boldsymbol{x}}, C))$ to denote this ratio. For example, we have $V_2((1, 2, 3), (2, 1, 3)) = 0.5$, because the 2^{nd} and 3^{rd} components are the top-2 components of $(1, 2, 3)$ while the 1^{st} and 3^{rd} components are the top-2 components of $(2, 1, 3)$.[12] Next, we give a formal definition of the top-k overlapping ratio. Before proceeding, we first define the set of top-k component $T_k(\cdot)$ of a vector, which is, $T_k(\boldsymbol{x}) \triangleq \{x : x \in \boldsymbol{x} \wedge \#\{x' : x' \in \boldsymbol{x} \wedge x' \geq x\} \leq k\}$. Then, the top-$k$ overlap ratio V_k between any pair of vectors $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ is defined as

$$V_k(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \triangleq \frac{1}{k}\left(\#\big[T_k(\boldsymbol{x}) \cap T_k(\tilde{\boldsymbol{x}})\big]\right). \tag{5.1}$$

Based on V_k, we formally define interpretation robustness as follows.

Definition 8. (β-Top-k Robustness or Interpretation Robustness) *For a given input $\boldsymbol{x}$ with label C, we say an interpretation method $\boldsymbol{g}(\cdot, C)$ is β-Top-k robust to ℓ_d-norm attack of size L if for any $\tilde{\boldsymbol{x}}$* s.t. $||\boldsymbol{x} - \tilde{\boldsymbol{x}}||_d \leq L$,

$$V_k\big(\boldsymbol{g}(\boldsymbol{x}, C), \boldsymbol{g}(\tilde{\boldsymbol{x}}, C)\big) \geq \beta. \tag{5.2}$$

Rényi Differential Privacy. RDP is a novel generalization of standard differential privacy. RDP uses Rényi divergence as the measure of the difference between distributions. Formally, for two distributions P and Q with the same support $\mathcal{S}$, the Rényi divergence of order $\alpha > 1$ is defined as:

$$D_\alpha\big(P||Q\big) \triangleq \frac{1}{\alpha - 1} \ln \mathbb{E}_{x\sim Q}\left(\frac{P(x)}{Q(x)}\right)^\alpha \quad \text{and} \quad D_\infty = \sup_{x\in\mathcal{S}} \ln \frac{P(x)}{Q(x)}. \tag{5.3}$$

[12]Then, the 3^{rd} component is the only overlapping component in top-2. We also note that the relative order between top-k components is not taken into account in this chapter.

In this chapter, we adopt a generalized RDP by considering "adjacent" inputs whose ℓ_d distance is no larger than L.[13]

Definition 9 (Rényi Differential Privacy)**.** *A randomized function $\grave{\boldsymbol{g}}$ is (α, ϵ, L)-Rényi differentially private to ℓ_d distance, if for any pair of inputs $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$* s.t. $||\boldsymbol{x} - \tilde{\boldsymbol{x}}||_d \leq L$,

$$D_\alpha\big(\grave{\boldsymbol{g}}(\boldsymbol{x}) \,||\, \grave{\boldsymbol{g}}(\tilde{\boldsymbol{x}})\big) \leq \epsilon. \tag{5.4}$$

Same as in standard DP, smaller values of ϵ correspond to a more private $\grave{\boldsymbol{g}}(\cdot)$. RDP generalizes the standard DP, which is RDP with $\alpha = \infty$. Recall that in this chapter $\boldsymbol{x}$ represents the input image and $\grave{\boldsymbol{g}}$ represents a (randomized) interpretation algorithm. To clarify our notations, we add ` sign only on the top of all randomized functions.

Intuitions of the RDP-Robustness Connection. Assume the randomized function $\grave{\boldsymbol{g}}(\cdot)$ is Rényi differentially private. According to the definition of RDP, for any input $\boldsymbol{x}$, $\grave{\boldsymbol{g}}(\boldsymbol{x})$ (which is a distribution) is insensitive to small perturbations on $\boldsymbol{x}$. Consider a deterministic algorithm $\boldsymbol{h}(\boldsymbol{x}) \triangleq \mathbb{E}_{\grave{\boldsymbol{g}}}\left[\grave{\boldsymbol{g}}(\boldsymbol{x})\right]$ that outputs the expectation of $\grave{\boldsymbol{g}}(\boldsymbol{x})$. Intuitive, $\boldsymbol{h}(\boldsymbol{x})$ is also insensitive to small perturbations on $\boldsymbol{x}$. In other words, the RDP property of $\grave{\boldsymbol{g}}(\cdot)$ leads to the robustness of $\boldsymbol{h}(\cdot)$.

Rényi Robustness. Inspired by the merits of RDP, we introduce a new robustness notion: Rényi robustness, whose merits are two-folds:

(i) The (α, ϵ, L)-RDP property of $\grave{\boldsymbol{g}}(\cdot)$ directly leads to the (α, ϵ, L)-Rényi robustness on $\mathbb{E}\left[\grave{\boldsymbol{g}}(\cdot)\right]$. Thus, similar to RDP, Rényi robustness also has many desirable properties.

(ii) Rényi robustness is closely related to other robustness notions. If setting $\alpha = \infty$, the robustness parameter ϵ measures the average relative change of output when the input is perturbed. In Theorem 11 of Section 5.4, we will also prove that Rényi robustness is closely related to β-top-k robustness.

Definition 10 (Rényi Robustness)**.** *For a given input $\boldsymbol{x} \in \mathcal{A}$, we say a deterministic algorithm $\boldsymbol{h}(\cdot) : \mathcal{A} \to [0,1]^n$ is (α, ϵ, L)-Rényi robust to ℓ_d-norm attack if for any $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$* s.t.

[13] Standard RDP assumes that the ℓ_0-norm of adjacent inputs is no more than 1.

$||\boldsymbol{x} - \tilde{\boldsymbol{x}}||_d \leq L,$

$$
\begin{aligned}
R_\alpha(\boldsymbol{h}(\boldsymbol{x}) \,||\, \boldsymbol{h}(\tilde{\boldsymbol{x}})) &\triangleq \frac{1}{\alpha - 1} \ln \left(\prod_{i=1}^{n} \boldsymbol{h}(\tilde{\boldsymbol{x}})_i \cdot \left(\frac{\boldsymbol{h}(\boldsymbol{x})_i}{\boldsymbol{h}(\tilde{\boldsymbol{x}})_i} \right)^{\alpha} \right) \\
&= \frac{1}{\alpha - 1} \ln \left(\prod_{i=1}^{n} \frac{\left(\boldsymbol{h}(\boldsymbol{x})_i\right)^{\alpha}}{\left(\boldsymbol{h}(\tilde{\boldsymbol{x}})_i\right)^{\alpha-1}} \right) \\
&\leq \epsilon,
\end{aligned}
\tag{5.5}
$$

where $\boldsymbol{h}(\cdot)_i$ *refers to the* i*-th components of* $\boldsymbol{h}(\cdot)$.

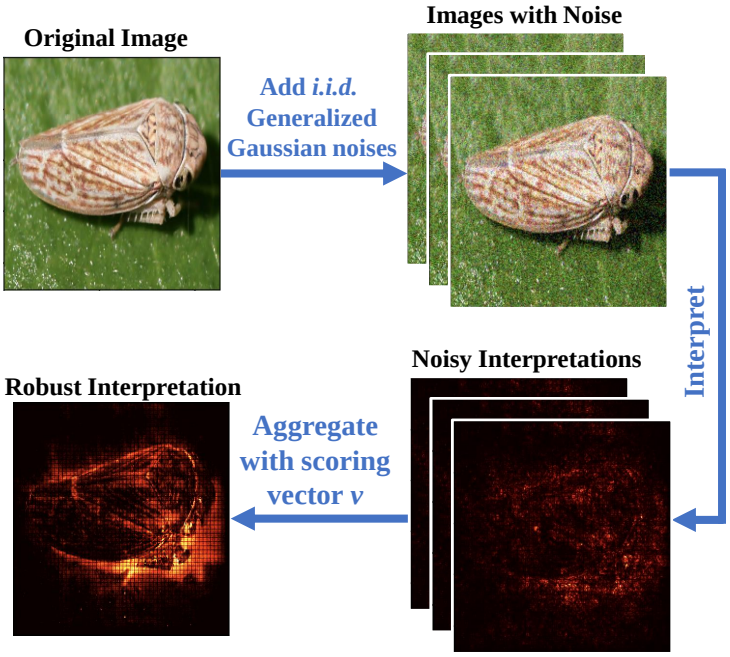

Figure 5.3: Workflow for our Rényi-Robust-Smooth with certifiable robustness against ℓ_d-norm attack.

5.3 The Rényi-Robust-Smooth Method

Our approach to generating robust interpretation is inspired by voting, which has been applied to improve the robustness of many algorithms with real-world applications [130]. We will generate T "voters" by introducing external noises to the input image, then use a voting rule to aggregate the outputs. Figure 5.3 shows the workflow of our approach, where generalized normal noises drawn from *generalized normal distribution* (GND) defined in Definition 11. .

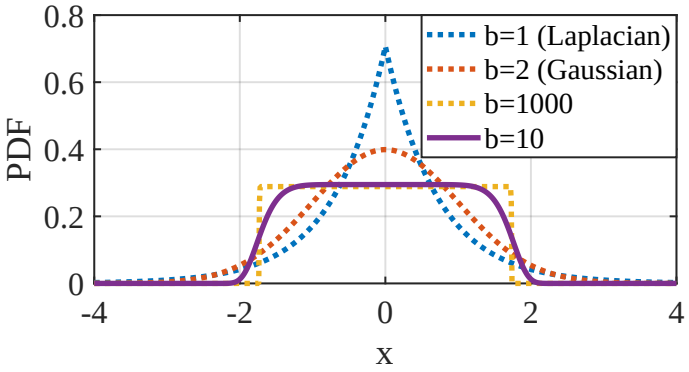

Figure 5.4: The probability density function (PDF) of generalized normal distributions.

Definition 11 (Generalized Normal Distribution)**.** *A random variable X follows generalized normal distribution $\mathcal{G}(\mu, \sigma, b)$ if its probability density function is:*

$$\frac{b \cdot \sqrt{\frac{\Gamma(3/b)}{\Gamma(1/b)}}}{2\sigma\Gamma(1/b)} \cdot \exp\left[-\left(\sqrt{\frac{\Gamma(3/b)}{\Gamma(1/b)}} \cdot \frac{|x-\mu|}{\sigma}\right)^b\right], \tag{5.6}$$

where μ, σ, and b correspond to the expectation, standard deviation, and shape factor of X. $\Gamma(\cdot)$ refers to gamma functions.

GND generalizes Gaussian distributions ($b = 2$) and Laplacian distribution ($b = 1$). Figure 5.4 plots the GNDs of $\sigma = 1$ under different shape factors. One can see that GNDs converge to the uniform distribution when $b \to \infty$.

Algorithm 9: Rényi-Robust-Smooth

1: **Inputs:** Base interpretation method $\boldsymbol{g}$, scoring vector $\boldsymbol{v}$, image $\boldsymbol{x}$ and the number of samples T
2: Generate *i.i.d.* noises $\boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_T \sim \mathcal{G}(\mathbf{0}, \sigma^2 I, b)$
3: Calculate the noisy interpretations $\grave{\boldsymbol{g}}_t^*(\boldsymbol{x}) \triangleq \boldsymbol{g}(\boldsymbol{x} + \boldsymbol{\delta}_t)$.
4: Re-scale $\grave{\boldsymbol{g}}_t^*(\boldsymbol{x})$ using scoring vector $\boldsymbol{v} = (v_1, \cdots, v_n)$. The noisy interpretation after re-scaling is denoted by $\grave{\boldsymbol{g}}_t(\boldsymbol{x})$, where $\grave{\boldsymbol{g}}_t(\boldsymbol{x})_i = v_j$ if and only if $\grave{\boldsymbol{g}}_t^*(\boldsymbol{x})_i$ is ranked j-th in $\grave{\boldsymbol{g}}_t^*(\boldsymbol{x})$.
5: **Output** Interpretation map $\boldsymbol{m} \triangleq \frac{1}{T}\sum_{t=1}^{T} \grave{\boldsymbol{g}}_t(\boldsymbol{x})$.

In Algorithm 9, we present our Rényi-Robust-Smooth in detail. The shape factor b is set according to the prior knowledge of the defender. We assume that the defender knows the attack is ℓ_d-norm attacks such that $d \leq d_{\text{prior}}$. For example, if the defender knows the attack is either ℓ_1-norm attack or ℓ_2-norm attack, then, $d_{\text{prior}} = 2$. Especially, $d_{\text{prior}} = \infty$

means the defender has no prior knowledge about the attack. Technically, we set the shape factor b according to the following setting:

$$b = \begin{cases} 1\,, & \text{when } d_{\text{prior}} = 1 \\ 2\lceil \frac{d_{\text{prior}}}{2} \rceil\,, & \text{when } 1 < d_{\text{prior}} \leq 2\lceil \frac{\ln n}{2} \rceil \\ \max\left\{2, 2\lceil \frac{\ln n}{2} \rceil\right\}\,, & \text{when } d_{\text{prior}} > 2\lceil \frac{\ln n}{2} \rceil \end{cases} . \tag{5.7}$$

In other words, b is the round-up of d_{prior} to the next even number, except that $d = 1$ or d is sufficiently large. We set an upper threshold on b because $\ell_{\ln n}$-norm is close to ℓ_∞-norm in practice. W.l.o.g. we set $v_1 \geq \cdots \geq v_n$ for the scoring vector $\boldsymbol{v}$. Our assumption guarantees that the pixels ranked higher in $\grave{\boldsymbol{g}}(\boldsymbol{x})$ will contribute more to our Rényi-Robust-Smooth map.

In comparison with Sparsified SmoothGrad [110], which only has theoretically guaranteed robustness against ℓ_2-norm attacks, our method has two main differences. First, we use generalized normal noise while SSG uses the standard Gaussian noise. Second, our scoring vector can be arbitrary while Sparsified SmoothGrad only allows binary vectors.

5.4 Rényi-Robust-Smooth with Certifiable Robustness

In this section, we first analyze the robustness of the expected output of Algorithm 9 in Section 5.4.1. Then, we show the tradeoff theorem between computational efficiency and robustness in Section 5.4.2. The big picture of our theoretical contributions is presented in Figure 5.5.

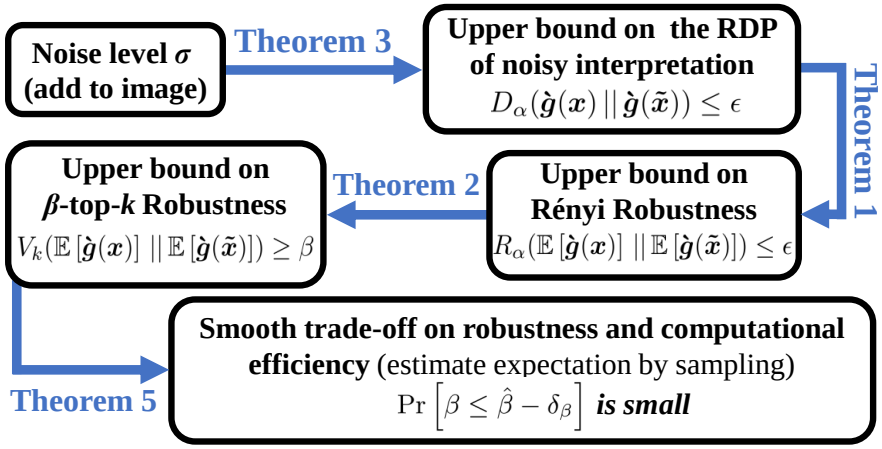

Figure 5.5: The workflow of our proofs to the robustness of our Rényi-Robust-Smooth.

5.4.1 The Expected Output of Algorithm 9

In this section, we discuss the expected output of Rényi-Robust-Smooth, which is denoted as $\mathbb{E}_{\grave{\boldsymbol{g}}}\left[\grave{\boldsymbol{g}}(\boldsymbol{x})\right]$[14]. As we will discuss in Theorem 12, the noise added to image can guarantee the RDP property of $\grave{\boldsymbol{g}}(\boldsymbol{x})$. According to the intuitions of RDP-robustness connection shown in Section 5.2, we may expect that $\mathbb{E}_{\grave{\boldsymbol{g}}}\left[\grave{\boldsymbol{g}}(\boldsymbol{x})\right]$ is robust to input perturbations. In all discussions of this subsection, we let $\boldsymbol{m} = \mathbb{E}_{\grave{\boldsymbol{g}}}[\grave{\boldsymbol{g}}(\boldsymbol{x})]$ and $\tilde{\boldsymbol{m}} = \mathbb{E}_{\grave{\boldsymbol{g}}}\left[\grave{\boldsymbol{g}}(\tilde{\boldsymbol{x}})\right]$. Next, we present Theorem 10, which connects RDP with Rényi robustness.

Theorem 10 (RDP $\rightarrow$ Rényi robustness)**.** *If a randomized function $\grave{\boldsymbol{g}}(\cdot)$ is (α, ϵ, L)-RDP to ℓ_d distance, then $\mathbb{E}_{\grave{\boldsymbol{g}}}\left[\grave{\boldsymbol{g}}(\cdot)\right]$ is (α, ϵ, L)-Rényi robust to ℓ_d distance.*

Proof of Theorem 10. Let $p_{(i,j)}$ (or $\tilde{p}_{(i,j)}$) denote the probability of the i-th pixel to be the j-th largest in the noisy interpretation (before rescaling) $\boldsymbol{g}(\boldsymbol{x}+\boldsymbol{\delta})$ (or $\boldsymbol{g}(\tilde{\boldsymbol{x}}+\boldsymbol{\delta})$). Then, the expectation of the i-th pixel in the robust interpretation map $\boldsymbol{m}_i = \mathbb{E}_{\grave{\boldsymbol{g}}}[\grave{\boldsymbol{g}}(\boldsymbol{x})_i] = \sum_{j=1}^{n} v_j\, p_{(i,j)}$. Similarly, for the perturbed input $\tilde{\boldsymbol{x}}$, we have $\tilde{\boldsymbol{m}}_i = \mathbb{E}_{\grave{\boldsymbol{g}}}[\grave{\boldsymbol{g}}(\tilde{\boldsymbol{x}})_i] = \sum_{j=1}^{n} v_j\, \tilde{p}_{(i,j)}$. By the definition of Rényi robustness, we have,

$$R_\alpha(\mathbb{E}_{\grave{\boldsymbol{g}}}\left[\grave{\boldsymbol{g}}(\boldsymbol{x})\right] \,||\, \mathbb{E}_{\grave{\boldsymbol{g}}}\left[\grave{\boldsymbol{g}}(\tilde{\boldsymbol{x}})\right]) = R_\alpha\left(\boldsymbol{m} \,||\, \tilde{\boldsymbol{m}}\right) = \frac{1}{\alpha-1}\ln\left(\prod_{i=1}^{n}\frac{\left(\sum_{j=1}^{n} v_j\, p_{(i,j)}\right)^{\alpha}}{\left(\sum_{j=1}^{k} v_j\, \tilde{p}_{(i,j)}\right)^{\alpha-1}}\right).$$

Then, we apply generalized Radon's inequality [131] and have,

$$R_\alpha(\mathbb{E}_{\grave{\boldsymbol{g}}}\left[\grave{\boldsymbol{g}}(\boldsymbol{x})\right] \,||\, \mathbb{E}_{\grave{\boldsymbol{g}}}\left[\grave{\boldsymbol{g}}(\tilde{\boldsymbol{x}})\right]) \leq \frac{1}{\alpha-1}\ln\left(\sum_{j=1}^{n} v_j \cdot \frac{p_{(i,j)}^{\alpha}}{\tilde{p}_{(i,j)}^{\alpha-1}}\right).$$

We note that the (α, ϵ)-RDP property of $\grave{\boldsymbol{g}}(\boldsymbol{x})$ provides $\frac{1}{\alpha-1}\ln\left(\sum_{j=1}^{n}\frac{p_{(i,j)}^{\alpha}}{\tilde{p}_{(i,j)}^{\alpha-1}}\right) \leq \epsilon$. Using the condition of $||\boldsymbol{v}||_1 = 1$ and $v_i \leq 1$, we have,

$$R_\alpha(\mathbb{E}_{\grave{\boldsymbol{g}}}\left[\grave{\boldsymbol{g}}(\boldsymbol{x})\right] \,||\, \mathbb{E}_{\grave{\boldsymbol{g}}}\left[\grave{\boldsymbol{g}}(\tilde{\boldsymbol{x}})\right]) \leq \epsilon.$$

By now, we already show the Rényi robustness property of $\mathbb{E}_{\grave{\boldsymbol{g}}}[\grave{\boldsymbol{g}}(\cdot)]$. □

In Theorem 11, we will show that the β-top-k robustness property can be guaranteed by the Rényi robustness property. To simplify our notation, we assume our (expected) Rényi-

[14] In this subsection, we neglect the footnote t to simplify the notation.

Robust-Smooth map $\boldsymbol{m} = (m_1, \cdots, m_n)$ is normalized ($||\boldsymbol{m}||_1 = 1$). We further let m_i^* be the i-th largest component in $\boldsymbol{m}$. Let $k_0 = \lfloor (1-\beta)k \rfloor + 1$ to denote the minimum number of changes to violet β-ratio overlapping of top-k. That is, to violet β-top-k robustness, there must be at most $(1-\beta)\cdot k$ (or equivalently, at most $k_0 = \lfloor (1-\beta)k \rfloor + 1$) components still in the top-k. Let $\mathcal{S} = \{k - k_0, \cdots, k + k_0 + 1\}$ denote the set of last k_0 components in top-k and the top k_0 components out of top-k.

Theorem 11 (Rényi robustness $\rightarrow$ β-top-k robustness)**.** *Let function $\boldsymbol{h}(\cdot)$ to be (α, ϵ, L)-Rényi robustness to ℓ_d distance. Then, $\boldsymbol{m} = \boldsymbol{h}(\boldsymbol{x})$ is β-Top-k robust to ℓ_d-norm attack of size L, if $\epsilon \leq \epsilon_{\text{robust}}(\alpha; \beta, k; \boldsymbol{m})$, who is defined as*

$$\epsilon_{\text{robust}}(\alpha; \beta, k; \boldsymbol{m}) \triangleq -\ln\left(2k_0\left(\frac{1}{2k_0}\sum_{i\in\mathcal{S}}(m_i^*)^{1-\alpha}\right)^{\frac{1}{1-\alpha}} + \sum_{i\notin\mathcal{S}} m_i^*\right).$$

Theorem 11 shows the level of Rényi robustness required by the β-top-k robustness condition. We provide some insights on ϵ_{robust}. There are two terms inside of the $\ln(\cdot)$ function. The first term corresponds to the minimum information loss to replace k_0 items from the top-k. The second term corresponds to the unchanged components in the process of replacement. The proof of Theorem 11 can be found in C.2.

Combining the results in Theorem 10 and Theorem 11, we know that β-top-k robustness can be guaranteed by the Rényi differential privacy property on the randomized interpretation algorithm $\grave{\boldsymbol{g}}(\boldsymbol{x}) = f_{\boldsymbol{v}}(\boldsymbol{g}(\boldsymbol{x}+\boldsymbol{\delta}, C))$, where $f_{\boldsymbol{v}}$ is the re-scaling function using scaling vector $\boldsymbol{v}$. Next, we show the RDP property of $\grave{\boldsymbol{g}}(\cdot)$. In the remainder of this chapter, we let $\Gamma(\cdot)$ denote the gamma function and let $\mathbb{1}(\cdot)$ denote the indicator function. To simplify notations, we let $\epsilon_\alpha\Big(\frac{L}{\sigma}\Big) = \frac{1}{\alpha-1}\ln\Big[\frac{\alpha}{\alpha-1}\exp\Big(\frac{(\alpha-1)L)}{\sqrt{2}\sigma}\Big) + \frac{\alpha-1}{2\alpha-1}\exp\Big(\frac{-\alpha L}{\sqrt{2}\sigma}\Big)\Big]$ and $\epsilon_b\Big(\frac{L}{\sigma}\Big) = \frac{1}{\Gamma(1/b)}\sum_{i=1}^{b/2}\binom{2i}{b}\Big(\frac{L}{\sigma^*}\Big)^{2i}\Gamma\Big(\frac{b+1-2i}{b}\Big)$, where $\sigma^* = \sqrt{\frac{\Gamma(1/b)}{\Gamma(3/b)}}\,\sigma$.

Theorem 12 (Noise level $\rightarrow$ RDP)**.** *For any re-scaling function $f_{\boldsymbol{v}}(\cdot)$, let $\grave{\boldsymbol{g}}(\boldsymbol{x}) = f_{\boldsymbol{v}}(\boldsymbol{g}(\boldsymbol{x}+\boldsymbol{\delta}))$ where $\boldsymbol{\delta} \sim \mathcal{G}(\boldsymbol{0}, \sigma^2 I, b)$. Then, $\grave{\boldsymbol{g}}$ has the following properties with ℓ_d distance:*
(i) $\Big(1, \epsilon_b\big(\frac{L}{\sigma}\cdot\exp\big[\mathbb{1}(d > 2\lceil\frac{\ln n}{2}\rceil)\big]\big), L\Big)$*-RDP for all* $d \geq 2$.
(ii) For all $\alpha \geq 1$*, we have* $\Big(\alpha, \frac{\alpha L^2}{2\sigma^2}, L\Big)$*-RDP when* $d \in (1, 2]$ *and* $\big(\alpha, \epsilon_\alpha\big(\frac{L}{\sigma}\big), L\big)$*-RDP when* $d = 1$.

Proof. According to the *post-processing* property of RDP [67], we know that $D_\alpha\big(\grave{\boldsymbol{g}}(\boldsymbol{x}) \,||\, \grave{\boldsymbol{g}}(\tilde{\boldsymbol{x}})\big) \leq D_\alpha(\boldsymbol{x}+\boldsymbol{\delta} \,||\, \tilde{\boldsymbol{x}}+\boldsymbol{\delta})$. When $d \in (1, 2]$, we always have $||\boldsymbol{x} - \tilde{\boldsymbol{x}}||_d \geq ||\boldsymbol{x} - \tilde{\boldsymbol{x}}||_2$. Thus, all

conclusion requiring $||\boldsymbol{x}-\tilde{\boldsymbol{x}}||_d \leq L$ will also hold when the requirement becomes $||\boldsymbol{x}-\tilde{\boldsymbol{x}}||_2 \leq L$. Then, the $d \in (1,2]$ case of (ii) in Theorem 12 follows by the standard conclusion of RDP on Gaussian mechanism and the $d = 1$ case follows by the standard conclusion of RDP on Laplacian mechanisms. In Lemma 7, we prove the RDP bound for generalized normal mechanisms of even shape factors. This bound can be directly applied to bound the KL-privacy of generalized normal mechanisms [132].

Lemma 7. *For any positive even shape factor b, letting $x \sim \mathcal{G}(0,\sigma,b)$ and $\tilde{x} \sim \mathcal{G}(L,\sigma,b)$, we have*

$$\lim_{\alpha\to 1^+} D_\alpha(x \,||\, \tilde{x}) = \epsilon_b\big(L/\sigma\big).$$

The proof of Lemma 7 can be found in C.2.3. At the high level, the proof of Lemma 7 uses the first-order approximation to calculate the limitation of $\alpha \to 1$. When $d \leq 2\lceil\frac{\ln n}{2}\rceil$, we always have $b \geq d$ and (i) of this case holds according to the same reason as (ii). When $d > 2\lceil\frac{\ln n}{2}\rceil$, we have $||\boldsymbol{x}-\tilde{\boldsymbol{x}}||_b \leq n^{1/b-1/d}\cdot||\boldsymbol{x}-\tilde{\boldsymbol{x}}||_d \leq e\cdot||\boldsymbol{x}-\tilde{\boldsymbol{x}}||_d$. Thus, (i) of Theorem 12 also holds when $d > 2\lceil\frac{\ln n}{2}\rceil$. □

Combining the conclusions of Theorem 10-12, we get the theoretical robustness of Rényi-Robust-Smooth in Table 5.1. In the table, $\epsilon^{-1}_{(\cdot)}(\cdot)$ denotes the inverse function of $\epsilon(\cdot)$.

Table 5.1: Theoretical β-top-k robustness for $\boldsymbol{m} = \mathbb{E}\left[\grave{\boldsymbol{y}}(\boldsymbol{x})\right]$ when the defender has different prior knowledge on the attack types. We note ϵ_{robust} is a function of β, and the definition of ϵ_{robust} can be found in Theorem 11.

Prior knowledge	Distribution of noise	Max attack size L without violating β-top-k robustness
$d_{\text{prior}} = 1$	Laplacian Distribution	$L = \sigma\cdot\sup_{\alpha>1}\epsilon_\alpha^{-1}(\epsilon_{\text{robust}}(\alpha))$
$d_{\text{prior}} \in (1,2]$	Gaussian Distribution	$L = \sigma\cdot\sup_{\alpha>1}\sqrt{2\epsilon_{\text{robust}}(\alpha)/\alpha}$
$d_{\text{prior}} \in (2,\infty]$	GND with shape factor b	$L = \sigma\cdot\exp\left(-\mathbb{1}(d > 2\lceil\frac{\ln n}{2}\rceil)\right)\cdot\epsilon_b^{-1}\left(\lim_{\alpha\to1^+}\epsilon_{\text{robust}}(\alpha)\right)$

5.4.2 The Smooth Trade-off Between Robustness and Computational Efficiency

In Section 5.4.1, we assumed that the value of $\boldsymbol{m} = \mathbb{E}[\grave{\boldsymbol{y}}(\boldsymbol{x})]$ in Algorithm 9 can be computed efficiently. However, there may not even exist a closed form expression of $\boldsymbol{m}$, and thus $\boldsymbol{m}$ may not be computed efficiently. In this section, we study the robustness-computational efficiency trade-off when $\boldsymbol{m}$ is approximated through sampling. That is, approximate $\boldsymbol{m}$ using $\sum_{t=1}^{T}\grave{\boldsymbol{y}}_t(\boldsymbol{x})$ (the same procedure as Algorithm 9). To simplify notation, we let $\hat{\beta}$ to denote

the calculated top-k robustness from Table 5.1. We use β to denote the real robustness parameter of $\boldsymbol{m}$. We note that our approach will become more computationally-efficient when T becomes smaller. In Theorem 13, we show that the Rényi robustness parameter ϵ_{robust} will have a larger probability of being larger when the number of samples T becomes larger. The conclusion on attack size L or robust parameter β follows by applying Theorem 13 to Table 5.1 or Theorem 11 respectively. The formal version of Theorem 13 can be found in C.2.4.

Theorem 13 (Smooth Trade-off Theorem, Informal)**.** *Letting $\hat{\epsilon}_{\text{robust}}$ denote the estimated Rényi robustness parameter from T samples, we have*

$$\Pr\left[\epsilon_{\text{robust}} \geq (1-\delta_\epsilon)\hat{\epsilon}_{\text{robust}}\right] \geq 1 - \mathit{negl}(T)\,,$$

where negl$(\cdot)$ refer to a negligible function[15]*.*

5.5 Generalizations of Our Framework

This section discusses a simple generalization of our framework to classification robustness, which is one of the core issues in the field of robust machine learning. We would also like to note that classification robustness is a simple extension of our approach, not the main topic of this chapter. Classifier $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^n$ maps images to a vector of scores.[16] Here, the classifier output a vector score for each class, which can be the output of the last-layer neural network, a regret, etc. The class with the largest score corresponds to the prediction of the classifier. Similar to the interpretation robustness, the classification robustness [108], [123], [124] is also defined based on the worst-case ℓ_d-norm perturbation. Formally,

Definition 12 (Classification Robustness [108], [123], [124])**.** *For a given input $\boldsymbol{x}$, we say a classifier $\boldsymbol{f}(\cdot)$ is robust to ℓ_d-norm attack of size L if for any $\tilde{\boldsymbol{x}}$* s.t. *$||\boldsymbol{x} - \tilde{\boldsymbol{x}}||_d \leq L$,*

$$\underset{i}{\operatorname{argmax}}\, \boldsymbol{f}(\tilde{\boldsymbol{x}})_i = \underset{i}{\operatorname{argmax}}\, \boldsymbol{f}(\boldsymbol{x})_i, \tag{5.8}$$

where $\boldsymbol{f}(\tilde{\boldsymbol{x}})_i$ denotes the score of the i-th label.

[15] The strict deinitionn of negligible functions can be found in `https://en.wikipedia.org/wiki/Negligible_function`

[16] We slightly abuse the notations and use n to denote the number of classes in 5.5

Next, we generalize Algorithm 9 to classification robustness. At the high level, We treat the vector score as a n-dimensional interpretation map, where we only care about the top-1 component of the score, which corresponds to the prediction. In other words, the classifications robustness can be viewed as the top-1 robustness property in interpretation robustness problems.

Algorithm 10: Rényi-Robust-Smooth for Classifications

1: **Inputs:** Classifier $\boldsymbol{f}$, the scoring vector $\boldsymbol{v}$, image $\boldsymbol{x}$ and the number of samples T
2: Generate *i.i.d.* noises $\boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_T \sim \mathcal{G}(\mathbf{0}, \sigma^2 I, b)$, where b is selected according to equation (5.7).
3: Calculate the score of each class $\grave{\boldsymbol{f}}_t^*(\boldsymbol{x}) \triangleq \boldsymbol{f}(\boldsymbol{x} + \boldsymbol{\delta}_t)$.
4: Re-scale $\grave{\boldsymbol{f}}_t^*(\boldsymbol{x})$ using scoring vector $\boldsymbol{v} = (v_1, \cdots, v_n)$. The score of each class after re-scaling is denoted by $\grave{\boldsymbol{f}}_t(\boldsymbol{x})$, where $\grave{\boldsymbol{f}}_t(\boldsymbol{x})_i = v_j$ if and only if $\grave{\boldsymbol{f}}_t^*(\boldsymbol{x})_i$ is ranked j-th in $\grave{\boldsymbol{f}}_t^*(\boldsymbol{x})$.
5: Calculate the score of each class $\boldsymbol{s} \triangleq \frac{1}{T}\sum_{t=1}^{T} \grave{\boldsymbol{f}}_t(\boldsymbol{x})$
6: **Output** The largest component of $\boldsymbol{s}$ as the robust classification.

Algorithm 10 presents a natural generalization of Algorithm 9 on classification robustness. We found Algorithm 10 shares a similar framework as Algorithm 1 in [123] except the following two main differences,

1. We use generalized normal distributions to defend ℓ_d-norm attacks for any $d \in [1, \infty]$, while [123] can only defend ℓ_1 and ℓ_2-norm attacks using Laplacian and Gaussian noise.

2. We allow the experimentalists to arbitrarily select scoring vector $\boldsymbol{v}$, while [123] requires $\boldsymbol{v} = (1, 0, \cdots, 0)$.

Similar to Algorithm 9 (robust interpretation), the generalized normal noise brings Algorithm 10 theoretically guaranteed robustness against general ℓ_d-norm attacks. In the next theorem, we show the robustness guarantee of Algorithm 10 against ℓ_d-norm attacks for $d \in [1, \infty]$.

Theorem 14 (Robustness of Algorithm 10)**.** *Using similar notations as Table 5.1, the robustness of Algorithm 10 is shown in Table 5.2, where*

$$\epsilon_{class} \triangleq -\ln\left(2\left(\frac{1}{2}(m_1^*)^{1-\alpha} + \frac{1}{2}(m_2^*)^{1-\alpha}\right)^{\frac{1}{1-\alpha}} + \sum_{i\geq 3} m_i^*\right),$$

and m_i^* represents the i-th largest component of $\frac{\boldsymbol{s}}{||\boldsymbol{s}||_2}$ (see Step 5 in Algorithm 10 for the definition of $\boldsymbol{s}$).

Table 5.2: Theoretical classification robustness when the defender has different prior knowledge about the attack types.

Prior knowledge	Distribution of noise	Max attack size L without changing top-1 classification
$d_{\text{prior}} = 1$	Laplacian Distribution	$L = \sigma \cdot \sup_{\alpha>1} \epsilon_\alpha^{-1}(\epsilon_{\text{class}}(\alpha))$
$d_{\text{prior}} \in (1, 2]$	Gaussian Distribution	$L = \sigma \cdot \sup_{\alpha>1} \sqrt{2\epsilon_{\text{class}}(\alpha)/\alpha}$
$d_{\text{prior}} \in (2, \infty]$	GND with shape factor b	$L = \sigma \cdot \exp\left(-\mathbb{1}(d > 2\lceil \frac{\ln n}{2} \rceil)\right) \cdot \epsilon_b^{-1}\left(\lim_{\alpha\to 1^+} \epsilon_{\text{class}}(\alpha)\right)$

5.6 Experimental Results

In this section, we experimentally evaluate the robustness and the interpretation accuracy of Rényi-Robust-Smooth. We show that our approach performs better on both ends than the existing approach of Sparsified SmoothGrad [110].

General setups of our implementation. We use PASCAL Visual Object Classes Challenge 2007 dataset (VOC2007, [133]) to evaluate both interpretation robustness and interpretation accuracy. This enables us to compare the annotated object positions in VOC2007 with the interpretation maps to benchmark the accuracy of interpretation methods. We adopt VGG-16 [134] as the CNN backbone for all the experiments in the main text. Additional experiments using ResNet-50 as the backbone can be found in C.3.3. Simple Gradient [99] is used as the base interpretation algorithm[17] (denoted as $\boldsymbol{g}(\cdot)$ in the input of Algorithm 9).

Defense and attack configurations. We focus our study on the most challenging case of ℓ_d-norm attack: ℓ_∞-norm attack. We examine the robustness and accuracy of our methods under the standard ℓ_∞-norm attack against the top-k component of Simple Gradient, which is firstly introduced by [6]. Formally, our attack method is presented in algorithm 11. To be more specific, we set the size of attack $L = 8/256 \approx 0.03$, learning rate $lr = 0.5$ and the number of iteration $N = 300$.

According to our noise setup discussed in Section 5.3, we set the shape factor of GND as $b = 10$ in consideration of VOC2007 dataset image size. To compare with state-of-the-art, we fix the standard deviation of noises to be 0.1. The scoring vector $\boldsymbol{v}$ used to aggregate "votes" is designed according to the power function $x^{-0.15}$. Mathematically, we take $\boldsymbol{v} = (v_1, \cdots, v_n)$ where $v_i = Z^{-1} \cdot i^{-0.15}$ and the normalization factor $Z = \sum_{i=1}^{n} i^{-0.15}$. The shape of the scoring vector is designed to optimize the experimental robustness (See Figure C.1 in C.3.1 for the

[17] We note that the base interpretation algorithm is not our baseline. The baseline throughout this chapter is Sparsified SmoothGrad.

Algorithm 11: ℓ_∞-norm Attack on Top-k Overlap [6]

Inputs: An integer k, learning rate lr, an input image $\boldsymbol{x} \in \mathbb{R}^n$; a interpretation method $\boldsymbol{g}(\cdot, \cdot)$, maximum ℓ_∞-norm perturbation L, and the number of iterations N
Output: Adversarial example $\tilde{\boldsymbol{x}}$.
Define $D(\boldsymbol{z}) = -\sum_{i \in B} \boldsymbol{g}(\boldsymbol{x})_i$, where B denotes the set of $\boldsymbol{g}(\boldsymbol{x})$'s top-$k$ components' subscripts.
Initialization: $\boldsymbol{x}^0 = \boldsymbol{x}$.
for $t = 1$ **to** N **do**
  $\boldsymbol{x}^t \leftarrow \boldsymbol{x}^{t-1} + lr \cdot \frac{\nabla D(\boldsymbol{x}^{t-1})}{\gamma}$
  if $||\boldsymbol{x}^t - \boldsymbol{x}||_\infty > \rho$ **then**
    $\boldsymbol{x}^t \leftarrow \boldsymbol{x} + \rho \cdot \frac{\boldsymbol{x}^t - \boldsymbol{x}}{||\boldsymbol{x}^t - \boldsymbol{x}||_\infty}$
  end if
end for
Output: $\tilde{\boldsymbol{x}} = \text{argmax}_{\boldsymbol{z} \in \{\boldsymbol{x}^0, \cdots, \boldsymbol{x}^N\}} D(\boldsymbol{z})$.

experimental robustness of other scoring vectors). We note that our theoretical guarantee of β-top-k robustness does not depend on a specific form of scoring vector. All experiments of this chapter are run on a Ubuntu 20.04LTS server with AMD EPYC 7H12, RTX 3090, and 256GB RAM. See Figure 5.1 in Section 5.1 for an illustration of our interpretation maps.

5.6.1 Robustness of Rényi-Robust-Smooth

We compare the β-top-k robustness of Rényi-Robust-Smooth with Sparsified SmoothGrad. Rényi-Robust-Smooth gets not only tighter robustness bound, but stronger robustness against ℓ_∞-norm attack in comparison with baseline. For both Sparsified SmoothGrad and our approach, we set the number of samples T to be 50.

Experimental Robustness. Figure 5.6 shows the β-top-k robustness of our approach in comparison with Sparsified SmoothGrad. Here, we adopt the transfer-attack setting, where an attack to the top-2500 of Simple Gradient is applied to both our approach and Sparsified SmoothGrad. We plot the top-k overlapping ratio β between the interpretation of the original image $\boldsymbol{x}$ and the adversarial example $\tilde{\boldsymbol{x}}$. Our approach has consistently better robustness ($\sim 12\%$) than Sparsified SmoothGrad under different settings on k. Also see Figure 5.8 for the experimental robustness comparison when the parameters T is different.

Theoretical Robustness Bound. The dash lines in Figure 5.6 presents the theoretical robustness bound under ℓ_∞-norm attack. One can see that our robustness bound is much tighter than the bound provided in [110]. This because the tool used by [110] is hard to be generalized to ℓ_∞-norm attack, and a direct transfer from ℓ_2-norm to ℓ_∞-norm will result in

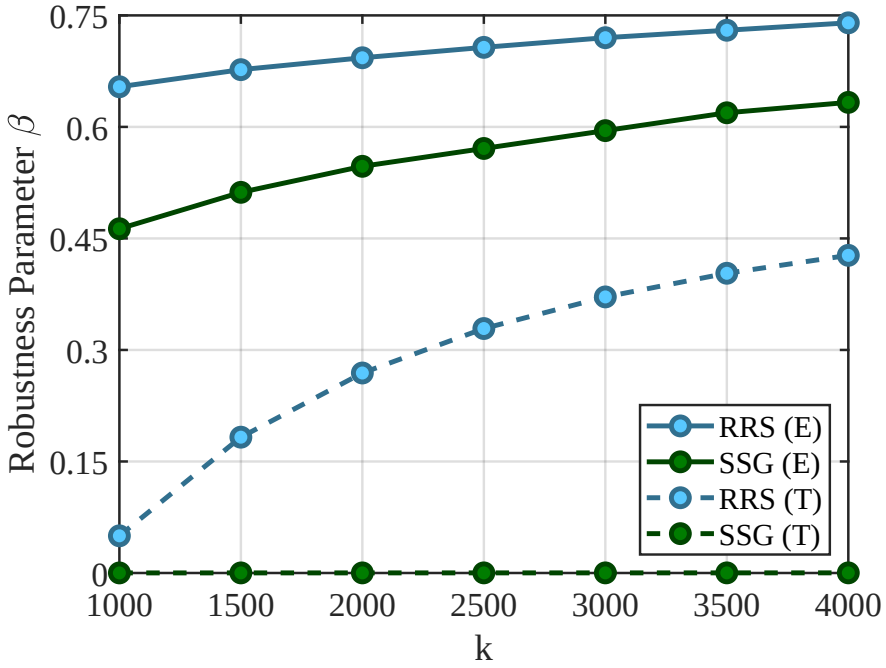

Figure 5.6: The β-top-k robustness of Rényi-Robust-Smooth (RRS) in comparison with Sparsified SmoothGrad (SSG). (E) and (T) corresponds to the experimental and theoretical robustness, respectively.

an extra loss of $\Theta(\sqrt{n})$.

We use the following experiments to further illustrate our theory under different attack sizes and different noise levels (the standard deviation of generalized normal distribution). The experimental settings are the same as Figure 5.6. Our results are summarized in Table 5.3 and Table 5.4, where β is the robustness parameter in Definition 8 (larger means more robust).

Table 5.3: Experimental and theoretical top-k robustness under different noise levels.

Noise level σ	0.07	0.1	0.15	0.2	0.3
Experimental β	0.535	0.665	0.720	0.745	0.748
Theoretical β	0.169	0.480	0.652	0.707	0.729

5.6.2 Accuracy of Interpretations

Accuracy is another main concern of interpretation maps. That is, to what extent the main attributes of interpretation overlap with the annotated objects. In this section, we introduce a generalization of *pointing game* [135], [136] to evaluate the performance of our interpretation method. In a pointing game, an interpretation method calculates the interpre-

Table 5.4: Experimental and theoretical top-k robustness under different size of attack L.

Attack size L	0.02	0.03	0.04
Experimental β	0.674	0.656	0.628
Theoretical β	0.672	0.508	0.271

tation map and compares it with the annotated object. If the top pixel in the interpretation map is within the object, a score "+1" will be to the object. Otherwise, a score "−1" will be granted. The pointing game score is the average score of all objects.

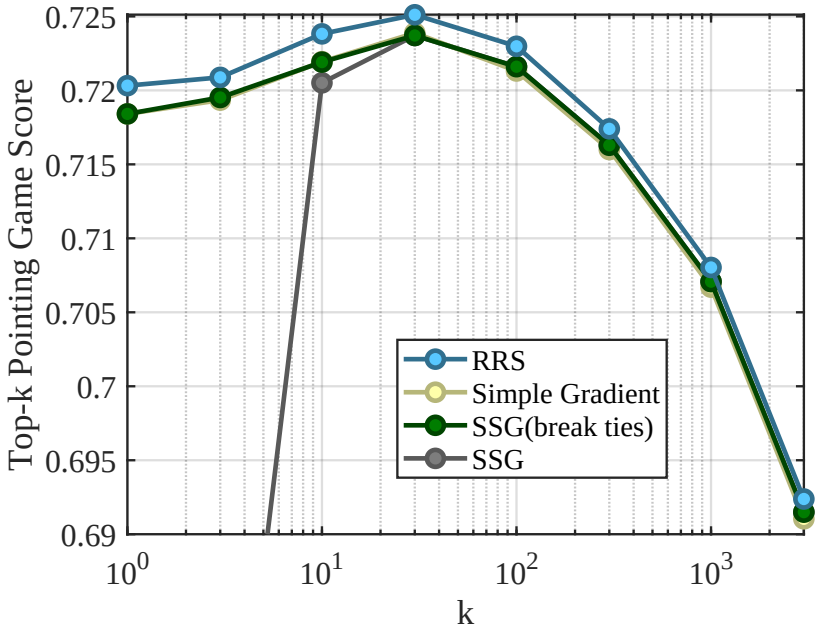

Figure 5.7: The accuracy of Rényi-Robust-Smooth (RRS) in comparison with Sparsified SmoothGrad (SSG) and Simple Gradient. The accuracy of SSG for $k < 10$ is too small and was ignored to improve the presentation. The accuracy of Simple Gradient is almost the same as SSG (break ties).

However, not only the top-1 pixel effect the quality of interpretation maps. Hence, we generalize the idea of pointing game to the top-k pixels by checking the ratio of top-k pixels of the interpretation map within the region of the object and applying the same procedure as the standard pointing game. Figure 5.7 shows the result of the top-k pointing game on various interpretation maps. The annotated object position in VOC2007 dataset (*e.g.*, the green boxes in Figure 5.1) is taken as the ground truth. The parameters of Sparsified SmoothGrad and our approach are both the same as the settings in Section 5.6.1. In figure 5.7, we compare

Rényi-Robust-Smooth with two versions of Sparsified SmoothedGad. The first version (SSG) follows the same setting as [110], which randomly breaks the ties when calculating its top-k component. For the second version (SSG (break ties)), we broke the ties according to descending order for the summation of all noisy interpretations. This order used for tie-breaking is the same as the order for standard SmoothedGrad. To our surprise, our approach's accuracy also turns out to be better than Sparsified SmoothGrad in both settings, even though robustness is usually considered an opposite property of accuracy.

5.6.3 Computational Efficiency-Robustness Tradeoff

Computational efficiency is another main concern of interpretation algorithms. However, most previous works on the interpretation of certified robustness did not pay much attention to computational efficiency.

If applying the settings of those works to larger images (*i.e.,* ImageNet or VOC figures) or complex CNNs (*i.e.,* VGG or GoogleLeNet), it may take hours to generate a single interpretation map. Thus, we experimentally verify our approach's performance when the number of generated noisy interpretations T is no larger than 30. At the high level, T can be used as a measure of computational resources because the time to compute on a noisy interpretation map is similar to our approach and Sparsified SmoothGrad. Experimentally, Figure 5.8 (right) shows that RRS's GPU consumption is almost the same as ($\sim$1% more than) SSG.

Figure 5.8 (left) shows the top-k robustness of our approach and Sparsified SmoothGrad when T is small and $k = 2500$. We observe that the robustness of both algorithms will decrease when T becomes smaller. However, our approach's robustness decreases much slower than Sparsified SmoothGrad when the computational resources become more limited. When $T = 5$, our Rényi-Robust-Smooth becomes *more than twice* more robust than Sparsified SmoothGrad. Figure C.4 shows similar behavior when the backbone of neural network is ResNet-50. In addition to SSG, we also show that RRS is more robust than the widely-used algorithm of Quadratic SmoothGrad [112]. The results are shown in C.3.2.

5.6.4 Robustness of Rényi-Robust-Smooth for Classifications

In Figure 5.9, we experimentally compare [108], [123], [124] with the classification version of Rényi-Robust-Smooth(Algorithm 10). We assume that the resolution of images is 224×224×3, which is the resolution of many popular datasets, including ImageNet. Here,

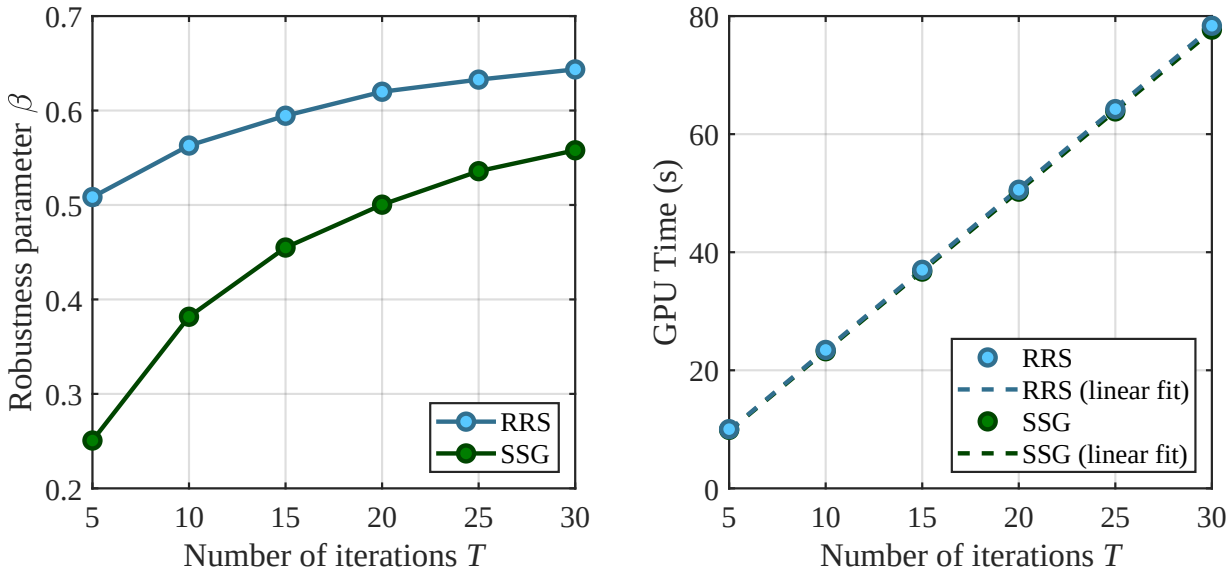

Figure 5.8: Left: the β-top-2500 robustness when the computational resources are highly constrained. Right: GPU consumption of RRS and SSG ($\sim 1\%$ less than RRS). The dashed lines show the linear fittings of RRS's and SSG's GPU consumption.

3 corresponds to the three color channels. We adopt the same setting as Figure 5 in [124], where the standard deviation of noise $\sigma = 1$, number of classes $n = 2$, and thus $m_2^* = 1 - m_1^*$. Figure 5.9 shows that the theoretical robustness of Rényi-Robust-Smooth (against ℓ_∞-norm attacks) significantly outperforms [108], [123], [124]. The theoretical robustness of baselines is calculated according to their reported ℓ_2-norm robustness[18]. Again, we would like to note that classification robustness is a simple extension of our approach, not the main topic of this chapter. We believe that extensive experiments for Algorithm 10 (*e.g.,* testing its accuracy on real-world datasets) can be a very interesting future direction.

5.7 Conclusions

In this chapter, we build a bridge to connect the property of RDP with the robustness of interpretation maps. Based on that, we propose a simple yet effective method to generate interpretation maps with certifiable robustness against broad kinds of attacks. Our approach can prevent the model interpreter and the model classifier from being confused by interpretation attacks. Our theoretical guarantee on robustness can provide tighter bounds than existing works against any ℓ_d-norm attacks for all $d \in [1, \infty]$. In Section 5.5, we extend

[18] According to norm inequalities, we calculate the ℓ_∞-norm robustness by dividing the ℓ_2-norm robustness by $\sqrt{\text{resolution}}$

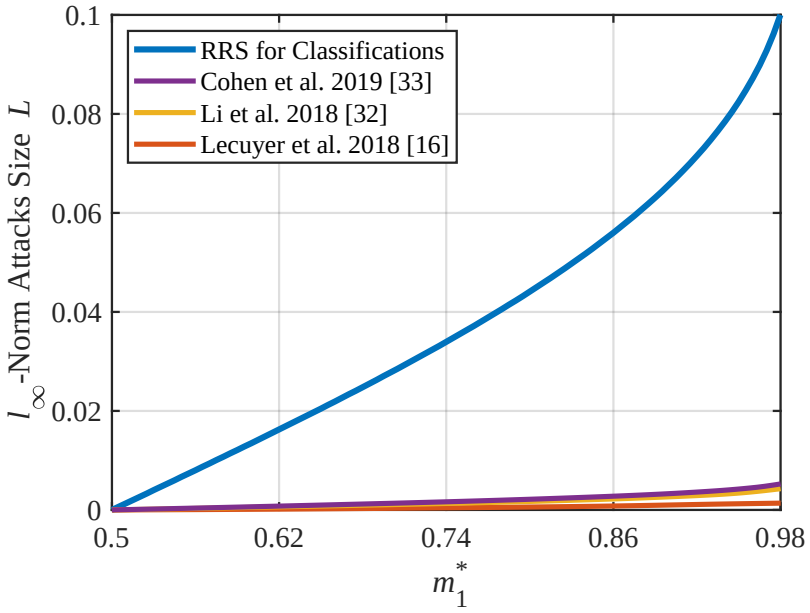

Figure 5.9: The theoretical robustness of RRS for classifications against ℓ_∞-norm attacks versus the theoretical robustness of literature baselines.

the proposed method to image classification tasks and show that its theoretical robustness (against ℓ_∞-norm attacks) outperforms the state of the art. Experimentally, we show that our approach can provide both better robustness and better accuracy than the recently proposed approach of Sparsified SmoothGrad.

REFERENCES

[1] N. N. Liu and Q. Yang, "Eigenrank: a ranking-oriented approach to collaborative filtering," in *Proc. Int. ACM SIGIR Conf. on Res. & Develop. in Inf. Retrieval*, 2008, pp. 83–90.

[2] J. Katz-Samuels and C. Scott, "Nonparametric preference completion," in *Proc. Int. Conf. on Artif. Intell. and Statist.*, 2018, pp. 632–641.

[3] J. Huang, A. Kapoor, and C. E. Guestrin, "Efficient probabilistic inference with partial ranking queries," 2012, *arXiv:1202.3734*.

[4] D. R. Hunter, "Mm algorithms for generalized bradley-terry models," *The Ann. of Statist.*, vol. 32, no. 1, pp. 384–406, Feb. 2004.

[5] K. E. Train, *Discrete Choice Methods with Simulation*. Cambridge, UK: Cambridge University Press, 2009.

[6] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3681–3688.

[7] T.-Y. Liu *et al.*, "Learning to rank for information retrieval," *Found. and Trends® in Inf. Retrieval*, vol. 3, no. 3, pp. 225–331, Mar. 2009.

[8] S.-w. Hwang and M.-W. Lee, "A uncertainty perspective on qualitative preference," in *Proc. Dagstuhl Seminar Proc.*, 2009, pp. 1–9.

[9] S. Wang, J. Sun, B. J. Gao, and J. Ma, "Adapting vector space model to ranking-based collaborative filtering," in *Proc. Int. Conf. on Inf. and Knowl. Manage.*, 2012, pp. 1487–1491.

[10] C. Fan and Z. Lin, "Collaborative ranking with ranking-based neighborhood," in *Proc. Asia-Pacific Web Conf.*, 2013, pp. 770–781.

[11] S. Wang, J. Sun, B. J. Gao, and J. Ma, "Vsrank: A novel framework for ranking-based collaborative filtering," *ACM Trans. on Intell. Syst. and Technol.*, vol. 5, no. 3, p. 51, Jun. 2014.

[12] D. Park, J. Neeman, J. Zhang, S. Sanghavi, and I. Dhillon, "Preference completion: Large-scale collaborative ranking from pairwise comparisons," in *Proc. Int. Conf. on Mach. Learn.*, 2015, pp. 1907–1916.

[13] D. McFadden and E. Choices, "Nobel prize lecture," *Am. Econ. Rev.*, vol. 91, p. 351, Oct. 2000.

[14] Y. Lu and S. N. Negahban, "Individualized rank aggregation using nuclear norm regularization," in *Proc. Commun., Control, and Comput.*, 2015, pp. 1473–1479.

[15] S. Oh, K. K. Thekumparampil, and J. Xu, "Collaboratively learning preferences from ordinal data," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2015, pp. 1909–1917.

[16] Z. Zhao, P. Piech, and L. Xia, "Learning mixtures of plackett-luce models," in *Proc. Int. Conf. on Mach. Learn.*, 2016, pp. 2906–2914.

[17] Z. Zhao, T. Villamil, and L. Xia, "Learning mixtures of random utility models," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4530–4538.

[18] A. Liu, Z. Zhao, C. Liao, P. Lu, and L. Xia, "Learning plackett-luce mixtures from partial preferences," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 01, 2019, pp. 4328–4335.

[19] H. A. Soufiani, W. Z. Chen, D. C. Parkes, and L. Xia, "Generalized method-of-moments for rank aggregation," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2013, pp. 2706–2714.

[20] H. A. Soufiani, D. C. Parkes, and L. Xia, "Preference elicitation for general random utility models," in *Proc. Conf. on Uncertainty in Artif. Intell.*, 2013, p. 596.

[21] H. Azari Soufiani, D. C. Parkes, and L. Xia, "Computing parametric ranking models via rank-breaking." in *Proc. Int. Conf. on Mach. Learn.*, 2014, pp. 360–368.

[22] W. Cheng, E. Hüllermeier, and K. J. Dembczynski, "Label ranking methods based on the plackett-luce model," in *Proc. Int. Conf. on Mach. Learn.*, 2010, pp. 215–222.

[23] D. Hughes, K. Hwang, and L. Xia, "Computing optimal bayesian decisions for rank aggregation via mcmc sampling." in *Proc. Conf. on Uncertainty in Artif. Intell.*, 2015, pp. 385–394.

[24] A. Khetan and S. Oh, "Data-driven rank breaking for efficient rank aggregation," in *Proc. Int. Conf. on Mach. Learn.*, 2016, pp. 89–98.

[25] L. Maystre and M. Grossglauser, "Fast and accurate inference of plackett-luce models," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2015, pp. 172–180.

[26] R. L. Plackett, "The analysis of permutations," *Applied Statist.*, vol. 24, pp. 193–202, Jun. 1975.

[27] R. D. Luce, "The choice axiom after twenty years," *J. of Math. Psychol.*, vol. 15, no. 3, pp. 215–233, Jun. 1977.

[28] I. Abraham, S. Chechik, D. Kempe, and A. Slivkins, "Low-distortion inference of latent similarities from a multiplex social network," *SIAM J. on Comput.*, vol. 44, no. 3, pp. 617–668, Mar. 2015.

[29] P. D. Hoff, A. E. Raftery, and M. S. Handcock, "Latent space approaches to social network analysis," *J. of the Amer. Statistical Assoc.*, vol. 97, no. 460, pp. 1090–1098, Dec. 2002.

[30] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *Proc. the 32nd Annu. ACM Symp. on Theory of Comput.*, 2000, pp. 163–170.

[31] P. Sarkar, D. Chakrabarti, and A. W. Moore, "Theoretical justification of popular link prediction heuristics." in *Proc. Int. Joint Conf. on Artif. Intell.*, vol. 22, 2011, p. 2722.

[32] M. Radovanovic, A. Nanopoulos, and M. Ivanovic, "Hubs in space: Popular nearest neighbors in high-dimensional data," *J. of Mach. Learn. Res.*, vol. 11, pp. 2487–2531, Sep. 2010.

[33] R. M. Bell and Y. Koren, "Lessons from the netflix prize challenge," *ACM SIGKDD Explorations Newslett.*, vol. 9, no. 2, pp. 75–79, Dec. 2007.

[34] J. Bennett, S. Lanning *et al.*, "The netflix prize," in *Proc. KDD Cup and Workshop*, 2007, p. 35.

[35] L. Terveen and W. Hill, "Beyond recommender systems: Helping people help each other," *HCI in the New Millennium*, vol. 1, no. 2001, pp. 487–509, Apr. 2001.

[36] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proc. Conf. on Uncertainty in Artif. Intell.*, 1998, pp. 43–52.

[37] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the web," in *Proc. the 10th Int. Conf. on World Wide Web*, 2001, pp. 613–622.

[38] L. Baltrunas, T. Makcinskas, and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering," in *Proc. the 4th ACM Conf. on Recommender Syst.*, 2010, pp. 119–126.

[39] T. Lu and C. Boutilier, "Effective sampling and learning for mallows models with pairwise-preference data," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3783–3829, Jan. 2014.

[40] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker, "Label ranking by learning pairwise preferences," *Artif. Intell.*, vol. 172, no. 16-17, pp. 1897–1916, Nov. 2008.

[41] H. A. Soufiani, D. C. Parkes, and L. Xia, "Random utility theory for social choice," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2012, pp. 126–134.

[42] L. L. Thurstone, "A law of comparative judgment," *Psychol. Rev.*, vol. 34, no. 4, p. 273, Jul. 1927.

[43] A. Khetan and S. Oh, "Computational and statistical tradeoffs in learning to rank," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2016, pp. 739–747.

[44] S. Negahban, S. Oh, and D. Shah, "Rank centrality: Ranking from pairwise com-parisons," *Operations Res.*, vol. 65, no. 1, pp. 266–287, Feb. 2016.

[45] Z. Zhao and L. Xia, "Composite marginal likelihood methods for random utility models," in *Proc. Int. Conf. on Mach. Learn.*, 2018, pp. 5922–5931.

[46] I. C. Gormley and T. B. Murphy, "Exploring voting blocs within the irish electorate: A mixture modeling approach," *J. of the Amer. Statistical Assoc.*, vol. 103, no. 483, pp. 1014–1027, Sep. 2008.

[47] S. Oh and D. Shah, "Learning mixed multinomial logit model from ordinal data," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2014, pp. 595–603.

[48] C. Mollica and L. Tardella, "Bayesian plackett–luce mixture models for partially ranked data," *Psychometrika*, vol. 82, no. 2, pp. 442–458, Jun. 2017.

[49] M. Tkachenko and H. W. Lauw, "Plackett-luce regression mixture model for heterogeneous rankings," in *Proc. the 25th ACM Int. on Conf. on Inf. and Knowl. Manage.*, 2016, pp. 237–246.

[50] Z. Zhao, H. Li, J. Wang, J. O. Kephart, N. Mattei, H. Su, and L. Xia, "A cost-effective framework for preference elicitation and aggregation," in *Proc. Conf. on Uncertainty in Artif. Intell.*, 2016, p. 179.

[51] J. I. Yellott Jr, "The relationship between luce's choice axiom, thurstone's theory of comparative judgment, and the double exponential distribution," *J. of Math. Psychol.*, vol. 15, no. 2, pp. 109–144, Apr. 1977.

[52] L. R. Ford Jr, "Solution of a ranking problem from binary comparisons," *The Amer. Math. Monthly*, vol. 64, no. 8P2, pp. 28–33, Oct. 1957.

[53] J.-P. Doignon, A. Pekeč, and M. Regenwetter, "The repeated insertion model for rankings: Missing link between two subset choice models," *Psychometrika*, vol. 69, no. 1, pp. 33–54, Mar. 2004.

[54] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The J. of Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, Jun. 1953.

[55] W. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, Apr. 1970.

[56] J. S. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling," *Statist. and Comput.*, vol. 6, pp. 113–119, Jun. 1996.

[57] N. Mattei and T. Walsh, "Preflib: A library for preferences http://www. preflib. org," in *Proc. Algorithmic Decision Theory: 3rd Int. Conf.*, 2013, pp. 259–270.

[58] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Found. and Trends in Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, Aug. 2014.

[59] A. Liu, Y. Lu, L. Xia, and V. Zikas, "How private are commonly-used voting rules?" in *Proc. Conf. on Uncertainty in Artif. Intell.*, 2020, pp. 629–638.

[60] J. Bogage and C. Ingraham, "USPS ballot problems unlikely to change outcomes in competitive states," *The Washington Post*, Accessed: Mar. 28, 2023. [Online]. Available: https://www.washingtonpost.com/business/2020/11/04/ballot-election-problems-usps/

[61] D. Leip, "Dave leip's atlas of the us presidential elections," Accessed: Mar. 28, 2023. [Online]. Available: https://uselectionatlas.org/

[62] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2019, pp. 15 479–15 488.

[63] G. E. Box, *Robustness in the Strategy of Scientific Model Building.* Amsterdam, Netherlands: Elsevier, 1979.

[64] D. A. Spielman, "The smoothed analysis of algorithms," in *Proc. Int. Symp. on Fundam. of Comput. Theory*, 2005, pp. 17–18.

[65] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. Annu. Int. Conf. on the Theory and Appl. of Cryptographic Techn.*, 2006, pp. 486–503.

[66] J. Dong, A. Roth, and W. J. Su, "Gaussian differential privacy," 2019, *arXiv:1905.02383*.

[67] I. Mironov, "Rényi differential privacy," in *Proc. Comput. Secur. Found. Symp.*, 2017, pp. 263–275.

[68] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Proc. Theory of Cryptogr. Conf.*, 2016, pp. 635–658.

[69] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," 2016, *arXiv:1603.01887*.

[70] A. Triastcyn and B. Faltings, "Bayesian differential privacy for machine learning," in *Proc. Int. Conf. on Mach. Learn.*, 2020, pp. 9583–9592.

[71] R. Hall, A. Rinaldo, and L. Wasserman, "Random differential privacy," 2011, *arXiv:1112.2680*.

[72] R. Bassily, A. Groce, J. Katz, and A. Smith, "Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy," in *Proc. Annu. Symp. on Found. of Comput. Sci.*, 2013, pp. 439–448.

[73] R. Banner, I. Hubara, E. Hoffer, and D. Soudry, "Scalable methods for 8-bit training of neural networks," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2018, pp. 5151–5159.

[74] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *The J. of Mach. Learn. Res.*, vol. 18, no. 1, pp. 6869–6898, Jan. 2017.

[75] M. Bun and T. Steinke, "Average-case averages: private algorithms for smooth sensitivity and mean estimation," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2019, pp. 181–191.

[76] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time," *J. of the ACM*, vol. 51, no. 3, pp. 385–463, May 2004.

[77] A. T. Kalai, A. Samorodnitsky, and S.-H. Teng, "Learning and smoothed analysis," in *2009 50th Annu. IEEE Symp. on Found. of Comput. Sci.*, 2009, pp. 395–404.

[78] B. Manthey and H. Röglin, "Worst-case and smoothed analysis of k-means clustering with bregman divergences," in *Proc. Int. Symp. on Algorithms and Comput.*, 2009, pp. 1024–1033.

[79] N. Haghtalab, T. Roughgarden, and A. Shetty, "Smoothed analysis of online and differentially private learning," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2020, pp. 9203–9215.

[80] L. Xia, "The smoothed possibility of social choice," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2020, pp. 11 044–11 055.

[81] D. Baumeister, T. Hogrebe, and J. Rothe, "Towards reality: Smoothed analysis in comput. social choice," in *Proc. Int. Conf. Auton. Agents & Multiagent Syst.*, 2020, pp. 1691–1695.

[82] L. Xia, "How likely are large elections tied?" in *Proc. ACM Conf. on Econ. & Comput.*, 2021, pp. 884–885.

[83] A. Liu and L. Xia, "The semi-random likelihood of doctrinal paradoxes," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 5124–5132.

[84] T. Brunsch, K. Cornelissen, B. Manthey, and H. Röglin, "Smoothed analysis of belief propagation for minimum-cost flow and matching," in *Proc. Int. Workshop on Algorithms and Comput.*, 2013, pp. 182–193.

[85] A. Bhaskara, M. Charikar, A. Moitra, and A. Vijayaraghavan, "Smoothed analysis of tensor decompositions," in *Proc. Annu. ACM Symp. on Theory of Comput.*, 2014, pp. 594–603.

[86] A. Blum and J. Dunagan, *Smoothed Analysis of the Perceptron Algorithm for Linear Programming*. Pittsburgh, PA, USA: Carnegie Mellon University, 2002.

[87] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proc. Annu. ACM Symp. on Theory of Comput.*, 2007, pp. 75–84.

[88] H. H. Nguyen, J. Kim, and Y. Kim, "Differential privacy in practice," *J. of Comput. Sci. & Engineering*, vol. 7, no. 3, pp. 177–186, Sep. 2013.

[89] A. G. Thakurta and A. Smith, "Differentially private feature selection via stability arguments, and the robustness of the lasso," in *Proc. Conf. on Learn. Theory*, 2013, pp. 819–850.

[90] L. Wasserman and S. Zhou, "A statistical framework for differential privacy," *J. of the Amer. Statistical Assoc.*, vol. 105, no. 489, pp. 375–389, Mar. 2010.

[91] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," in *Proc. Int. Conf. on Mach. Learn.*, 2015, pp. 1376–1385.

[92] F. Zhu, R. Gong, F. Yu, X. Liu, Y. Wang, Z. Li, X. Yang, and J. Yan, "Towards unified int8 training for convolutional neural network," in *Proc. the IEEE/CVF Conf. on Comput. Vision & Pattern Recognit.*, 2020, pp. 1969–1979.

[93] P. Laird and R. Saul, "Discrete sequence prediction and its applications," *Mach. Learn.*, vol. 15, no. 1, pp. 43–68, Apr. 1994.

[94] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[95] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. the IEEE/CVF Conf. on Comput. Vision & Pattern Recognit.*, 2016, pp. 770–778.

[96] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[97] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *Proc. the IEEE Int. Conf. on Comput. Vision*, 2015, pp. 1134–1142.

[98] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann, "Devnet: A deep event network for multimedia event detection and evidence recounting," in *Proc. the IEEE/CVF Conf. on Comput. Vision & Pattern Recognit.*, 2015, pp. 2568–2577.

[99] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2013, *arXiv:1312.6034*.

[100] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. on Mach. Learn.*, 2017, pp. 3319–3328.

[101] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. Int. Conf. on Mach. Learn.*, vol. 70, 2017, pp. 3145–3153.

[102] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proc. the IEEE Int. Conf. on Comput. Vision*, 2017, pp. 618–626.

[103] J. Heo, S. Joo, and T. Moon, "Fooling neural network interpretations via adversarial model manipulation," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2019, pp. 2921–2932.

[104] G. Quellec, K. Charrière, Y. Boudi, B. Cochener, and M. Lamard, "Deep image mining for diabetic retinopathy screening," *Med. Image Anal.*, vol. 39, pp. 178–193, Jul. 2017.

[105] G. Ramakrishnan, J. Henkel, Z. Wang, A. Albarghouthi, S. Jha, and T. Reps, "Semantic robustness of models of source code," 2020, *arXiv:2002.03043*.

[106] A. Shafahi, P. Saadatpanah, C. Zhu, A. Ghiasi, C. Studer, D. Jacobs, and T. Goldstein, "Adversarially robust transfer learning," in *Proc. Int. Conf. on Learn. Representations*, 2020, pp. 1–14.

[107] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Sci. Data*, vol. 5, Aug. 2018, Art. no. 180161.

[108] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," in *Proc. Symp. on Secur. and Privacy (SP)*, 2019, pp. 656–672.

[109] V. Feldman, I. Mironov, K. Talwar, and A. Thakurta, "Privacy amplification by iteration," in *Proc. Annu. Symp. on Found. of Comput. Sci.*, 2018, pp. 521–532.

[110] A. Levine, S. Singla, and S. Feizi, "Certifiably robust interpretation in deep learning," 2019, *arXiv:1905.12105*.

[111] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. the IEEE/CVF Conf. on Comput. Vision & Pattern Recognit.*, 2016, pp. 2921–2929.

[112] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," 2017, *arXiv:1706.03825*.

[113] D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam, "Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models," 2019, *arXiv:1908.01224*.

[114] A. Sinha, H. Namkoong, and J. Duchi, "Certifying some distributional robustness with principled adversarial training," 2017, *arXiv:1710.10571*.

[115] A. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1660–1669.

[116] L. Tong, Z. Chen, J. Ni, W. Cheng, D. Song, H. Chen, and Y. Vorobeychik, "Facesec: A fine-grained robustness evaluation framework for face recognition systems," in *Proc. the IEEE/CVF Conf. on Comput. Vision & Pattern Recognit.*, 2021, pp. 13 254–13 263.

[117] X. Li, D. Pan, and D. Zhu, "Defending against adversarial attacks on medical imaging ai system, classification or detection?" in *Proc. Int. Symp. on BioMed. Imag.*, 2021, pp. 1677–1681.

[118] M. Singh, N. Kumari, P. Mangla, A. Sinha, V. N. Balasubramanian, and B. Krishnamurthy, "On the benefits of attributional robustness," 2019, *arXiv:1911.13073*.

[119] P. Abolghasemi, A. Mazaheri, M. Shah, and L. Boloni, "Pay attention!-robustifying a deep visuomotor policy through task-focused visual attention," in *Proc. the IEEE/CVF Conf. on Comput. Vision & Pattern Recognit.*, 2019, pp. 4254–4262.

[120] X. Li and D. Zhu, "Robust detection of adversarial attacks on medical images," in *Proc. Int. Symp. on BioMed. Imag.*, 2020, pp. 1154–1158.

[121] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proc. Annu. ACM Symp. on Theory of Comput.*, 2009, pp. 371–380.

[122] Y. Zhu, X. Yu, M. Chandraker, and Y.-X. Wang, "Private-knn: Practical differential privacy for computer vision," in *Proc. the IEEE/CVF Conf. on Comput. Vision & Pattern Recognit.*, 2020, pp. 11 854–11 862.

[123] B. Li, C. Chen, W. Wang, and L. Carin, "Certified adversarial robustness with additive noise," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2019, pp. 9459–9469.

[124] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proc. Int. Conf. on Mach. Learn.*, 2019, pp. 1310–1320.

[125] S. Kotyan and D. V. Vargas, "Adversarial robustness assessment: Why in evaluation both l_0 and l_∞ attacks are necessary," *PloS One*, vol. 17, Apr. 2022, Art. no. e0265723.

[126] W. Chu and D. Cai, "Deep feature based contextual model for object detection," *NeuroComput.*, vol. 275, pp. 1035–1042, Jan. 2018.

[127] J. Chen, X. Wu, V. Rastogi, Y. Liang, and S. Jha, "Robust attribution regularization," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2019, pp. 14 302–14 312.

[128] M. Singh, N. Kumari, P. Mangla, A. Sinha, V. N. Balasubramanian, and B. Krishnamurthy, "Attributional robustness training using input-gradient spatial alignment," in *Proc. Eur. Conf. Comput. Vision*, 2020, pp. 515–533.

[129] A. Sarkar, A. Sarkar, and V. N. Balasubramanian, "Enhanced regularizers for attributional robustness," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2532–2540.

[130] J. A. Jenkins, "Examining the robustness of ideological voting: Evidence from the confederate house of representatives," *Amer. J. of Political Sci.*, vol. 44, pp. 811–822, Oct. 2000.

[131] D. M. Batinetu-Giurgiu and O. T. Pop, "A generalization of radon's inequality," *Creative Math. & Inf*, vol. 19, no. 2, pp. 116–121, Aug. 2010.

[132] Y.-X. Wang, J. Lei, and S. E. Fienberg, "On-average kl-privacy and its equivalence to generalization for max-entropy mechanisms," in *Proc. Int. Conf. on Privacy in Statist. Databases*, 2016, pp. 121–134.

[133] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge 2007 (VOC2007) results," Accessed: Mar. 30, 2023. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

[134] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[135] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," *Int. J. of Comput. Vision*, vol. 126, no. 10, pp. 1084–1102, Oct. 2018.

[136] R. Fong, M. Patrick, and A. Vedaldi, "Understanding deep networks via extremal perturbations and smooth masks," in *Proc. the IEEE Int. Conf. on Comput. Vision*, 2019, pp. 2950–2958.

[137] M. Marchesi, G. Orlandi, F. Piazza, and A. Uncini, "Fast neural networks without multipliers," *IEEE Trans. on Neural Netw.*, vol. 4, no. 1, pp. 53–62, Jan. 1993.

[138] C. Z. Tang and H. K. Kwan, "Multilayer feedforward neural networks with single powers-of-two weights," *IEEE Trans. on Signal Process.*, vol. 41, no. 8, pp. 2724–2727, Aug. 1993.

[139] W. Balzer, M. Takahashi, J. Ohta, and K. Kyuma, "Weight quantization in boltzmann machines," *Neural Netw.*, vol. 4, no. 3, pp. 405–409, Jan. 1991.

[140] E. Fiesler, A. Choudry, and H. J. Caulfield, "Weight discretization paradigm for optical neural networks," in *Proc. Opt. Interconnections & Netw.*, vol. 1281, 1990, pp. 164–173.

[141] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2016, pp. 4114–4122.

[142] Y. Guo, "A survey on methods and theories of quantized neural networks," 2018, *arXiv:1808.04752*.

[143] S. Anwar, K. Hwang, and W. Sung, "Fixed point optimization of deep convolutional neural networks for object recognition," in *Proc. Int. Conf. on Acoust., Speech & Signal Process.*, 2015, pp. 1131–1135.

[144] M. Kim and P. Smaragdis, "Bitwise neural networks," 2016, *arXiv:1601.06071*.

[145] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental network quantization: Towards lossless cnns with low-precision weights," 2017, *arXiv:1702.03044*.

[146] D. Lin, S. Talathi, and S. Annapureddy, "Fixed point quantization of deep convolutional networks," in *Proc. Int. Conf. on Mach. Learn.*, 2016, pp. 2849–2858.

[147] X. Lin, C. Zhao, and W. Pan, "Towards accurate binary convolutional neural network," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2017, pp. 344–352.

[148] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: training deep neural networks with binary weights during propagations," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2015, pp. 3123–3131.

[149] V. Vanhoucke, A. Senior, and M. Z. Mao, "Improving the speed of neural networks on cpus," in *Proc. Deep Learn. & Unsupervised Feature Learn. NIPS Workshop*, 2011, p. 4.

[150] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Proc. Eur. Conf. on Comput. Vision*, 2016, pp. 525–542.

[151] A. Mishra, E. Nurvitadhi, J. J. Cook, and D. Marr, "WRPN: Wide reduced-precision networks," in *Proc. Int. Conf. on Learn. Representations*, 2018, pp. 1–11.

[152] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns," in *Proc. Annu. Conf. of the Int. Speech Commun. Assoc.*, 2014, pp. 1058–1062.

[153] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic, "QSGD: communication-efficient sgd via gradient quantization and encoding," in *Proc. Int. Conf. on Neural Inf. Process. Syst.*, 2017, pp. 1707–1718.

[154] Y. Du, S. Yang, and K. Huang, "High-dimensional stochastic gradient quantization for communication-efficient edge learning," *IEEE Trans. on Signal Process.*, vol. 68, pp. 2128–2142, Mar. 2020.

APPENDIX A
APPENDIX FOR CHAPTER 2

A.1 Notation Tables and Additional Examples

This section contains a list of notations in the chapter, an example illustrating how the model works, and Figure A.1 for our algorithm analyses in Section 2.3 and Appendix A.2.3.

Notations.

- n: number of agents.
- m: number of alternatives.
- $\mathcal{X} = \{x_1, \cdots, x_n\}$: the set of agents.
- $\mathcal{Y} = \{y_1, \cdots, y_m\}$: the set of alternatives.
- $\theta(x_i, y_j) = \exp(-\|x_i - y_j\|_2)$: the parameter in the Plackett-Luce model. $\theta(x_i, y_j)$ is also y_j's expected utility to agent x_i.
- R_i: the full ranking of agent i.
- $\mathcal{R}(y_j)$: y_j's ranking in R.
- $y_{j_1} \succ_i y_{j_2}$: y_{j_1} is ahead of y_{j_2} in agent x_i's ranking R_i (or equivalently, agent x_i prefers y_{j_1} to y_{j_2}).
- $\mathcal{D}_X$: the distribution of each x_i.
- $\mathcal{D}_Y$: the distribution of each y_i.
- $f_X(\cdot)$: the probability density function of $\mathcal{D}_X$.
- $f_Y(\cdot)$: the probability density function of $\mathcal{D}_Y$.
- $c_X = \frac{\sup f_X(x)}{\inf f_X(x)}$: the parameter in the near-uniform distribution $\mathcal{D}_X$.
- $c_Y = \frac{\sup f_Y(y)}{\inf f_Y(y)}$: the parameter in the near-uniform distribution $\mathcal{D}_Y$.

Portions of this chapter have previously appeared as: A. Liu, Q. Wu, Z. Liu, and L. Xia."Near-neighbor methods in random preference completion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4336-4343.

- $\mathcal{O}_i$: The observed ranking from agent i over the subset $\mathcal{O}_i$.
- p: the probability an alternative y_j is sampled in $\mathcal{O}_i$.
- $R^{\mathcal{O}}$: ordered list over $\mathcal{O} \subseteq \mathcal{Y}$ that is consistent with R.
- L_i: the ground-truth ranking. Specifically $L_i = (y_{j_1}, y_{j_2} \ldots, y_{j_m})$, where $\theta(x_i, y_{j_1}) > \theta(x_i, y_{j_2}) > \cdots > \theta(x_i, y_{j_m})$.
- PC problem: the preference completion problem.
- NN problem: the near neighbor problem.
- k: the number of agents returned by a kNN algorithm.
- $\tau(n, m)$: the quality parameter of a kNN algorithm for agents. The distance between any agent returned by the algorithm and input x_i needs to be smaller than $\tau(n, m)$.
- KT: (un-normalized) Kendall-Tau distance.
- $\mathrm{NK}(R_i, R_j)$: the normalized KT distance between R_i and R_j.
- $\mathbf{I}(\cdot)$: an indicator function.
- $\mathcal{G}_i(x) = \mathbb{E}[\mathrm{NK}(R_i, R_x) \mid x_i, x]$.
- $\mathrm{sign}(x)$: returns 1 when x is positive and returns -1 otherwise.
- $\cosh(x) = \frac{e^x + e^{-x}}{2}$: hyperbolic cosine function.
- $$p_i(y_1, y_2) \triangleq \Pr[y_1 \succ_i y_2 \mid y_1, y_2, x_i] = \frac{\theta(x_i, y_1)}{\theta(x_i, y_1) + \theta(x_i, y_2)} \tag{A.1}$$
$$p_x(y_1, y_2) \triangleq \Pr[y_1 \succ_x y_2 \mid y_1, y_2, x] = \frac{\theta(x, y_1)}{\theta(x, y_1) + \theta(x, y_2)} \tag{A.2}$$
- $$\begin{cases} \Phi(y_1, y_2, x, x_i) & = \frac{e^{-\Delta^{(i)}_{1,2}} - 1}{e^{-\Delta^{(i)}_{1,2}} + 1} \cdot \frac{\mathrm{sign}(y_1 - x) - \mathrm{sign}(y_2 - x)}{4 \cosh^2\left(\Delta^{(x)}_{1,2}/2\right)} \\ \Delta^{(i)}_{1,2} & = |y_2 - x_i| - |y_1 - x_i| \\ \Delta^{(x)}_{1,2} & = |y_2 - x| - |y_1 - x| \end{cases} \tag{A.3}$$

- $g(x) \equiv \frac{e^{-x}-1}{e^{-x}-1}$: the first part of Φ where we set $x = \Delta_{1,2}^{(i)}$.
- $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$, $\mathcal{E}_4$: events controlling the locations of y_1 and y_2 in the analysis of Lemma 2 and Lemma 4.
- $\vec{F_i} = (F_{i,1}, F_{i,2}, \cdots F_{i,n})$, the (expected) feature of agents x_i, where

$$F_{i,t} \equiv \mathbb{E}_{R_i,R_t,\mathcal{Y}}[\mathrm{NK}(R_i, R_t) \mid \mathcal{X}] = \mathcal{G}_t(x_i).$$

- $\hat{F}_{i,t} = \mathrm{NK}(R_i^{\mathcal{O}_i}, R_t^{\mathcal{O}_t})$, an estimated version of $F_{i,t}$.
- $D(x_i, x_j) \equiv \frac{1}{n-2} \sum_{t \notin \{i,j\}} |F_{i,t} - F_{j,t}|$, the distance function based on expected feature F.
- $\mathbb{D}(x_i, x_j) \equiv \mathbb{E}\left[D(x_i, x_j) \mid x_i, x_j\right]$, the expectation of $D(x_i, x_j)$.
- $\hat{D}(x_i, x_j) \equiv \frac{1}{n-2} \left|\hat{F}_{i,t} - \hat{F}_{j,t}\right|$: our estimate of D.
- $|x_i - \mathcal{X}^*| \equiv \max_{x^* \in \mathcal{X}_x} |x_i - x^*|$: our definition of distance between arbitrary agent x_i and arbitrary set of agent $\mathcal{X}^*$.

Example 4 (Latent, realized, and observed ranking)**.** *Let* $\mathcal{X} = \{x_1\}$ *and* $\mathcal{Y} = \{y_1, \ldots, y_4\}$. *Let* $x_1 = 0$, $y_1 = 0.1$, $y_2 = 0.2$, $y_3 = 0.3$, *and* $y_4 = 0.4$.

- Latent ranking: *the latent ranking* L_i *is determined by* y_ℓ*s' utilities to* x_i, *i.e.,* $L_1 = (y_1, y_2, y_3, y_4)$.
- Realized ranking: *the realized ranking of* x_i *is* x_i*'s preferences under the random utility model. It guarantees that only* y_{ℓ_1} *is* more likely *to be ranked ahead of* y_{ℓ_2} *when* $u(x_i, y_{\ell_1}) > u(x_i, y_{\ell_2})$. *One specific realization of the ranking in our example could be* $R_1 = (y_2, y_1, y_3, y_4)$.
- Observed ranking: *we do not observe the entire* R_i. *Instead, we only observe agent* i*'s ranking over a subset of* $\mathcal{Y}$. *For example, here* $\mathcal{O}_1 = \{y_1, y_2, y_4\}$. *Then we have* $R_1^{\mathcal{O}_1} = (y_2, y_1, y_4)$.

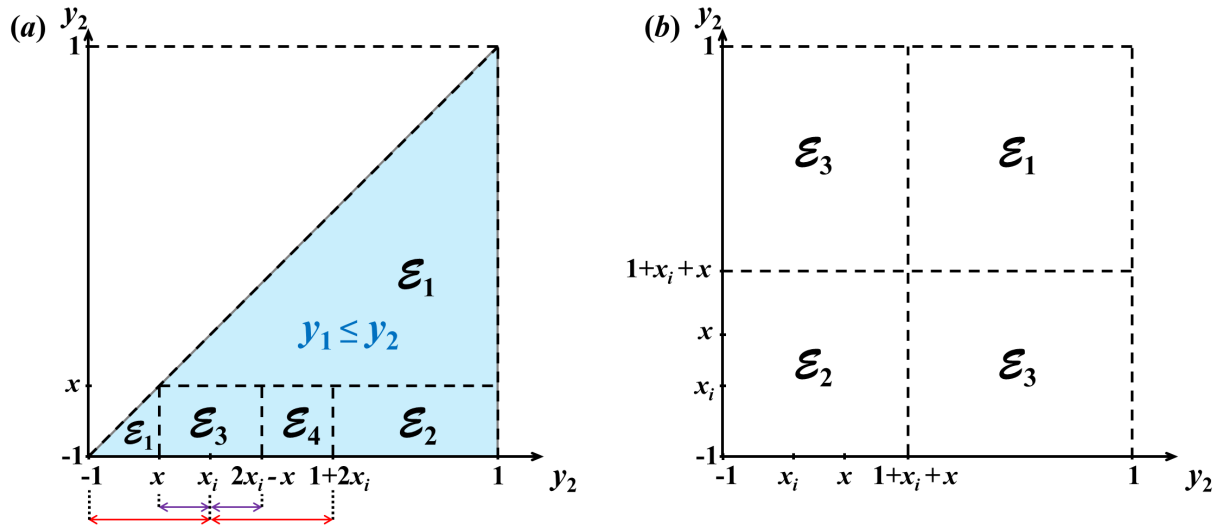

Figure A.1: Events defined in the proof of Theorem 1. (a): the events $\mathcal{E}_1, \cdots \mathcal{E}_4$ defined in Lemma 2 in Section 2.3. (b): the events defined in Appendix A.2.3 (proof of part 2).

A.2 Missing Proofs for Theorem 1 (Main Negative Result)

We have the following missing components in the proof of Theorem 1.

- Section A.2.1 proves Lemma 1(concentration of NK).
- Section A.2.2 shows the proof for Equation. 2.5 and Fact 1.
- The missing analysis for part 2 and part 3 in Lemma 2 is in Section A.2.3.
- Section A.2.4 presents how Lemma 1 and Lemma 2 together imply Theorem 1.

A.2.1 Proof of Lemma 1 (Concentration of NK)

Proof of Lemma 1. We repeat the statement below.

Lemma 8 (Repeat of Lemma 1)**.** *Let* $\mu = \mathbb{E}[\mathrm{NK}(R_i, R_j) \mid \mathcal{X}]$. *We have*

$$\Pr\left[|\mathrm{NK}(R_i, R_j) - \mu| \ge \delta\mu \mid \mathcal{X}\right] \le 4m \exp\left(-\frac{\delta^2 m\mu}{6}\right) \tag{A.4}$$

Proof of Lemma 1. Let $S = \{\{\ell_1, \ell_2\} : \ell_1 \ne \ell_2 \in [m]\}$ be the collection of all possible alternative subsets of size 2. We have

$$\mathrm{NK}(R_i, R_j) = \frac{1}{|S|} \sum_{\{\ell_1,\ell_2\} \in S} \mathrm{NK}(R_i^{\{\ell_1,\ell_2\}}, R_j^{\{\ell_1,\ell_2\}}). \tag{A.5}$$

We know that

- Each $\mathrm{NK}(R_i^{\{\ell_1,\ell_2\}}, R_j^{\{\ell_1,\ell_2\}})$ follows the same distribution for all $\{\ell_1, \ell_2\} \in S$. Furthermore,
$$\mathbb{E}[\mathrm{NK}(R_i, R_j) \mid \mathcal{X}] = \mathbb{E}[\mathrm{NK}(R_i^{\{\ell_1,\ell_2\}}, R_j^{\{\ell_1,\ell_2\}}) \mid \mathcal{X}] = \mu = \Theta(1). \tag{A.6}$$
In other words, μ is not a function of m or n.
- Since terms are not independent, we cannot directly use Chernoff bounds.

Our crucial observation is that we may partition the set S into a total number of $\Theta(m)$ subsets so that terms inside each subset are independent. Then we may apply a union bound across subsets. More specifically, define

$$S_i = \{\{\ell_1, \ell_2\} : |\ell_2 - \ell_1| = i\}.$$

for $1 \le i \le m-1$. For example:

- $S_1 = \{\{1,2\},\{2,3\},\cdots,\{m-1,m\}\}$.
- $S_2 = \{\{1,3\},\{2,4\},\cdots,\{m-2,m\}\}$.

Now we claim that we may further partition each S_i into two sets so that terms inside each set are independent.

We first demonstrate how this can be done for S_1 and S_2. Then we describe a general scheme to partition every S_i.

Partitioning S_1. We may partition S_1 into

$$S_1^{(1)} = \{\{1,2\},\{3,4\},\cdots\}$$
$$S_1^{(2)} = \{\{2,3\},\{4,5\},\cdots\}$$

One can see that $S_1^{(1)}$ and $S_1^{(2)}$ are a partition of S_1 and terms $\mathrm{NK}(R_i^{\{\ell_1,\ell_2\}}, R_j^{\{\ell_1,\ell_2\}})$ are independent for $\{\ell_1, \ell-2\} \in S_1^{(t)}$ $(t = 1, 2)$.

Partitioning S_2. We need three steps to construction the partition. First, let

$$S_{2,1} = \{\{1,3\},\{3,5\},\cdots,\{m-2,m\}\}$$
$$S_{2,2} = \{\{2,4\},\{4,6\},\cdots,\{m-3,m-1\}\}$$

Second, we may partition each of $S_{2,1}$ and $S_{2,2}$:

$$S_{2,1}^{(1)} = \{\{1,3\},\{5,7\},\cdots\} \quad \text{and} \quad S_{2,1}^{(2)} = \{\{3,5\},\{7,9\},\cdots\},$$
$$S_{2,2}^{(1)} = \{\{2,4\},\{6,8\},\cdots\} \quad \text{and} \quad S_{2,2}^{(2)} = \{\{4,6\},\{8,10\},\cdots\}$$

Third, we define

$$S_2^{(1)} = S_{2,1}^{(1)} \cup S_{2,2}^{(1)} \quad \text{and} \quad S_2^{(2)} = S_{2,1}^{(2)} \cup S_{2,2}^{(2)}$$

$S_2^{(1)}$ and $S_2^{(2)}$ are the desired partition for S_2.

The general scheme. The general scheme works as follows. First, define $S_{i,j}$ $(1 \le j \le i)$:

$$S_{i,j} = \{\{j, j+i\}, \{j+i, j+2i\}, \cdots\}.$$

Then define $S_{i,j}^{(1)}$ be the set that consists of the "odd terms" in $S_{i,j}$:

$$S_{i,j}^{(1)} = \{\{j, j+i\}, \{j+2i, j+3i\}, \cdots\}$$
$$S_{i,j}^{(2)} = \{\{j+i, j+2i\}, \{j+3i, j+4i\}, \cdots\}$$

Finally, define

$$S_{i,j}^{(1)} = \bigcup_{j \le i} S_{i,j}^{(1)} \quad \text{and} \quad S_{i,j}^{(2)} = \bigcup_{j \le i} S_{i,j}^{(2)}.$$

For each $S_i^{(t)}$ $(t = 1, 2)$, we can apply a Chernoff bound:

$$\Pr\left[\left|\frac{1}{|S^{(t)_i}|}\sum_{\{\ell_1,\ell_2\}\in S_i^{(t)}} \mathrm{NK}(R_i^{\{\ell_1,\ell_2\}}, R_i^{\{\ell_1,\ell_2\}}) - \mu\right| > \delta\mu \middle| \mathcal{X}\right] < 2\exp\left(-\frac{\delta^2\mu m}{6}\right)$$

Then we may apply a union bound across all $S_i^{(t)}$ $(i \le 2m$ and $t = 1, 2)$:

$$\Pr\left[|\mathrm{NK}(R_i, R_j) - \mu| \ge \delta\mu\right] \le 4m\exp\left(-\frac{\delta^2 m\mu}{6}\right) \tag{A.7}$$

□

□

A.2.2 Missing Proofs for the Facts

Proof for Equation 2.5. We repeat Equation. 2.5 below.

Fact 2. *Using the notations in Section 2.3.1,*

$$\frac{\partial \mathcal{G}_i(x)}{\partial x} = \mathbb{E}\left[\,\Phi(y_1, y_2, x, x_i)\middle|\, x_i, x\right], \tag{A.8}$$

where

$$\begin{cases} \Phi(y_1, y_2, x, x_i) & = \frac{e^{-\Delta_{1,2}^{(i)}}-1}{e^{-\Delta_{1,2}^{(i)}}+1} \cdot \frac{\operatorname{sign}(y_1-x)-\operatorname{sign}(y_2-x)}{4\cosh^2\left(\Delta_{1,2}^{(x)}/2\right)} \\ \Delta_{1,2}^{(i)} & = |y_2 - x_i| - |y_1 - x_i| \\ \Delta_{1,2}^{(x)} & = |y_2 - x| - |y_1 - x| \end{cases} \tag{A.9}$$

Proof. We have

$$\begin{aligned} &\frac{\partial \mathcal{G}_i(x)}{\partial x} = \mathbb{E}_{y_1,y_2}\left[(1-2p_i)\frac{\partial p_x}{\partial x}\middle| x_i, x\right] \\ &= \mathbb{E}_{y_1,y_2}\left[\frac{e^{-|x_i-y_2|} - e^{-|x_i-y_1|}}{e^{-|x_i-y_1|} + e^{-|x_i-y_2|}} \cdot \frac{[\operatorname{sign}(x-y_2) - \operatorname{sign}(x-y_1)] \cdot e^{-|x-y_2|-|x-y_1|}}{\left(e^{-|x-y_1|} + e^{-|x-y_2|}\right)^2} \mid x_i, x\right] \\ &= \mathbb{E}_{y_1,y_2}\left[\frac{e^{-\Delta_{1,2}^{(i)}} - 1}{e^{-\Delta_{1,2}^{(i)}} + 1} \cdot \frac{\operatorname{sign}(x-y_2) - \operatorname{sign}(x-y_1)}{4\cosh^2\left(\Delta_{1,2}^{(x)}/2\right)} \mid x_i, x\right], \end{aligned}$$

where $\Delta_{1,2}^{(i)} = |y_2 - x_i| - |y_1 - x_i|$ and $\Delta_{1,2}^{(x)} = |y_2 - x| - |y_1 - x|$. □

Proof for Fact 1. We repeat Fact 1 below.

Fact 3 (Repeat of Fact 1)**.** *We have*

$$\mathbb{E}[\Phi \mid x, x_i] = \sum_{t=2,3,4} \mathbb{E}[\Phi \mid x, x_i, \mathcal{E}_t] \Pr[\mathcal{E}_t \mid x, x_i] > 0. \tag{A.10}$$

Proof. Let $g(x) = \frac{e^{-x}-1}{e^{-x}+1}$. This function is closely related to the logistic function and possesses the following properties:

1. $g(x)$ is an odd function: $g(x) = -g(-x)$.
2. $g''(x) \cdot g(x) \geq 0$.
3. $|g(x)| \leq \frac{1}{2}|x|$.

Intuitively, property (ii) is used to give lower bound on $\left|g(\Delta_{1,2}^{(i)})\right|$ and property (iii) is for its upper bound.

Recall also that:

$$\begin{cases} \Phi(y_1, y_2, x, x_i) & = \frac{e^{-\Delta_{1,2}^{(i)}}-1}{e^{-\Delta_{1,2}^{(i)}}+1} \cdot \frac{\text{sign}(y_1-x)-\text{sign}(y_2-x)}{4\cosh^2\left(\Delta_{1,2}^{(x)}/2\right)} \\ \Delta_{1,2}^{(i)} & = |y_2 - x_i| - |y_1 - x_i| \\ \Delta_{1,2}^{(x)} & = |y_2 - x| - |y_1 - x| \end{cases} \tag{A.11}$$

- $\mathcal{E}_1$: when $y_2 \geq y_1 \geq x$ or $y_1 \leq y_2 \leq x$. Thus, $\Pr[\mathcal{E}_1 \mid y_1 \leq y_2] = \frac{x^2+1}{2}$.
- $\mathcal{E}_2$: when $y_1 < x$ and $y_2 \geq 1 + 2x_i$. Thus, $\Pr[\mathcal{E}_2 \mid y_1 \leq y_2] = -(x+1)x_i$.
- $\mathcal{E}_3$: when $y_1 < x$ and $x < y_2 < 2x_i - x$. Thus, $\Pr[\mathcal{E}_3 \mid y_1 \leq y_2] = (x+1)(x_i - x)$.
- $\mathcal{E}_4$: when $y_1 < x$ and $2x_i - x \leq y_2 < 1 + 2x_i$. Thus, $\Pr[\mathcal{E}_4 \mid y_1 \leq y_2] = \frac{(x+1)^2}{2}$.

We now presents our formal analysis for $\mathbb{E}[\Phi \mid x, x_i, \mathcal{E}_t]$, where $t \in \{1, 2, 3, 4\}$.

Analysis for $\mathcal{E}_1$: We always have $\text{sign}(x - y_2) - \text{sign}(x - y_1) = 0$ and thus

$$\mathbb{E}_{y_1 \leq y_2}\left[\Phi(y_1, y_2, x, x_i) \mid x, x_i, \mathcal{E}_1\right] = 0.$$

Analysis for $\mathcal{E}_2$: We observe that $\Delta_{1,2}^{(x)} \leq 2$, $\Delta_{1,2}^{(i)} \in [0, 1 - 2x_i + x] \subseteq [0, 2]$, and $\text{sign}(x - y_2) - \text{sign}(x - y_1) = -2$. Thus, we have $\Phi(y_1, y_2, x, x_i) > 0$. We have

$$\begin{aligned} \mathbb{E}_{y_1 \leq y_2}[\Phi(y_1, y_2, x, x_i) \mid x, x_i, \mathcal{E}_2] &= \mathbb{E}_{y_1 \leq y_2}\left[g(\Delta_{1,2}^{(i)}) \cdot \frac{-2}{4\cosh^2\left(\Delta_{1,2}^{(x)}/2\right)} \mid x, x_i, \mathcal{E}_2\right] \\ &\geq \mathbb{E}_{y_1 \leq y_2}\left[g\left(\Delta_{1,2}^{(i)}\right) \cdot \frac{-1}{2\cosh^2(1)}\middle| x, x_i, \mathcal{E}_2\right] \\ &= \frac{-2}{(e + e^{-1})^2} \cdot \mathbb{E}_{y_1 \leq y_2}\left[g\left(\Delta_{1,2}^{(i)}\right)\middle| x, x_i, \mathcal{E}_2\right] \\ &\geq^* \frac{-2}{(e + e^{-1})^2} \cdot \mathbb{E}_{y_1 \leq y_2}\left[\frac{1}{2} \cdot \frac{e^{-2} - 1}{e^{-2} + 1} \cdot \Delta_{1,2}^{(i)}\middle| x, x_i, \mathcal{E}_2\right] \\ &= \frac{e - e^{-1}}{2\left(e + e^{-1}\right)^3} \cdot (1 - 2x_i + x). \end{aligned}$$

The inequality $\geq^*$ relies on property 2 of the function $g(\cdot)$.

Analysis for $\mathcal{E}_3$: We observe that $\Delta_{1,2}^{(x)} \in [-(1+x), 2x_i - 2x]$, $\Delta_{1,2}^{(i)} \in [-(1+x_i), 0]$ and $\text{sign}(x-y_2) - \text{sign}(x-y_1) = -2$. Thus, we have $\Phi(y_1, y_2, x, x_i) < 0$. We have:

$$\begin{aligned}
\mathbb{E}_{y_1 \le y_2}[\Phi(y_1, y_2, x, x_i) \mid x, x_i, \mathcal{E}_3] &= \mathbb{E}_{y_1 \le y_2}\left[g\left(\Delta_{1,2}^{(i)}\right) \cdot \frac{-2}{4\cosh^2\left(\Delta_{1,2}^{(x)}/2\right)} \mid x, x_i, \mathcal{E}_3 \right] \\
&\ge \mathbb{E}_{y_1 \le y_2}\left[g\left(\Delta_{1,2}^{(i)}\right) \cdot \frac{-1}{2\cosh^2(0)} \middle| x, x_i, \mathcal{E}_3 \right] \\
&\ge^{**} -\frac{1}{2} \cdot \mathbb{E}_{y_1 \le y_2}\left[-\frac{1}{2} \cdot \Delta_{1,2}^{(i)} \mid x, x_i, \mathcal{E}_3 \right] = -\frac{1+x_i}{8}.
\end{aligned}$$

The inequality $\ge^{**}$ uses on property 3 of the function $g(\cdot)$.

Analysis For $\mathcal{E}_4$: we observe that $\Delta_{1,2}^{(x)} \in [2x_i - 3x - 1, 1 + 2x_i - x] \subseteq [0, 2 + 2x_i]$, $\Delta_{1,2}^{(i)} \in [-(1+x), 1+x]$ and $\text{sign}(x-y_2) - \text{sign}(x-y_1) = -2$. Let $y_1' = 1 + 2x_i - y_2$ and $y_2' = 1 + 2x_i - y_1$. Observe that $y_1' < x$ and $2x_i - x \le y_2' < 1 + 2x_i$. Thus, we have $\Delta_{1',2'}^{(i)} = -\Delta_{1,2}^{(i)}$ where $\Delta_{1',2'}^{(i)} \equiv |y_2' - x_i| - |y_1' - x_i|$. Applying properties (i) and (iii) of $g\left(\Delta_{1,2}^{(i)}\right)$, we have,

$$\begin{aligned}
&\mathbb{E}_{y_1 \le y_2}[\Phi(y_1, y_2, x, x_i) \mid x, x_i, \mathcal{E}_4] \\
&= \frac{1}{2} \cdot \mathbb{E}_{y_1 \le y_2}\left[\Phi(y_1, y_2, x, x_i) + \Phi(y_1', y_2', x, x_i) \mid x, x_i, \mathcal{E}_4, \Delta_{1,2}^{(i)} \ge 0 \right] \\
&= \frac{1}{2} \cdot \mathbb{E}_{y_1 \le y_2}\left[g\left(\Delta_{1,2}^{(i)}\right) \cdot \left(\frac{-2}{4\cosh^2\left(\Delta_{1,2}^{(x)}/2\right)} - \frac{-2}{4\cosh^2\left(\Delta_{1',2'}^{(x)}/2\right)} \right) \mid x, x_i, \mathcal{E}_4, \Delta_{1,2}^{(i)} \ge 0 \right] \\
&\ge \frac{1}{2} \cdot \mathbb{E}_{y_1 \le y_2}\left[g\left(\Delta_{1,2}^{(i)}\right) \cdot \left(\frac{-2}{4\cosh^2(0)} - \frac{-2}{4\cosh^2(1+x_i)} \right) \mid x, x_i, \mathcal{E}_4, \Delta_{1,2}^{(i)} \ge 0 \right] \\
&\ge \frac{1+x}{8}\left(\frac{1}{\cosh^2(1+x_i)} - 1 \right).
\end{aligned}$$

Summing up $\mathcal{E}_1 \cdots \mathcal{E}_4$, for $-1 < x \le x_i \le -0.5$, we have

$$\begin{aligned}
&\mathbb{E}_{y_1 \le y_2}[\Phi(y_1, y_2, x, x_i) \mid x, x_i] = \sum_{1 \le j \le 4} \mathbb{E}_{y_1 \le y_2}[\Phi \mid \mathcal{E}_j, x_i, x] \cdot \Pr[\mathcal{E}_j, x_i, x \mid y_1 \le y_2] \\
&\ge 0 \cdot \Pr[\mathcal{E}_1, x_i, x \mid y_1 \le y_2] + \left(\frac{e - e^{-1}}{2\left(e + e^{-1}\right)^3} \cdot (1 - 2x_i + x) \right) \cdot [-x_i \cdot (x+1)] \\
&\quad - \frac{1 + x_i}{8} \cdot [(x+1)(x_i - x)] + \left[\frac{1+x}{8} \left(\frac{1}{\cosh^2(1 + x_i)} - 1 \right) \right] \cdot \frac{(1+x)^2}{2} \\
&\ge (x+1) \left[\frac{e - e^{-1}}{(e + e^{-1})^3} \cdot x_i^2 + \frac{(1+x_i)^2}{8} \left(\frac{1}{\cosh^2(1+x_i)} - 1 \right) \right] \\
&= c_1(x_i) \cdot (x+1) > 0.
\end{aligned}$$

□

A.2.3 Missing Analysis for Part 2 and Part 3 in Lemma 2

Analysis for Part 2. The main result for part 2 analysis in Lemma 2 can be summarized by the following fact.

Fact 4. *$\mathcal{G}_i(x) - \mathcal{G}_i(x_i) \ge 0$ if $x_i \in [-1, 0]$ and $x \in [x_i, -x_i]$.*

Proof. The technique we use is similar to part 1 of the analysis. Specifically, we define the following events and analyze the behavior of $\mathcal{G}_i(x)$ conditioned on these events(see Figure A.1):

- $\mathcal{E}_1$: when $y_1, y_2 \ge x + x_i + 1$.
- $\mathcal{E}_2$: when $y_1, y_2 < x + x_i + 1$.
- $\mathcal{E}_3$: when $y_1 \ge x + x_i + 1 \ge y_2$ or $y_2 \ge x + x_i + 1 \ge y_1$.

For any $\ell \in \{1, 2, 3\}$, we define $\mathcal{G}_i(x) \mid \mathcal{E}_\ell \equiv \mathbb{E}_{y_1, y_2}\left[p_x(1 - p_i) + (1 - p_x)p_i \mid x_i, x, \mathcal{E}_\ell\right]$

For $\mathcal{E}_1$, we have,

$$\begin{aligned}
[\mathcal{G}_i(x) - \mathcal{G}_i(x_i)] \mid \mathcal{E}_1 &= \mathbb{E}_{y_1, y_2}\left[p_x(1 - p_i) + (1 - p_x)p_i - 2p_i(1 - p_i) \mid x, x_i, \mathcal{E}_1\right] \\
&= \mathbb{E}_{y_1, y_2}[(1 - 2p_i)(p_x - p_i) \mid x, x_i, \mathcal{E}_1].
\end{aligned}$$

Note that conditioned on $\mathcal{E}_1$, $y_1, y_2 \ge x \ge x_i$ for $\mathcal{E}_1$. Therefore,

$$p_x = \frac{e^{-(y_1 - x)}}{e^{-(y_1 - x)} + e^{-(y_2 - x)}} = \frac{e^{-(y_1 - x_i)}}{e^{-(y_1 - x_i)} + e^{-(y_2 - x_i)}} = p_i.$$

Thus, we have $[\mathcal{G}_i(x) - \mathcal{G}_i(x_i)] \mid \mathcal{E}_1 = 0$.

For $\mathcal{E}_2$, where $y_1, y_2 \in [-1, x + x_i + 1)$, we define $y_1' = (x + x_i + 1) - y_1$ and $y_2' = (x + x_i + 1) - y_2$. We may assume that $y_1 \geq y_2$ wlog. We always have $y_1' \leq y_2'$ and $y_1', y_2' \in [-1, x + x_i + 1]$. By further defining $p_i' = \Pr[y_1' \succ_i y_2' \mid x_i]$ and $p_x' = \Pr[y_1' \succ_x y_2' \mid x]$, we have $p_i' = p_x$, $p_x' = p_i$.Then, we leverage the symmetry property and cancel some terms in $\mathcal{E}_i$. Because $\mathcal{D}_Y$ is uniform, we can always map y_1 to y_1' and map y_2 to y_2' when calculating expecation. Thus,

$$\begin{aligned}
&\mathcal{G}_i(x) \mid \mathcal{E}_2 \\
&= \frac{1}{2}\Big(\mathbb{E}_{\boldsymbol{y_1 \geq y_2}}\left[p_i(1-p_x) + (1-p_i)p_x \mid x_i, x, \mathcal{E}_2\right] \\
&\qquad + \mathbb{E}_{y_1 \leq y_2}\left[p_i(1-p_x) + (1-p_i)p_x \mid x_i, x, \mathcal{E}_2\right]\Big) \\
&\qquad + \frac{1}{2}\Big(\mathbb{E}_{\boldsymbol{y_1 \leq y_2}}\left[p_i'(1-p_x') + (1-p_i')p_x' \mid x_i, x, \mathcal{E}_2\right] \\
&\qquad + \mathbb{E}_{y_1 \leq y_2}\left[p_i(1-p_x) + (1-p_i)p_x \mid x_i, x, \mathcal{E}_2\right]\Big) \\
&= \mathbb{E}_{y_1 \leq y_2}\left[p_i + (1-2p_i)p_x \mid \mathcal{E}_2\right].
\end{aligned} \tag{A.12}$$

Similarly, we have

$$\mathcal{G}_i(x_i) \mid \mathcal{E}_2 = \mathbb{E}_{y_1 \leq y_2}\left[p_i(1-p_i) + p_x(1-p_x) \mid \mathcal{E}_2\right]. \tag{A.13}$$

Combining Equation A.12 and A.13, we have:

$$\begin{aligned}
&\mathbb{E}_{y_1, y_2}\left[\mathcal{G}_i(x) - \mathcal{G}_i(x_i) \mid x, x_i, \mathcal{E}_2\right] \\
&= \mathbb{E}_{y_1 \leq y_2}\left[p_i + (1-2p_i)p_x \mid \mathcal{E}_2\right] - \mathbb{E}_{y_1 \leq y_2}\left[p_i(1-p_i) + p_x(1-p_x) \mid \mathcal{E}_2\right] \\
&= \mathbb{E}_{y_1 \leq y_2}\left[p_i^2 - 2p_x p_i + p_x^2 \mid \mathcal{E}_2\right] \geq 0.
\end{aligned}$$

For $\mathcal{E}_3$, we have

$$\begin{aligned}
\mathbb{E}_{y_1, y_2}\left[\mathcal{G}_i(x) - \mathcal{G}_i(x_i) \mid x, x_i, \mathcal{E}_3\right] &= \mathbb{E}_{y_1, y_2}\left[p_x(1-p_i) + (1-p_x)p_i - 2p_i(1-p_i) \mid x, x_i, \mathcal{E}_3\right] \\
&= \mathbb{E}_{y_1, y_2}[(1-2p_i)(p_x - p_i) \mid x, x_i, \mathcal{E}_3].
\end{aligned}$$

When $y_2 \geq x + x_i + 1 \geq y_1$, we have $|y_1 - x_i| \leq |y_2 - x_i|$ and thus $1 - 2p_i \leq 0$. Then,

$$p_i = \frac{e^{-|x_i - y_1|}}{e^{-|x_i - y_1|} + e^{-|y_2 - x_i|}} = \frac{e^{-(|x_i - y_1| - |x - x_i|)}}{e^{-(|x_i - y_1| - |x - x_i|)} + e^{-|y_2 - x|}} \geq \frac{e^{-|x - y_1|}}{e^{-|x - y_1|} + e^{-|y_2 - x_i|}} = p_x.$$

Thus, we have $\mathbb{E}_{y_1 \leq y_2}\left[\mathcal{G}_i(x) - \mathcal{G}_i(x_i) \mid x, x_i, \mathcal{E}_3\right] \geq 0$. We may argue that

$$\mathbb{E}_{y_1 \geq y_2}\left[\mathcal{G}_i(x) - \mathcal{G}_i(x_i) \mid x, x_i, \mathcal{E}_3\right] \geq 0$$

using similar techniques. Combining all results of $\mathcal{E}_3$, we have,

$$\mathbb{E}_{y_1, y_2}\left[\mathcal{G}_i(x) - \mathcal{G}_i(x_i) \mid x, x_i, \mathcal{E}_3\right] \geq 0.$$

Fact 4 follows by combining the results of $\mathcal{E}_1, \mathcal{E}_2$ and $\mathcal{E}_3$. □

Analysis for Part 3. The main result for part 3 analysis in Lemma 2 can be summarized by the following fact.

Fact 5. $\frac{\partial \mathcal{G}_i(x)}{\partial x} > 0$ *if* $x_i \in [-1, -0.5]$ *and* $x \in (-x_i, 1)$.

Proof. This can be proved by a direct application of Fact 4. □

A.2.4 Lemma 1 and Lemma 2 Imply Theorem 1

Now we may use a standard argument to prove Theorem 1.

Proof of Theorem 1. Let $C = \max\left\{\frac{k \log^2 n}{n}, \frac{\log^2 n}{\sqrt{m}}\right\}$. By using a simple Chernoff bound, one may check that any sub-interval $\in [-1, 1]$ of size C will contain at least k agents with high probability (the probability is $1 - \exp(-\Theta(\log^2 n))$. Define

$$\begin{aligned} \mathcal{I}_1 &= [-1, -1 + C] \\ \mathcal{I}_2 &= [-1 + (\log n)C, 1]. \end{aligned} \tag{A.14}$$

Using Lemma 1 and the fact that $\mathcal{G}_i'(x)$ is bounded away from 0 when $x \in [-1, x_i]$, One can see that conditioned on $\mathcal{X}$, with probability $1 - \exp(-\Theta(\log^2 n))$:

$$\mathrm{NK}(R_i, R_{j_1}) \leq \mathrm{NK}(R_i, R_{j_2}) \tag{A.15}$$

when $x_{j_1} \in \mathcal{I}_1$ and $x_{j_2} \in \mathcal{I}_2$.

This means that no agents in $\mathcal{I}_2$ will be part of the k agents output by KT-kNN. In other words, all the agents returned by KT-kNN are in $\mathcal{I}_1$. Using the fact that $C \to 0$ when $n, m \to \infty$, we complete our proof.

□

A.2.5 A Concrete and Small Negative Example

Below is a small example illustrating the poor performance of KT-kNN.

Example 5. *We consider 1-dimensional latent space such that* $\mathcal{X} = \{x_1, x_2, x_3\}$ *and* $\mathcal{Y} = \{y_1, y_2\}$*, where* $x_1 = 0.2$*,* $x_2 = 0.2, x_3 = -1, y_1 = 0$ *and* $y_2 = 1$*. We have Let* $\mathcal{F}$ *be the* σ*-algebra generated by* $\mathcal{X}$ *and* $\mathcal{Y}$*.*

$$\begin{aligned}
&\mathbb{E}_{R_1,R_2}\left[\mathrm{NK}(R_1,R_2) \mid \mathcal{X},\mathcal{Y}\right] \\
&= \Pr[y_1 \succ_1 y_2 \mid \mathcal{F}] \cdot \Pr[y_2 \succ_2 y_1 \mid \mathcal{F}] + \Pr[y_2 \succ_1 y_1 \mid \mathcal{F}] \cdot \Pr[y_1 \succ_2 y_2 \mid \mathcal{F}] \\
&= \frac{e^{-0.2}}{e^{-0.2}+e^{-0.8}} \cdot \frac{e^{-0.8}}{e^{-0.2}+e^{-0.8}} + \frac{e^{-0.8}}{e^{-0.2}+e^{-0.8}} \cdot \frac{e^{-0.2}}{e^{-0.2}+e^{-0.8}} \approx 0.4576.
\end{aligned} \tag{A.16}$$

$$\begin{aligned}
&\mathbb{E}_{R_1,R_3}\left[\mathrm{NK}(R_1,R_3) \mid \mathcal{X},\mathcal{Y}\right] \\
&= \Pr[y_1 \succ_1 y_2 \mid \mathcal{F}] \cdot \Pr[y_2 \succ_3 y_1 \mid \mathcal{F}] + \Pr[y_2 \succ_1 y_1 \mid \mathcal{F}] \cdot \Pr[y_1 \succ_3 y_2 \mid \mathcal{F}] \\
&= \frac{e^{-0.2}}{e^{-0.2}+e^{-0.8}} \cdot \frac{e^{-2}}{e^{-2}+e^{-1}} + \frac{e^{-0.8}}{e^{-0.2}+e^{-0.8}} \cdot \frac{e^{-1}}{e^{-1}+e^{-2}} \approx 0.4327.
\end{aligned} \tag{A.17}$$

The above analysis shows KT-kNN *"thinks"* x_3 *is closer to* x_1 *than* x_2*, who is the real nearest neighbor of* x_1

A.3 Missing Proofs for Theorem 2 (Main Positive Results)

A.3.1 Missing Proof for Lemma 3

We first prove Eq. 2.8. Specifically, we need:

Fact 6. *For any* $y_{\ell_1}, y_{\ell_2} \in \mathcal{Y}$ *and* $x_i, x_j \in \mathcal{X}$*, we have* $\left|\Pr[y_{\ell_1} \succ_i y_{\ell_2}] - \Pr[y_{\ell_1} \succ_j y_{\ell_2}]\right| \leq ||x_i - x_j||_2$.

Proof of Fact 6. We first give an upper bound on $\Pr[y_{\ell_1} \succ_i y_{\ell_2}] - \Pr[y_{\ell_1} \succ_j y_{\ell_2}]$,

$$\begin{aligned}
&\Pr[y_{\ell_1} \succ_i y_{\ell_2}] - \Pr[y_{\ell_1} \succ_j y_{\ell_2}] \\
=& \frac{e^{-||x_i - y_{\ell_1}||_2}}{e^{-||x_i - y_{\ell_1}||_2} + e^{-||x_i - y_{\ell_2}||_2}} - \frac{e^{-||x_j - y_{\ell_1}||_2}}{e^{-||x_j - y_{\ell_2}||_2} + e^{-||x_j - y_{\ell_1}||_2}} \\
\leq& \frac{e^{-||x_i - y_{\ell_1}||_2}}{e^{-||x_i - y_{\ell_1}||_2} + e^{-||x_i - y_{\ell_2}||_2}} - \frac{e^{-||x_i - y_{\ell_1}||_2 - |x_i - x_j|}}{e^{-||x_i - y_{\ell_1}||_2 - ||x_i - x_j||_2} + e^{-||x_i - y_{\ell_2}||_2 + ||x_i - x_j||_2}} \\
=& \frac{e^{-||x_i - y_{\ell_1}||_2 - ||x_i - y_{\ell_2}||_2} \left(e^{2||x_i - x_j||_2} - 1\right)}{e^{-||x_i - y_{\ell_1}||_2 - ||x_i - y_{\ell_2}||_2} \left(e^{2||x_i - x_j||_2} + 1\right) + e^{-2||x_i - y_{\ell_1}||_2} + e^{-2||x_i - y_{\ell_2}||_2 + 2||x_i - x_j||_2}} \\
\leq& \frac{e^{2||x_i - x_j||_2} - 1}{e^{2||x_i - x_j||_2} + 1} \leq ||x_i - x_j||_2,
\end{aligned}$$

where the last inequality holds because $\frac{e^x - 1}{e^x + 1} \leq \frac{x}{2}$ for any $x \geq 0$. Similarly, for the lower bound, we have

$$\begin{aligned}
&\Pr[y_{\ell_1} \succ_i y_{\ell_2}] - \Pr[y_{\ell_1} \succ_j y_{\ell_2}] \\
=& \frac{e^{-||x_i - y_{\ell_1}||_2}}{e^{-||x_i - y_{\ell_1}||_2} + e^{-||x_i - y_{\ell_2}||_2}} - \frac{e^{-||x_j - y_{\ell_1}||_2}}{e^{-||x_j - y_{\ell_1}||_2} + e^{-||x_j - y_{\ell_2}||_2}} \\
\geq& \frac{e^{-||x_i - y_{\ell_1}||_2}}{e^{-||x_i - y_{\ell_1}||_2} + e^{-||x_i - y_{\ell_2}||_2}} - \frac{e^{-||x_i - y_{\ell_1}||_2 + ||x_i - x_j||_2}}{e^{-||x_i - y_{\ell_1}||_2 + ||x_i - x_j||_2} + e^{-||x_i - y_{\ell_2}||_2 - ||x_i - x_j||_2}} \\
=& \frac{e^{-||x_i - y_{\ell_1}||_2 - ||x_i - y_{\ell_2}||_2} \left(e^{-2||x_i - x_j||_2} - 1\right)}{e^{-||x_i - y_{\ell_1}||_2 - ||x_i - y_{\ell_2}||_2} \left(e^{-2||x_i - x_j||_2} + 1\right) + e^{-2||x_i - y_{\ell_1}||_2} + e^{-2||x_i - y_{\ell_2}||_2 - 2||x_i - x_j||_2}} \\
\geq& \frac{e^{-2||x_i - x_j||_2} - 1}{e^{-2||x_i - x_j||_2} + 1} \geq -||x_i - x_j||_2.
\end{aligned}$$

Fact 6 follows. □

Fact 7. *Using the notations in Lemma 3, we have*

$$\mathbb{E}_{y_1 \leq y_2}[\Phi \mid \mathcal{E}_2, x, x_t] = \Omega(1). \tag{A.18}$$

Proof. Note that $x \geq 2x_t + c$ or $x_t \leq \frac{x-c}{2}$. Conditioned on $\mathcal{E}_2$, we have $\Delta_{1,2}^{(t)} \geq c - \frac{3(x+c)}{8} \geq \frac{c}{4}$,

$\Delta_{1,2}^{(x)} \leq 2c$. We now get

$$\begin{aligned}
\mathbb{E}_{y_1 \leq y_2}[\Phi \mid \mathcal{E}_2] &= \mathbb{E}_{y_1 \leq y_2}\left[\frac{e^{-\Delta_{1,2}^{(t)}}-1}{e^{-\Delta_{1,2}^{(t)}}+1} \cdot \frac{-2}{4\cosh^2(\Delta_{1,2}^{(x)}/2)} \mid \mathcal{E}_2\right] \\
&\geq \mathbb{E}_{y_1 \leq y_2}\left[\frac{e^{-c/4}-1}{e^{-c/4}+1} \cdot \frac{-2}{4\cosh^2(c)} \mid \mathcal{E}_2\right] \\
&= \frac{1-e^{-c/4}}{1+e^{-c/4}} \cdot \frac{1}{2\cosh^2(c)} = \Omega(1).
\end{aligned}$$

□

A.3.2 Concentration Among $\mathbb{D}$, D, and $\hat{D}$ Functions

This section presents concentration results ($\hat{D}$ concentrates at D and D concentrates at $\mathbb{D}$). We aim to design a concentration bound so that we can use $\tau(n)$ (quality requirement of the kNN problem) to determine the sample complexity (the size of n and m). Then we may use n, m, and $\tau(n)$ to determine the number k of neighbors we can keep.

We need to use Chernoff bounds twice. First, we need to bound the difference between $\mathbb{D}(x_i, x_j)$ and $D(x_i, x_j)$. This bound is on the number of agents (i.e., we need sufficient number of agents to get concentration). We use a multiplicative Chernoff bound here because additive bounds require an extra (and unnecessary) effort to compute the variance of $\mathbb{D}(x_i, x_j)$.

Second, we need to bound the difference between $D(x_i, x_j)$ and $\hat{D}(x_i, x_j)$. The difference between $\hat{D}(x_i, x_j)$ and $D(x_i, x_j)$ comes from the extra randomness of $\hat{F}_{i,t}$ (or NK $\left(R_i^{\mathcal{O}_i}, R_j^{\mathcal{O}_j}\right)$). We first derive an tail bound of NK for partial observation ($p \leq 1$) from our tail bound for full observation ($p = 1$, Lemma 1). Then, we apply trivial union bound on $\hat{F}$ to get the second bound.

Concentration between $\mathbb{D}(x_i, x_j)$ and $D(x_i, x_j)$.

We use the standard chernoff bound:

$$\Pr\left[|D(x_i, x_j) - \mathbb{D}(x_i, x_j)| \geq \delta_1' \mathbb{D}(x_i, x_j)\right] \leq 2\exp\left(-\frac{(\delta_1')^2 \mathbb{D}(x_i, x_j)}{3} \cdot n\right). \tag{A.19}$$

Set δ_1 as $\delta_1' \cdot \mathbb{D}(x_i, x_j)$. We have

$$\Pr\left[|D(x_i, x_j) - \mathbb{D}(x_i, x_j)| \geq \delta_1\right] \leq 2\exp\left(-\frac{\delta_1^2}{3\mathbb{D}(x_i, x_j)} \cdot n\right) \leq 2\exp\left(-\frac{\delta_1^2}{3} \cdot n\right). \tag{A.20}$$

Concentration between $D(x_i, x_j)$ and $\hat{D}(x_i, x_j)$. First, we recall the (full-observation) concentration bound for the NK function in Lemma 1:

$$\Pr\left[|\mathrm{NK}(R_i, R_t) - F_{i,t}| \geq \delta_2 F_{i,t}\right] \leq 4m \exp\left(-\frac{\delta_2^2 F_{i,t}}{6} \cdot m\right). \tag{A.21}$$

and produce a bound for partial-observation. Our procedure consists of three steps: *Step 1.* show that number of alternatives ranked by both agents is $\Omega(p^2 m)$ with high probability, *step 2.* give a tail bound on $\hat{F}$ conditioned on both agents rank a subset of $\Omega(p^2 m)$ alternatives, and *step 3.* combine results from step 1 and 2, then apply a union bound.

Step 1. We first show that $m_{i,t} \equiv |\mathcal{O}_i \cap \mathcal{O}_j|$, the number of alternatives ranked by both x_i and x_t, is no less than $p^2 m/2$ with high probability. In our setting: both x_i and x_j ranks any specific alternative with probability p (independently). We apply Chernoff bound and have

$$\Pr\left[m_{i,t} \geq \frac{p^2 m}{2}\right] \geq 1 - \exp\left(-\frac{p^2}{8} \cdot m\right). \tag{A.22}$$

Step 2. Because the normalized Kendall-tau distance only considers the alternatives ranked by both agents, we have

$$\begin{aligned} \Pr\left[\left|\hat{F}_{i,t} - F_{i,t}\right| \geq \delta_2 F_{i,t}\right] &= \Pr\left[\left|\mathrm{NK}\left(R_i^{\mathcal{O}_i}, R_t^{\mathcal{O}_t}\right) - F_{i,t}\right| \geq \delta_2 F_{i,t}\right] \\ &\leq 4m_{i,t} \exp\left(-\frac{\delta_2^2 F_{i,t}}{6} \cdot m_{i,t}\right). \end{aligned} \tag{A.23}$$

Note that $m_{i,t}$ is a random variable and we give the tail bound to $\hat{F}_{i,t}$ on the condition of $m_{i,t} \geq \frac{p^2 m}{2}$,

$$\Pr\left[\left|\hat{F}_{i,t} - F_{i,t}\right| \geq \delta_2 F_{i,t} \mid m_{i,t} \geq \frac{p^2 m}{2}\right] \leq 2p^2 m \exp\left(-\frac{\delta_2^2 F_{i,t}}{12} \cdot p^2 m\right). \tag{A.24}$$

Step 3. Combining the conditional tail bound of $\hat{F}$ and the tail bound of $m_{i,t}$, we have,

$$\Pr\left[\left|\hat{F}_{i,t} - F_{i,t}\right| \geq \delta_2 F_{i,t}\right] \leq 2p^2 m \exp\left(-\frac{\delta_2^2 F_{i,t}}{12} \cdot p^2 m\right) + \exp\left(-\frac{p^2}{8} \cdot m\right). \tag{A.25}$$

Then, we give a trivial bounds to $F_{i,t}$ to simplify the bound. Because normalized Kendall-tau distance is defined as "the ratio of disagreeing pairs", we know $F_{i,t} \equiv \mathbb{E}[\mathrm{NK}(R_i, R_t) \mid \mathcal{X}] \leq 1$.

Recall that $p_i \equiv \Pr[y_1 \succ_i y_2]$ and $p_t \equiv \Pr[y_1 \succ_t y_2]$, we have

$$\begin{aligned} F_{i,t} &\equiv \mathbb{E}[\mathrm{NK}(R_i, R_t) \mid \mathcal{X}] = \mathbb{E}_{y_1,y_2}[p_i(1-p_t) + p_t(1-p_i) \mid x_i, x_t] \\ &\geq \min_{x_i,x_t \in [-c,c]} p_i(1-p_t) + p_t(1-p_i) = \frac{2e^{-2c}}{(1+e^{-2c})^2}. \end{aligned} \tag{A.26}$$

Applying this trivial bound on $F_{i,t}$ to the tail bound of $\hat{F}_{i,t}$, we have,

$$\Pr\left[\left|\hat{F}_{i,t} - F_{i,t}\right| \geq \frac{\delta_2}{2}\right] \leq 2p^2 m \exp\left(-c_5\delta_2^2 \cdot p^2 m\right) + \exp\left(-\frac{1}{8} \cdot p^2 m\right). \tag{A.27}$$

where c_5 is a positive constant that depends only on c. Then, we recall the definition of $D(x_i, x_j), \hat{D}(x_i, x_j)$ and have,

$$\begin{aligned} &\Pr\left[\left|\hat{D}(x_i, x_j) - D(x_i, x_j)\right| \geq \delta_2 \mid \mathcal{X}\right] \\ =\ &\Pr\left[\frac{1}{n-2}\sum_{t \notin \{i,j\}} \left||\hat{F}_{i,t} - \hat{F}_{j,t}| - |F_{i,t} - F_{j,t}|\right| \geq \delta_2 \mid \mathcal{X}\right]. \end{aligned} \tag{A.28}$$

By applying a simple union bound, we have,

$$\begin{aligned} &\Pr\left[\left|\hat{D}(x_i, x_j) - D(x_i, x_j)\right| \geq \delta_2 \mid \mathcal{X}\right] \\ &\leq 4p^2 mn \exp\left(-c_5\delta_2^2 \cdot p^2 m\right) + 2n \exp\left(-\frac{1}{8} \cdot p^2 m\right). \end{aligned} \tag{A.29}$$

(A.20) and (A.29) give us the following lemma:

Lemma 9. *Using the notations in Section 2.4,*

$$\begin{aligned} &\Pr\left[\left|\hat{D}(x_i, x_j) - \mathbb{D}(x_i, x_j)\right| \geq \delta_1 + \delta_2 \mid \mathcal{X}\right] \\ &\leq 4p^2 mn \exp\left(-c_5\delta_2^2 \cdot p^2 m\right) + 2n \exp\left(-\frac{1}{8} \cdot p^2 m\right) + 2\exp\left(-\frac{\delta_1^2}{3} \cdot n\right). \end{aligned}$$

A.3.3 Proof of Theorem 2

Now we can use Lemma 9 to prove Theorem 2. Let $\mathcal{X}_k^*$ denote the set of all k (ground-truth) nearest neighbors of x_i in terms of latent distance. In the next two corollaries (derived from Lemma 9), we give an upper bound to $\hat{D}(x_i, x_j)$ for any $x_j \in \mathcal{X}_k^*$ and give a lower bound on for any x_j such that $|x_i - x_j| \geq \tau(n)$. Once the lower bound is larger or equal to the

upper bound, one can see that no agent further than $\tau(n)$ will be returned by Anchor-kNN

Specifically, Lemma 9 leads to the following corollaries:

Corollary 1. *Under the same setting of Theorem 2, we have*

$$\Pr\left[\left(\max_{x_j\in\mathcal{X}_k^*}\hat{D}(x_i,x_j)\right)\le\frac{\tau^2(n)}{8c_Xc\cdot\ln n}\right]\ge 1-o(1). \tag{A.30}$$

Proof. This proof consists of two steps: *Step 1.* show all k (real) nearest neighbors are close to x_i with high probability, and *Step 2.* apply Lemma 9 to get an upper bound of $\hat{D}(x_i,x_j)$.

We need more notations. Let $|x_i-\mathcal{X}^*|=\max_{x_j\in\mathcal{X}^*}|x_i-x_j|$. We first show $|x_i-\mathcal{X}_k^*|\le 2c_Xc\cdot\frac{k}{n}$ with high probability. Let random variable n_{near} be the number of agents in $\left[x_i-2c_Xc\cdot\frac{k}{n},x_i+2c_Xc\cdot\frac{k}{n}\right]$. We note that $\mathbb{E}\left[n_{\text{near}}\right]\ge 2k$, and by a Chernoff bound, we have

$$\Pr\left[n_{\text{near}}\ge k\right]\ge 1-\exp\left(-k/4\right).$$

Also, when $n_{\text{near}}\ge k$, we know all k (real) nearest neighbors are no further than $2c_Xc\cdot\frac{k}{n}$. We have

$$\Pr\left[|x_i-\mathcal{X}_k^*|\le 2c_Xc\cdot\frac{k}{n}\right]\ge 1-\exp\left(-k/4\right).$$

Recall that $\mathbb{D}(x_i,x_j)\le|x_i-x_j|$ (Lemma 3), we have

$$\Pr\left[\left(\max_{x_j\in\mathcal{X}_k^*}\mathbb{D}(x_i,x_j)\right)\le 2c_Xc\cdot\frac{k}{n}\right]\ge 1-\exp\left(-k/4\right).$$

Using that $2c_Xc\cdot\frac{k}{n}=o\left(\tau^2(n)\cdot\ln^{-1}n\right)$, we have

$$\Pr\left[\left(\max_{x_j\in\mathcal{X}_k^*}\mathbb{D}(x_i,x_j)\right)>\frac{\tau^2(n)}{72c_Xc\cdot\ln n}\right]\le\exp\left(-k/4\right). \tag{A.31}$$

Recalling the conclusion of Lemma 9 and letting $\delta_1=\delta_2=\delta/2$, we have

$$\begin{aligned}&\Pr\left[\left|\hat{D}(x_i,x_j)-\mathbb{D}(x_i,x_j)\right|\ge\delta\mid\mathcal{X}\right]\\&\le 4p^2mn\exp\left(-\frac{c_5\delta^2}{4}\cdot p^2m\right)+2n\exp\left(-\frac{1}{8}\cdot p^2m\right)+2\exp\left(-\frac{\delta^2}{12}\cdot n\right).\end{aligned}$$

Letting $\delta = \frac{\tau^2(n)}{9c_X c \cdot \ln n}$, we have

$$\begin{aligned}
&\Pr\left[\left|\hat{D}(x_i,x_j) - \mathbb{D}(x_i,x_j)\right| \geq \frac{\tau^2(n)}{9c_X c \cdot \ln n} \mid \mathcal{X}\right] \\
&\leq 4p^2mn \exp\left(-\frac{c_5}{324c_X^2c^2} \cdot \frac{p^2m \cdot \tau^4(n)}{\ln^2 n}\right) + 2n\exp\left(-\frac{1}{8} \cdot p^2m\right) \\
&\qquad + 2\exp\left(-\frac{1}{972c_X^2c^2} \cdot \frac{n \cdot \tau^4(n)}{\ln^2 n}\right).
\end{aligned} \tag{A.32}$$

Combining (A.31-A.32) through union bound, we have

$$\begin{aligned}
&\Pr\left[\left(\max_{x_j \in \mathcal{X}_k^*} \hat{D}(x_i,x_j)\right)\right] \\
&\geq 4p^2mn \exp\left(-\frac{c_5}{324c_X^2c^2} \cdot \frac{p^2m \cdot \tau^4(n)}{\ln^2 n}\right) + 2n\exp\left(-\frac{1}{8} \cdot p^2m\right) \\
&\quad + 2\exp\left(-\frac{1}{54c_X c} \cdot \frac{n \cdot \tau^2(n)}{\ln n}\right) + \exp(-k/4).
\end{aligned}$$

When $m = \omega\left(\frac{\ln^3 n}{p^2 \cdot \tau^4(n)} \cdot \ln^2\left(\frac{\ln^3 n}{p^2 \cdot \tau^4(n)}\right)\right)$, $\tau(n) = \omega\left(\sqrt{\ln n \cdot n^{-1/2}}\right)$, $n = \omega(1)$, and $k = \omega(1)$, all four terms will become $o(1)$ and Corollary 1 follows. □

Using a similar analysis, we also have

Corollary 2. *Under the same setting of Theorem 2, we have*

$$\Pr\left[\left(\min_{|x_i - x_j| \geq \tau(n)} D(x_i,x_j)\right) \geq \frac{\tau^2(n)}{8c_X c \cdot \ln n}\right] \geq 1 - o(1). \tag{A.33}$$

Theorem 2 follows from Corollary 1 and Corollary 2.

A.4 Near Neighbor (NN) and Preference Completion (PC)

This section examines the relationship between the near neighbor problem and the preference completion problem.

The preference completion problem uses the partial rankings $R_i^{\mathcal{O}_i}$ of all agents to infer the ground-truth rankings. We define the latent/ground-truth ranking of agent i to be $L_i = (y_{j_1}, \cdots, y_{j_m})$ such that $\theta(x_i, y_{j_1}) \geq \theta(x_i, y_{j_2}) \geq \cdots \geq \theta(x_i, y_{j_m})$, where the ties are broken arbitrarily. We focus on recovering L_i instead of R_i because R_i can be viewed as noisy observations of the true signal L_i.

We now explain how any $k(n)$-NN solver (defined in Section 2.2) can be used to solve the preference completion problem. We use the widely known technique (see e.g., [2]) and include the explanation and analysis outline for completeness.

kNN-based preference completion algorithm. For any agent i and any $k(n)$-NN solver with parameter $\tau(n)$, let $\mathcal{X}^*_{k\text{NN}}$ be the output of the solver. We first define the the pairwise comparison estimator:

$$\hat{\Pr}[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i] \equiv \frac{\#\text{Agents prefer } y_{\ell_1} \text{ over } y_{\ell_2} \text{ in } \mathcal{X}^*_{k\text{NN}}}{|\mathcal{X}^*_{k\text{NN}}|}. \tag{A.34}$$

Then we define the score function

$$S_i(y_\ell) = \sum_{\ell \in [m]} \mathbf{I}(\hat{\Pr}[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i] > 0.5). \tag{A.35}$$

Then our estimator $\hat{L}_i$ is

$$\hat{L}_i = (\ell_1, \ell_2, \cdots, \ell_m) \quad \text{s.t.} \quad S_i(\ell_1) \geq S_i(\ell_2) \geq \cdots \geq S_i(\ell_m). \tag{A.36}$$

We shall refer to $\hat{L}_i$ as the *standard preference estimator* because most of the preference aggregation algorithm operates in a similar manner (see also Algorithm 12).

Algorithm 12: Standard preference estimator

1 **Input:** $\mathcal{X}^*_{k\text{NN}}$ returned by a kNN solver, agent i
2 **Output:** $\hat{L}_i$
3 Compute $\hat{\Pr}[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i] \equiv \frac{\#\text{Agents prefer } y_{\ell_1} \text{ over } y_{\ell_2} \text{ in } \mathcal{X}^*_{k\text{NN}}}{|\mathcal{X}^*_{k\text{NN}}|}$..
4 Compute $S_i(y_\ell) = \sum_{\ell \in [m]} \mathbf{I}(\hat{\Pr}[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i] > 0.5)$.
5 Find $(\ell_1, \ell_2, \cdots, \ell_m)$ such that $S_i(\ell_1) \geq S_i(\ell_2) \geq \cdots \geq S_i(\ell_m)$
6 Return $\hat{L}_i \leftarrow \{\ell_1, \cdots, \ell_m\}$.

Below is the main result.

Lemma 10. *If $\mathcal{X}^*_{kNN}$ in Algorithm 12 is returned by an k-NN solver with parameter $\tau(n)$, then* $\text{NK}(\hat{L}_i, L_i) = o(1)$ *when* $p^2 k \geq \frac{\log^2 n}{\tau^2(n)}$.

Proof. First, by using Fact 6, we can see that

$$\max_{x_j \in \mathcal{X}^*_{k\mathrm{NN}}} \left|\hat{\Pr}[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i] - \Pr[y_{\ell_1} \succ_j y_{\ell_2} \mid x_i]\right| \le \tau(n).$$

Then by using a standard Chernoff bound over all the agents in $\mathcal{X}^*_{k\mathrm{NN}}$, we have

$$\Pr\left[\left|\hat{\Pr}[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i] - \Pr[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i]\right| \ge \delta + \tau(n)\right] \le 2\exp\left(-\frac{\delta^2}{3} \cdot p^2 k\right).$$

Recall that $\mathcal{L}_i(y_j)$ is y_j's ranking in the ground-truth list L_i. Using the fact that $p^2 k \ge \frac{\log^2 n}{\tau^2(n)}$ and a union bound, we have

$$\Pr\left[\exists \ell_1, \ell_2 : \left|\hat{\Pr}[y_{\ell_1} \succ_i y_{\ell_2}] - \Pr[y_{\ell_1} \succ_i y_{\ell_2}]\right| \ge 2\tau(n)\right] \le n^{-10}$$

The exponent 10 is chosen arbitrarily and is not optimized. Note that $\mathcal{L}(y_{\ell_1}) < \mathcal{L}(y_{\ell_2})$ if and only if $\Pr[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i] > \frac{1}{2}$. Together with the fact that $\mathcal{D}_Y$ are near-uniform, we have

$$|\hat{\mathcal{L}}_i(y_\ell) - \mathcal{L}_i(y_\ell)| = o(n),$$

whic implies $\mathrm{NK}(\hat{\mathcal{L}}_i, \mathcal{L}_i) = o(1)$. □

We may use a similar technique to prove that KT-kNN cannot produce an $\hat{L}_i$ in reasonable quality:

Lemma 11. *Let $\mathcal{X}^*_{kNN}$ be the return of Algorithm* KT-kNN. $\mathrm{NK}(\hat{L}_i, L_i) = \Theta(1)$ *even when $n \to \infty$, $m \to \infty$, and $p = \Theta(1)$.*

A.5 Details of the Experiments

A.5.1 Synthetic Data

The case $d = 1$. We now explain the setup for producing Figure 2.2(a-c). We let $\mathcal{D}_X = \mathcal{D}_Y = \mathrm{Uniform}([-c, c])$, $n = 5000$, and $m = 42586 \approx n \ln n$. This scale is closer to publicly available data. The preferences are generated according to the distance-based random preference model. The *pairwise (prediction) error* is defined as $\left|\hat{\Pr}[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i] - \Pr[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i]\right|$. The *average latent distance* is the average of the latent distances between x_i and x_j for all x_j from the output.

We also examine the NN problem under different settings of c and $\theta(x, y)$ and produce the prediction error for different algorithms, including Anchor-kNN,KT-kNN and Ground-Truth-kNN. See Figure A.2. Our conclusion remains unchanged.

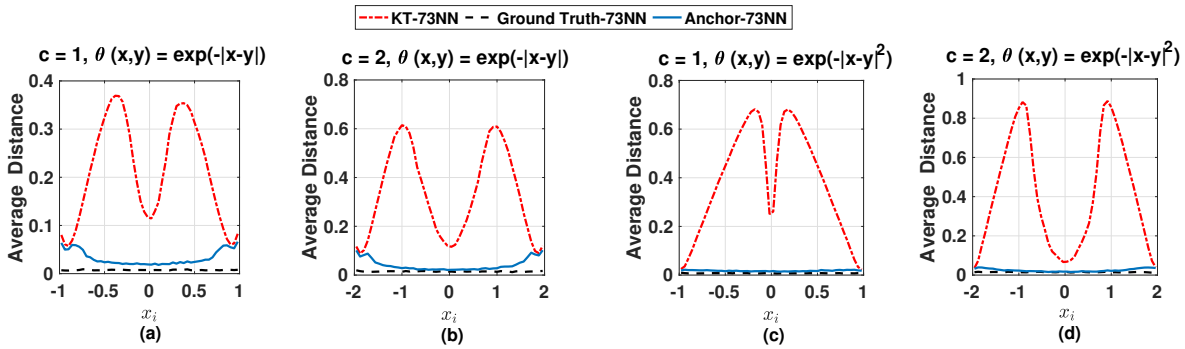

Figure A.2: Validation for different c and $\theta(x, y)$ when $k = 73 \approx \ln^2 n$ (a) $c = 1$ and $\theta(x, y) = e^{-|x-y|}$. (b) $c = 2$ and $\theta(x, y) = e^{-|x-y|}$. (c) $c = 1$ and $\theta(x, y) = e^{-|x-y|^2}$. (d) $c = 2$ and $\theta(x, y) = e^{-|x-y|^2}$.

The case $1 \leq d \leq 10$. In Figure 2.2d and 2.2e, all settings are the same as 1D except that we let $\mathcal{D}_X = \mathcal{D}_Y = \text{Uniform}\left(\left[-2^{2/d-1}, 2^{2/d-1}\right]^d\right)$. In this way, the volume of the latent space remains a constant for different dimensions.

High dimensional space and low m. We also examine the case when m is much smaller because Theorem 2 suggests that Anchor-kNN's performance is "reasonable" even when m is small. We set $n = 5000$ and $m = 618 \approx \ln^3 n$. We note that Anchor-kNN consistently outperforms KT-kNN in terms of both average latent distance and prediction error. See Figure A.3 for the result.

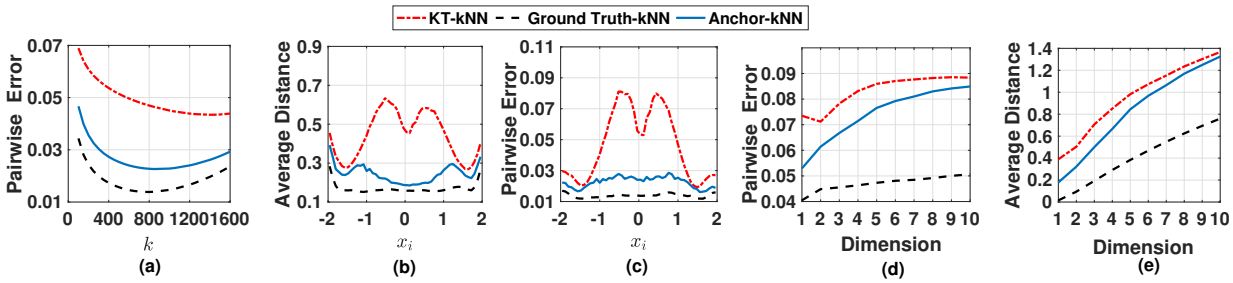

Figure A.3: Simulation on small m ($m \approx \ln^3 n$). (a) Average pairwise probability prediction error ($\left|\hat{\Pr}\left[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i\right] - \Pr\left[y_{\ell_1} \succ_i y_{\ell_2} \mid x_i\right]\right|$) for different $k \in [101, 1601]$. (b) The average latent distance between x_i and its predicted 801 near neighbors. (c) Pairwise prediction error when $k = 801$ (optimal k for ground-truth kNN). (d) Validation for high dimension ($1 \sim 10$ dimensional latent space). Let $k = 73 \approx \ln^2 n$ and measure the prediction errors when $k = 73$. (e) The average latent distance ($k = 73$).

A.5.2 Real Dataset

Dataset. Our experiments closely follow [2]. We used the Netflix dataset [33], [34], which contains more than 100 million ratings from 480 thousand users on around 18 thousand movies. The ratings are integers between 1 and 5.

We apply a standard procedure to translate rating data into ranking data [1]. We take 1000 movies with the most ratings and randomly extract 500 users who rated more than 100 different movies. We use 45% of the triples (user-id, movie, rating) as the training set and 55% of the triples are split into testing set. We drop users if they have fewer than 50 ratings in the training set and fewer than 10 ratings in the testing set.

Our evaluation metrics include Kendall's tau coefficient, Spearman Rho, Precision@5 and NDCG@5. Kendall-Tau and Spearman Rho measure the rank correlation between the predicted ranking and the ground-truth. Precision@5 and NDCG@5 evaluate the effectiveness of the predicted ranking in the top items. For all the metrics, a higher score indicates better performance. In Precision@5, we assume that any rating equals to 5 corresponds to a relevant item. See also [7].

We compare Anchor-kNN with KT-kNN and a standard collaborative filtering algorithm. The collaborative filtering algorithm uses cosine similarities as the distance metrics. All the algorithms use the standard preference estimator (see Algorithm 12). We examine these algorithms' performance for different k. Table 2.1 shows the full results.

Table A.1: Summary of results on Netflix dataset; expanded version of Table 2.1.

KT coefficient	CF	KT-kNN	Anchor-kNN	Spearman	CF	KT-kNN	Anchor-kNN
k =5	0.0531	0.0548	**0.0989**		0.0787	0.0811	**0.1462**
k= 10	0.0883	0.0977	**0.1456**		0.1306	0.1436	**0.2143**
k= 15	0.1199	0.1259	**0.1646**		0.1770	0.1860	**0.2423**
k= 20	0.1365	0.1444	**0.1771**		0.2010	0.2127	**0.2599**
k= 25	0.1403	0.1570	**0.1869**		0.2214	0.2307	**0.2742**
Precision@5	CF	KT-kNN	Anchor-kNN	NDCG@5	CF	KT-kNN	Anchor-kNN
k =5	0.3286	0.3386	**0.4045**		0.7886	0.7848	**0.8186**
k= 10	0.3977	0.4091	**0.4559**		0.8126	0.8178	**0.8389**
k= 15	0.4291	0.4573	**0.4850**		0.8290	0.8382	**0.8526**
k= 20	0.4432	0.4659	**0.5091**		0.8435	0.8442	**0.8616**
k= 25	0.4718	0.4823	**0.5077**		0.8486	0.8500	**0.8639**

APPENDIX B
APPENDIX FOR CHAPTER 4

B.1 Additional Discussions About the Motivating Example

B.1.1 Detailed Settings About Figure 4.1

In the motivating example, the adversary's utility represent $\max_{x':||x-x'||_1} u(x, x', t)$, where $u(x, x', t)$ is the adjusted utility defined above Lemma 13. We set the threshold of accuracy $t = 51\%$ in Figure 4.1. In other words, the adversary omitted the utilities coming from the predictors with no more than 51% accuracy. To compare the privacy of different years, we assume that the US government only publish the number of votes from top-2 candidate. The δ parameter plotted in Figure 4.1 follows the database-wise privacy parameter $\delta_{\epsilon,\mathcal{M}(x)}$ in Definition 6, where $\epsilon = \ln\left(\frac{0.51}{0.49}\right)$. One can see that the threshold $t = \frac{e^\epsilon}{e^\epsilon+1}$, which matches the setting of Lemma 13. In all experiments related to our motivating example, we directly calculated the probabilities and our numerical results does not include any randomness.

B.1.2 The Motivating Example Under Other Settings

Figure B.1 presents different settings for the privacy level of US presidential elections (the motivating example). In all our settings, we set $t = \frac{e^\epsilon}{e^\epsilon+1}$. Figure B.1 shows similar information as Figure 4.1 under different settings (threshold of accuracy $t \in \{0.5, 0.51, 0.6\}$ and ratio of lost votes = 0.01% or 0.2%). In all settings of Figure B.1, the US presidential election is much more private than what DP predicts. Figure B.1(d) shows that the US presidential election is also private ($\delta = o(1/n)$ and $\epsilon \approx 0.4$) when there are only 0.01% of votes got lost.

B.2 Additional Related Works and Discussions

Quantized neural networks [137]–[140] were initially designed to make hardware implementations of DNNs easier. In the recent decade, quantized neural networks becomes a research hotspot again owing to its growing applications on mobile devises [141], [74], [142]. In quantized neural networks, the weights [143]–[147], activation functions [148]–[151] and/or gradients [73], [152]–[154] are quantized. When the gradients are quantized, both the

Portions of this chapter have previously appeared as: A. Liu, Y. Wang, and L. Xia. "Smoothed differential privacy." 2021, *arXiv:2107.01559*.

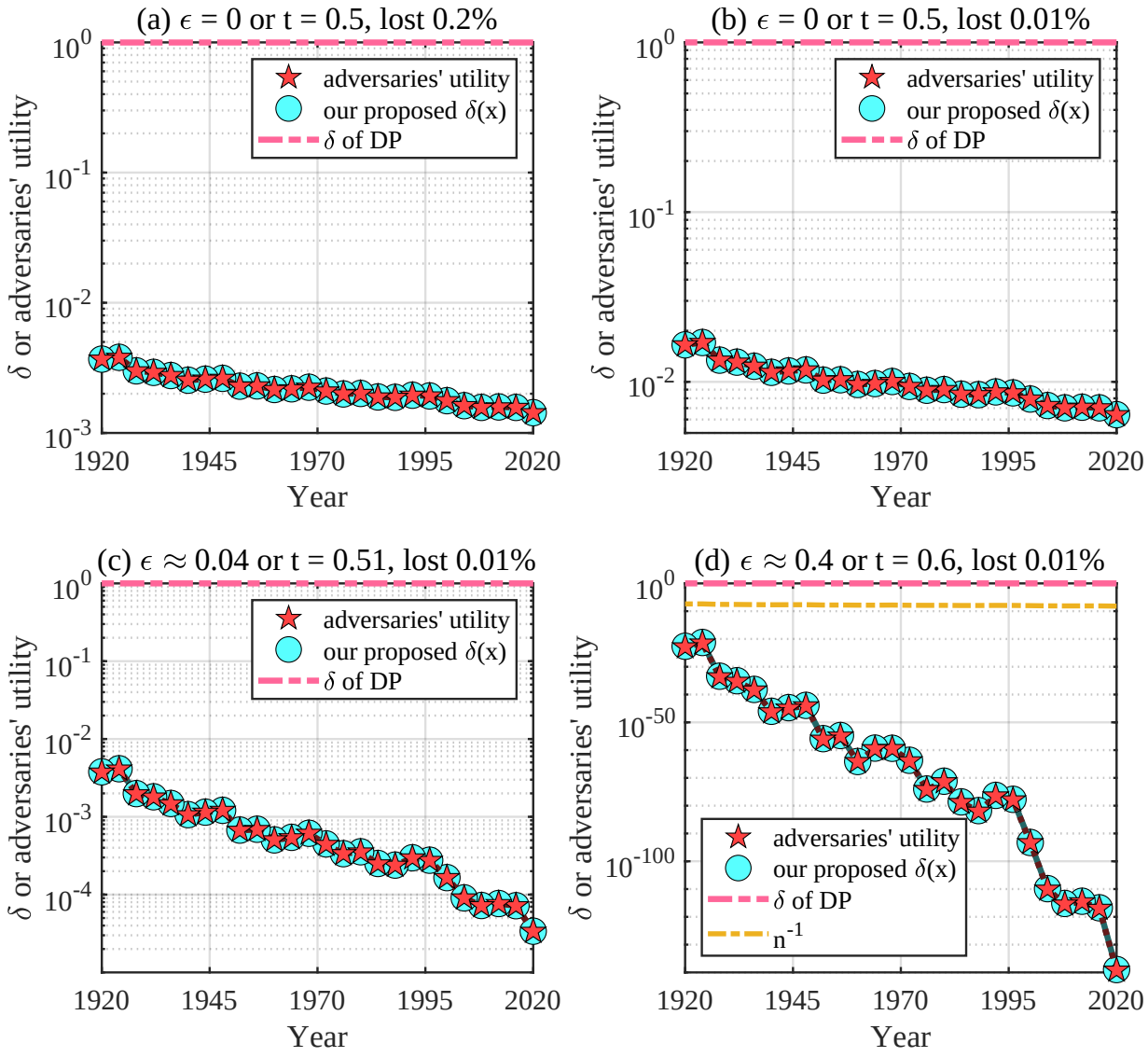

Figure B.1: The privacy level of US presidential elections under different settings.

training and inference of DNN are accelerated [73], [92]. Gradient quantization can also save communication costs when the DNNs are trained on distributed systems [142].

B.3 Detailed Explanations About DP and Smoothed DP

In this section, we present the formal definition of "probabilities" used in Justification 2 of DP and smoothed DP. To simplify notations, we define the δ-approximate max divergence between two random variables Y and Z as,

$$D_{\infty}^{\delta}(Y||Z) = \max_{\mathcal{S}\in \mathrm{Supp}(Y):\Pr[Y\in\mathcal{S}]\geq\delta} \left[\ln\left(\frac{\Pr[Y\in\mathcal{S}]-\delta}{\Pr[Z\in\mathcal{S}]}\right)\right], \tag{B.1}$$

where $\mathrm{Supp}(Y)$ represents the support of random variable Y. Especially, we use $D_\infty(Y||Z)$ to represent the $D^0_\infty(Y||Z)$, which is the max divergence. One can see that a mechanism $\mathcal{M}$ is (ϵ, δ)-DP if and only if for any pair of neighboring databases x, x': $D^\delta_\infty\big(\mathcal{M}(x)||\mathcal{M}(x')\big) \leq \epsilon$. The next lemma shows the connection between $D_\infty(Y||Z)$ and $D^\delta_\infty(Y||Z)$.

Lemma 12 (Lemma 3.17 in [58]). *$D^\delta_\infty(Y||Z) \leq \epsilon$ if and only if there exists a random variable Y' such that $\Delta(Y, Y') \leq \delta$ and $D_\infty(Y'||Z) \leq \epsilon$, where $\Delta(Y, Z) = \max_{\mathcal{S}} \big|\Pr[Y \in \mathcal{S}] - \Pr[Z \in \mathcal{S}]\big|$.*

In [58], "$\frac{\Pr[\mathcal{M}(x)\in\mathcal{S}]}{\Pr[\mathcal{M}(x')\in\mathcal{S}]} \geq e^\epsilon$ happens with no more than δ probability" means that there exists a random variable Y such that $D_\infty\big(\mathcal{M}(x)||Y\big) \leq \epsilon$ and $\Delta\big(Y, \mathcal{M}(x')\big) \leq \delta$. In other words, this means: with modifying the distribution of $\mathcal{M}(x')$ by at most δ (for its mass), the probability ratio between $\mathcal{M}(x)$ and the modified distribution is always upper bounded by e^ϵ. By now, we explained the meaning of "probability" in [58].

B.4 An Additional Justification of Smoothed DP

Justification 3: $d_{\epsilon,\mathcal{M}}(x, x')$ tightly bounds Bayesian adversary's utilities. We consider the same adversary as in Justification 1. Since the adversary has no information about the missing entry, he/she may assume a uniform prior distribution about the missing entry. Let $X_i \in \mathcal{X}$ denote the missing entry. Observing output a from mechanism $\mathcal{M}$, the adversary's posterior distribution is

$$\begin{aligned}\Pr\left[X_i|a, x_{-i}\right] &= \frac{\Pr\left[a|X_i, x_{-i}\right] \cdot \Pr[X_i|x_{-i}]}{\Pr[a|x_{-i}]} \\ &= \frac{\Pr[\mathcal{M}(x_{-i} \cup \{X_i\}) = a] \cdot \Pr[X_i]}{\sum_{X'} \big(\Pr[\mathcal{M}(x_{-i} \cup \{X'\}) = a] \cdot \Pr[X']\big)} \\ &= \frac{\Pr[\mathcal{M}(x_{-i} \cup \{X_i\}) = a]}{\sum_{X'} \Pr[\mathcal{M}(x_{-i} \cup \{X'\}) = a]}.\end{aligned} \tag{B.2}$$

A Bayesian predictor predicts the missing entry X_i through maximizing the posterior probability. Then, for the uniform prior, when the output is a, the 0/1 loss of the Bayesian predictor is defined as follows, where a correct prediction has zero loss and any incorrect prediction has loss one.

$$\begin{aligned}\ell_{0\text{-}1}(a, x_{-i}) &= 0^2 \cdot \max_i \left(\Pr\left[X_i|a, x_{-i}\right]\right) + 1^2 \cdot \Big(1 - \max_i \left(\Pr\left[X_i|a, x_{-i}\right]\right)\Big) \\ &= 1 - \max_i \left(\Pr\left[X_i|a, x_{-i}\right]\right)\end{aligned} \tag{B.3}$$

Then, we define the adjusted utility of adversary (in Bayesian prediction), which is the expectation of a normalized version of $\ell_{0\text{-}1}$. Mathematically, for a database x, we define the adjusted utility with threshold t as follows,

$$u(t, x_{-i}) = \frac{1}{1-t} \cdot \max_{X_i} \left(\mathbb{E}_{a \sim \mathcal{M}(x_{-i} \cup X_i)} \Big[\max \left\{ 0, 1 - t - \ell_{0\text{-}1}(a) \right\} \Big] \right). \tag{B.4}$$

In short, $u(t, x_{-i})$ is the worst case expectation of $1 - \ell_{0\text{-}1}$ while the contribution from predictors with loss larger than $1 - t$ is omitted. Especially, when the threshold $t \geq 1/|\mathcal{X}|$, an always correct predictor ($\ell_{0\text{-}1} = 0$) has utility 1 and a random guess predictor ($\ell_{0\text{-}1} = 1 - 1/|\mathcal{X}|$) has utility 0. For example, we let $\mathcal{X} = \{0, 1\}$ and consider the coin-flipping mechanism $\mathcal{M}_{\text{CF}}$ with support $\mathcal{X}$, which output X_i with probability p and output $1 - X_i$ with probability $1 - p$. When $p = 1$, the entry X_i is non-private because the adversary can directly learn it from the output of $\mathcal{M}_{\text{CF}}$. Correspondingly, the adjusted utility of adversary is 1 for any threshold $t \in (0, 1)$. When $p = 0.5$, the mechanism give a output uniformly at random from $\mathcal{X}$. In this case, the output of $\mathcal{M}$ cannot provide any information to the adversary. Correspondingly, the adjusted utility of adversary is 0 for any threshold $t \in (0.5, 1)$. In the next lemma, we reveal a connection between the adversary's utility u and $d_{\epsilon,\mathcal{M}}(x, x') + d_{\epsilon,\mathcal{M}}(x', x)$.

Lemma 13. *Given mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{A}$ and any pair of neighboring databases $x, x' \in \mathcal{X}^n$,*

$$u\Big(\frac{e^\epsilon}{e^\epsilon + 1}, x \cap x'\Big) < d_{\epsilon,\mathcal{M}}(x, x') + d_{\epsilon,\mathcal{M}}(x', x).$$

Lemma 13 shows that the adjusted utility is upper bounded by $d_{\epsilon,\mathcal{M}}$. Especially, when $|\mathcal{X}| = 2$, we provide both upper and lower bounds to the adjusted utility in Lemma 14 in Appendix B.5.1, which means that $d_{\epsilon,\mathcal{M}}(x, x')$ is a good measure for the privacy level of $\mathcal{M}$ when $|\mathcal{X}| = 2$. In the following corollary, we show that $\delta_{\epsilon,\mathcal{M}}(x)$ upper bounds the adjusted utility of adversary.

Corollary 3. *Given mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{A}$ and any pair of neighboring databases $x, x' \in \mathcal{X}^n$,*

$$u\Big(\frac{e^\epsilon}{e^\epsilon + 1}, x \cap x'\Big) < 2 \cdot \delta_{\epsilon,\mathcal{M}}(x). \tag{B.5}$$

δ in smoothed DP bounds the adversary's utility in (Bayesian) predictions under realistic settings. Following similar reasoning as Justification 1, we know that the utility under realistic settings (or the smoothed utility) of the adversary cannot be larger than 2δ. Mathematically, a (ϵ, δ, Π)-smoothed DP mechanism $\mathcal{M}$ can guarantee

$$\max_{\vec{\pi}\in\Pi^n}\left(\mathbb{E}_{x\sim\vec{\pi}}\left[u\Big(x, x', \frac{e^\epsilon}{e^\epsilon+1}\Big)\right]\right) < 2\cdot\delta. \tag{B.6}$$

B.5 Missing Proofs for Section 4.3: Smoothed Differential Privacy

To simplify notations, we let $d_{\mathcal{M},\mathcal{S},\epsilon}(x, x') = \big(\Pr[\mathcal{M}(x)\in\mathcal{S}] - e^\epsilon\cdot\Pr[\mathcal{M}(x')\in\mathcal{S}]\big)$. In the next section, we tightly bound $d_{\epsilon,\mathcal{M}}$.

B.5.1 A Tight Bound

Lemma 14. *Given $\mathcal{X}$ such that $|\mathcal{X}| = 2$ and mechanism $\mathcal{M}: \mathcal{X}^n \to \mathcal{A}$ and any pair of neighboring databases $x, x' \in \mathcal{X}^n$,*

$$u\Big(\frac{e^\epsilon}{e^\epsilon+1}, x\cap x'\Big) < d_{\epsilon,\mathcal{M}}(x, x') + d_{\epsilon,\mathcal{M}}(x', x) \le 3\cdot u\Big(\frac{e^\epsilon}{e^\epsilon+1}, x\cap x'\Big).$$

Proof. To simplify notations, for any output a, we let

$$p(a) = \Pr[\mathcal{M}(x) = a] \quad \text{and} \quad p'(a) = \Pr[\mathcal{M}(x') = a].$$

Then, we define the utility of adversary when the database is x

$$u(x, x', t) = \frac{1}{1-t}\cdot\mathbb{E}_{a\sim\mathcal{M}(x)}\Big[\max\big\{0, 1-t-\ell_{0\text{-}1}(a, x\cap x')\big\}\Big].$$

Let X_i be the different entry between x and x', we have,

$$u(t, x\cap x') = \max_{X_i}\big[u(x, x', t)\big].$$

To further simplify notations we define the adjusted utility with the threshold for output a as follows.

$$u\Big(a, x, x', \frac{e^\epsilon-1}{e^\epsilon+1}\Big) = \max\left\{0,\ \frac{|p(a)-p'(a)|}{p(a)+p'(a)} - \frac{e^\epsilon-1}{e^\epsilon+1}\right\}.$$

Note that the threshold is for the utility (not for the accuracy). Then, it's easy to find that

$$u\Big(x, x', \frac{e^\epsilon}{e^\epsilon+1}\Big) = \mathbb{E}_{a\sim\mathcal{M}(x)}\left[u\Big(a, x, x', \frac{e^\epsilon-1}{e^\epsilon+1}\Big)\right].$$

Using the above notations, we have,

$$\begin{aligned}
&\mathbb{E}_{a\sim\mathcal{M}(x)}\left[u\Big(a, x, x', \frac{e^\epsilon-1}{e^\epsilon+1}\Big)\right] \\
=& \sum_{a:p(a)>e^\epsilon\cdot p'(a)} p(a)\cdot\left(\frac{p(a)-p'(a)}{p(a)+p'(a)} - \frac{e^\epsilon-1}{e^\epsilon+1}\right) + \sum_{a:p'(a)>e^\epsilon\cdot p(a)} p(a)\cdot\left(\frac{p'(a)-p'(a)}{p(a)+p'(a)} - \frac{e^\epsilon-1}{e^\epsilon+1}\right).
\end{aligned}$$

Then, we let $r = \frac{p(a)}{p'(a)}$ and analyze the first term,

$$\begin{aligned}
&\sum_{a:p(a)>e^\epsilon\cdot p'(a)} p(a)\cdot\left(\frac{p(a)-p'(a)}{p(a)+p'(a)} - \frac{e^\epsilon-1}{e^\epsilon+1}\right) \\
=& \sum_{a:p(a)>e^\epsilon\cdot p'(a)} p(a)\cdot\left(-\frac{2\cdot p'(a)}{p(a)+p'(a)} + \frac{2}{e^\epsilon+1}\right) \\
=& \sum_{a:p(a)>e^\epsilon\cdot p'(a)} p(a)\cdot\frac{2\cdot(r-e^\epsilon)}{(e^\epsilon+1)\cdot(r+1)}
\end{aligned}$$

Then, we analyze $\max_{\mathcal{S}} d_{\mathcal{M},\mathcal{S},\epsilon}(x, x')$.

$$\max_{\mathcal{S}} d_{\mathcal{M},\mathcal{S},\epsilon}(x, x') = \sum_{a:p(a)>e^\epsilon\cdot p'(a)} p(a) - e^\epsilon\cdot p'(a) = \sum_{a:p(a)>e^\epsilon\cdot p'(a)} p(a)\cdot\frac{r-e^\epsilon}{r}.$$

For the upper bound of utility, we have,

$$\begin{aligned}
\frac{2}{e^\epsilon+1}\cdot\max_{\mathcal{S}} d_{\mathcal{M},\mathcal{S},\epsilon}(x, x') &= \sum_{a:p(a)<e^\epsilon\cdot p'(a)} p(a)\cdot\frac{2\cdot(r-e^\epsilon)}{(e^\epsilon+1)\cdot r} \\
&< \text{the first term of } \mathbb{E}_{a\sim\mathcal{M}(x)}\left[u\Big(a, x, x', \frac{e^\epsilon-1}{e^\epsilon+1}\Big)\right].
\end{aligned}$$

By symmetry, we have,

$$\frac{2}{e^\epsilon+1}\cdot\max_{\mathcal{S}} d_{\mathcal{M},\mathcal{S},\epsilon}(x', x) < \text{the second term of } \mathbb{E}_{a\sim\mathcal{M}(x)}\left[u\Big(a, x, x', \frac{e^\epsilon-1}{e^\epsilon+1}\Big)\right].$$

The upper bound part of lemma follows by combining the above two bounds. For the lower

bound, we have,

$$\frac{2}{e^\epsilon+3}\cdot\max_{\mathcal{S}} d_{\mathcal{M},\mathcal{S},\epsilon}(x,x') = \sum_{a:p(a)>e^\epsilon\cdot p'(a)} p(a)\cdot\frac{2\cdot(r-e^\epsilon)}{(e^\epsilon+3)\cdot r}$$
$$\geq \text{the first term of } \mathbb{E}_{a\sim\mathcal{M}(x)}\left[u\Big(a,x,x',\frac{e^\epsilon-1}{e^\epsilon+1}\Big)\right].$$

By symmetry, we have,

$$\frac{2}{e^\epsilon+3}\cdot\max_{\mathcal{S}} d_{\mathcal{M},\mathcal{S},\epsilon}(x',x) \geq \text{the second term of } \mathbb{E}_{a\sim\mathcal{M}(x)}\left[u\Big(a,x,x',\frac{e^\epsilon-1}{e^\epsilon+1}\Big)\right].$$

The lower bound part of lemma follows by combining the above two bounds. □

B.5.2 The Proof for Lemma 13

Lemma 13 *Given mechanism $\mathcal{M}:\mathcal{X}^n\to\mathcal{A}$ and any pair of neighboring databases $x,x'\in\mathcal{X}^n$,*

$$u\Big(\frac{e^\epsilon}{e^\epsilon+1}, x\cap x'\Big) < d_{\epsilon,\mathcal{M}}(x,x') + d_{\epsilon,\mathcal{M}}(x',x).$$

Proof. We assume that $\mathcal{X}=\{X_1,\cdots,X_m\}$. Then, using the notations in the proof of Lemma 14, we know that

$$\begin{aligned}
u(t,x_{-i}) &= \frac{1}{1-t}\cdot\max_{X_i}\Big(\mathbb{E}_{a\sim\mathcal{M}(x_{-i}\cup X_i)}\Big[\max\big\{0,1-t-\ell_{0\text{-}1}(a)\big\}\Big]\Big)\\
&= \frac{1}{1-t}\cdot\max_{X_i}\left(\mathbb{E}_{a\sim\mathcal{M}(x_{-i}\cup X_i)}\Big[\max\Big\{0,\ \frac{\max_i\Pr[\mathcal{M}(x_{-i}\cup\{X_i\})=a]}{\sum_{X'\in\mathcal{X}}\Pr[\mathcal{M}(x_{-i}\cup\{X'\})=a]}-t\Big\}\Big]\right)\\
&\leq \frac{1}{1-t}\cdot\max_{X_i}\left(\mathbb{E}_{a\sim\mathcal{M}(x_{-i}\cup X_i)}\Big[\max\Big\{0,\ \frac{\max_i\Pr[\mathcal{M}(x_{-i}\cup\{X_i\})=a]}{\sum_{X'\in\mathcal{X}^*}\Pr[\mathcal{M}(x_{-i}\cup\{X'\})=a]}-t\Big\}\Big]\right)
\end{aligned}$$

where $\mathcal{X}^*\subseteq\mathcal{X}$, $|\mathcal{X}^*|=2$ and $\arg\max_{X'\in\mathcal{X}}\Pr[\mathcal{M}(x_{-i}\cup\{X'\})=a]\in\mathcal{X}^*$. In words, $\mathcal{X}^*$ is a set of two missing data while the MLE prediction is included in the database. Now, we reduced the $u(t,x_{-i})$ of general $\mathcal{X}$ to a set of two possible missing data. Then, Lemma 13 follows by the direct application of Lemma 14. □

B.5.3 The Proof for Lemma 7

Lemma 7 (DP in $\delta_{\epsilon,\mathcal{M}}(x)$) *Mechanism $\mathcal{M}: \mathcal{X}^n \to \mathcal{A}$ is (ϵ, δ)-differentially private if and only if,*

$$\max_{x \in \mathcal{X}^n} \Big(\delta_{\epsilon,\mathcal{M}}(x) \Big) \leq \delta.$$

Proof. The "if direction": By the definition of $\delta_{\epsilon,\mathcal{M}}(x)$, we know that,

$$\delta_{\epsilon,\mathcal{M}}(x) \geq \max_{\mathcal{S}, x': ||x - x'||_1 \leq 1} \big(d_{\mathcal{M},\mathcal{S},\epsilon}(x, x') \big).$$

Thus, for all $x \in \mathcal{X}^n$, we have $\delta \geq \delta_{\epsilon,\mathcal{M}}(x) \geq \max_{\mathcal{S}, x': ||x-x'||_1 \leq 1} \big(d_{\mathcal{M},\mathcal{S},\epsilon}(x, x') \big)$, which is equivalent with

$$\begin{aligned}
&\text{For all } x, \max_{\mathcal{S}, x: ||x - x'||_1 \leq 1} \big(\Pr[\mathcal{M}(x) \in \mathcal{S}] - e^{\epsilon} \cdot \Pr[\mathcal{M}(x') \in \mathcal{S}] \big) \leq \delta \\
\Leftrightarrow\ &\text{For all } \mathcal{S}, x, x' \text{ such that } ||x - x'||_1 \leq 1, \Pr[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(x') \in \mathcal{S}] + \delta.
\end{aligned}$$

One can see that the above statement is the same as the requirement of DP in Definition 5.

The "only if direction": In the definition of DP, we note database x and x' are interchangeable. Thus,

$$\begin{aligned}
&\text{For all } \mathcal{S}, x, x' \text{ such that } ||x - x'||_1 \leq 1, \Pr[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(x') \in \mathcal{S}] + \delta \\
\Leftrightarrow\ &\text{For all } x, \max_{\mathcal{S}, x: ||x - x'||_1 \leq 1} \big(\Pr[\mathcal{M}(x') \in \mathcal{S}] - e^{\epsilon} \cdot \Pr[\mathcal{M}(x) \in \mathcal{S}] \big) \leq \delta \\
\Leftrightarrow\ &\text{For all } x, \max_{\mathcal{S}, x: ||x - x'||_1 \leq 1} \big(d_{\mathcal{M},\mathcal{S},\epsilon}(x', x) \big) \leq \delta
\end{aligned}$$

Then, we combine the bounds for $d_{\mathcal{M},\mathcal{S},\epsilon}(x', x)$ and $d_{\mathcal{M},\mathcal{S},\epsilon}(x, x')$. Then, for all $x \in \mathcal{X}^n$,

$$\begin{aligned}
\delta_{\epsilon,\mathcal{M}}(x) &= \max \Big(0, \max_{\mathcal{S}, x': ||x - x'||_1 \leq 1} \big(d_{\mathcal{M},\mathcal{S},\epsilon}(x, x') \big), \max_{\mathcal{S}, x': ||x - x'||_1 \leq 1} \big(d_{\mathcal{M},\mathcal{S},\epsilon}(x', x) \big) \Big) \\
&\leq \max \Big(0, \delta, \delta \Big) = \delta.
\end{aligned}$$

By now, we proved both directions of Theorem 7. □

B.6 Missing Proofs for Section 4.4: The Properties of Smoothed DP

To simplify notations, we let $d_{\mathcal{M},\mathcal{S},\epsilon}(x,x') = \big(\Pr\left[\mathcal{M}(x)\in\mathcal{S}\right] - e^{\epsilon}\cdot\Pr\left[\mathcal{M}(x')\in\mathcal{S}\right]\big)$ in all proofs of this section.

B.6.1 The Proof for Proposition 2

Proposition 2 (post-processing). *Let $\mathcal{M}:\mathcal{X}^n\to\mathcal{A}$ be a (ϵ,δ,Π)-smoothed DP mechanism. Given $f:\mathcal{A}\to\mathcal{A}'$ be any arbitrary function (either deterministic or randomized), we know $f\circ\mathcal{M}:\mathcal{X}^n\to\mathcal{A}'$ is also (ϵ,δ,Π)-smoothed DP.*

Proof. For any fixed database x, any database x' such that $||x-x'||_1\le 1$ and any set of output $\mathcal{S}\subseteq\mathcal{A}'$, we let $\mathcal{T}=\{y\in\mathcal{A}:\ f(y)\in\mathcal{A}'\}$. Then, we have,

$$\begin{aligned}\Pr\left[f\big(\mathcal{M}(x)\big)\in\mathcal{S}\right] &= \Pr\left[\mathcal{M}(x)\in\mathcal{T}\right]\\ &\le e^{\epsilon}\cdot\Pr\left[\mathcal{M}(x')\in\mathcal{T}\right]+\delta_{\epsilon,\mathcal{M}}(x)\\ &= e^{\epsilon}\cdot\Pr\left[f\big(\mathcal{M}(x')\big)\in\mathcal{S}\right]+\delta_{\epsilon,\mathcal{M}}(x).\end{aligned}$$

Considering that the above inequality holds for any pair of neighboring x,x' and any $\mathcal{S}$, we have,

$$\forall x,\ \ \delta_{\epsilon,\mathcal{M}}(x)\ge\max_{\mathcal{S},x':||x-x'||_1\le 1}\big(d_{f\circ\mathcal{M},\mathcal{S},\epsilon}(x,x')\big),$$

where, $d_{f\circ\mathcal{M},\mathcal{S},\epsilon}(x,x') = \big(\Pr\left[f\big(\mathcal{M}(x)\big)\in\mathcal{S}\right] - e^{\epsilon}\cdot\Pr\left[f\big(\mathcal{M}(x')\big)\in\mathcal{S}\right]\big)$. Using the same procedure, we have,

$$\forall x,\ \ \delta_{\epsilon,\mathcal{M}}(x)\ge\max_{\mathcal{S},x':||x-x'||_1\le 1}\big(d_{f\circ\mathcal{M},\mathcal{S},\epsilon}(x',x)\big),$$

Combining the above two inequalities, we have,

$$\forall x,\ \delta_{\epsilon,\mathcal{M}}(x)\ge\delta_{\epsilon,f\circ\mathcal{M}}(x).$$

Then, for any $\vec{\pi}$,

$$\mathbb{E}_{x\sim\vec{\pi}}\left[\delta_{\epsilon,\mathcal{M}}(x)\right]\ge\mathbb{E}_{x\sim\vec{\pi}}\left[\delta_{\epsilon,f\circ\mathcal{M}}(x)\right].$$

Thus,

$$\max_{\vec{\pi}} \left(\mathbb{E}_{x\sim\vec{\pi}}\left[\delta_{\epsilon,\mathcal{M}}(x)\right]\right) \geq \max_{\vec{\pi}} \left(\mathbb{E}_{x\sim\vec{\pi}}\left[\delta_{\epsilon,f\circ\mathcal{M}}(x)\right]\right).$$

Then, Proposition 2 follows by the definition of smoothed DP. □

B.6.2 The Proof for Proposition 3

Proposition 3 (Composition). *Let $\mathcal{M}_i : \mathcal{X}^n \to \mathcal{A}_i$ be an $(\epsilon_i, \delta_i, \Pi)$-smoothed DP mechanism for any $i \in [k]$. Define $\mathcal{M}_{[k]} : \mathcal{X}^n \to \prod_{i=1}^k \mathcal{A}_i$ as $\mathcal{M}_{[k]}(x) = \big(\mathcal{M}_1(x), \cdots, \mathcal{M}_k(x)\big)$. Then, $\mathcal{M}_{[k]}$ is $\Big(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i, \Pi\Big)$-smoothed DP.*

Proof. We first prove the $k = 2$ case of Proposition 3.

By the definition of $d_{\mathcal{M},\mathcal{S},\epsilon}(x, x')$, for any $\mathcal{S}_1 \in \mathcal{A}_1$ and any x' such that $||x - x'||_1 \leq 1$,

$$\begin{aligned} & d_{\mathcal{M}_1,\mathcal{S}_1,\epsilon_1}(x, x') = \Pr\left[\mathcal{M}_1(x) \in \mathcal{S}_1\right] - e^{\epsilon_1} \cdot \Pr\left[\mathcal{M}(x') \in \mathcal{S}_1\right] \\ \Leftrightarrow \quad & \frac{\Pr\left[\mathcal{M}_1(x) \in \mathcal{S}_1\right] - d_{\mathcal{M}_1,\mathcal{S}_1,\epsilon_1}(x, x')}{\Pr\left[\mathcal{M}_1(x') \in \mathcal{S}_1\right]} = e^{\epsilon_1}. \end{aligned}$$

Similarly, for for any $\mathcal{S}_2 \in \mathcal{A}_2$ and any x' such that $||x - x'||_1 \leq 1$, we have,

$$\frac{\Pr\left[\mathcal{M}_2(x) \in \mathcal{S}_2\right] - d_{\mathcal{M}_2,\mathcal{S}_2,\epsilon_2}(x, x')}{\Pr\left[\mathcal{M}_2(x') \in \mathcal{S}_2\right]} = e^{\epsilon_2}.$$

Combining the above two equations, we have,

$$\begin{aligned} e^{\epsilon_1+\epsilon_2} &= \frac{\Pr\left[\mathcal{M}_1(x) \in \mathcal{S}_1\right] - d_{\mathcal{M}_1,\mathcal{S}_1,\epsilon_1}(x, x')}{\Pr\left[\mathcal{M}_1(x') \in \mathcal{S}_1\right]} \cdot \frac{\Pr\left[\mathcal{M}_2(x) \in \mathcal{S}_2\right] - d_{\mathcal{M}_2,\mathcal{S}_2,\epsilon_2}(x, x')}{\Pr\left[\mathcal{M}_2(x') \in \mathcal{S}_2\right]} \\ &\leq \frac{\Pr\left[\mathcal{M}_1(x) \in \mathcal{S}_1\right] \cdot \Pr\left[\mathcal{M}_2(x) \in \mathcal{S}_2\right] - \big(d_{\mathcal{M}_1,\mathcal{S}_1,\epsilon_1}(x, x') + d_{\mathcal{M}_2,\mathcal{S}_2,\epsilon_2}(x, x')\big)}{\Pr\left[\mathcal{M}_1(x') \in \mathcal{S}_1\right] \cdot \Pr\left[\mathcal{M}_2(x') \in \mathcal{S}_2\right]}, \end{aligned}$$

which is equivalent with,

$$\begin{aligned} & d_{\mathcal{M}_1,\mathcal{S}_1,\epsilon_1}(x, x') + d_{\mathcal{M}_2,\mathcal{S}_2,\epsilon_2}(x, x') \\ \leq\ & \Pr\left[\mathcal{M}_1(x) \in \mathcal{S}_1\right] \cdot \Pr\left[\mathcal{M}_2(x) \in \mathcal{S}_2\right] \\ & \quad - e^{\epsilon_1+\epsilon_2} \cdot \big(\Pr\left[\mathcal{M}_1(x') \in \mathcal{S}_1\right] \cdot \Pr\left[\mathcal{M}_2(x') \in \mathcal{S}_2\right]\big). \end{aligned} \tag{B.7}$$

Note that in the $k = 2$ case, $\mathcal{M}_{[k]}(x) = \big(\mathcal{M}_1(x), \mathcal{M}_2(x)\big)$. Thus, for any $x \in \mathcal{X}$, we have $\Pr\left[\mathcal{M}_{[k]}(x) \in \mathcal{S}_1 \times \mathcal{S}_2\right] = \Pr\left[\mathcal{M}_1(x) \in \mathcal{S}_1\right] \cdot \Pr\left[\mathcal{M}_2(x) \in \mathcal{S}_2\right]$. Combining with (B.7), for any

$\mathcal{S} \in \mathcal{A}_1 \times \mathcal{A}_2$ and any x, x' such that $||x - x'||_1 \leq 1$,

$$\begin{aligned} d_{\mathcal{M}_{[k]},\mathcal{S},\epsilon_1+\epsilon_2}(x, x') &= \Pr\left[\mathcal{M}_{[k]}(x) \in \mathcal{S}\right] - e^{\epsilon_1+\epsilon_2} \cdot \Pr\left[\mathcal{M}_{[k]}(x') \in \mathcal{S}\right] \\ &\geq d_{\mathcal{M}_1,\mathcal{S}_1,\epsilon_1}(x, x') + d_{\mathcal{M}_2,\mathcal{S}_2,\epsilon_2}(x, x'). \end{aligned}$$

By symmetry, for any $\mathcal{S} \in \mathcal{A}_1 \times \mathcal{A}_2$ and any x, x' such that $||x - x'||_1 \leq 1$, we have,

$$d_{\mathcal{M}_{[k]},\mathcal{S},\epsilon_1+\epsilon_2}(x', x) \geq d_{\mathcal{M}_1,\mathcal{S}_1,\epsilon_1}(x', x) + d_{\mathcal{M}_2,\mathcal{S}_2,\epsilon_2}(x', x).$$

Then, we the definition of $\delta_{\epsilon,\mathcal{M}}(x)$,

$$\begin{aligned} &\delta_{\epsilon_1+\epsilon_2,\mathcal{M}_{[k]}}(x) \\ =\ &\max\Big(0, \max_{\mathcal{S},x':||x-x'||_1\leq 1}\big(d_{\mathcal{M}_{[k]},\mathcal{S},\epsilon_1+\epsilon_2}(x', x)\big), \max_{\mathcal{S},x':||x-x'||_1\leq 1}\big(d_{\mathcal{M}_{[k]},\mathcal{S},\epsilon_1+\epsilon_2}(x, x')\big)\Big) \\ \geq\ &\delta_{\epsilon_1,\mathcal{M}_1}(x) + \delta_{\epsilon_2,\mathcal{M}_2}(x). \end{aligned}$$

By the definition of the smoothed DP, we proved the $k = 2$ case of Proposition 3. We prove the $k > 2$ cases by induction. Here, $\mathcal{M}_{[k]}$ is treated as $\big(\mathcal{M}_{[k-1]}, \mathcal{M}_k\big)$. Then, $\mathcal{M}_{[k]}$ will reduce to $\mathcal{M}_{[k-1]}$ by applying the conclusion in $k = 2$ case. □

B.6.3 The Proof for Proposition 4

Proposition 4 (Pre-processing for deterministic functions). *Let $f : \mathcal{X}^n \to \tilde{\mathcal{X}}^n$ be a deterministic function and $\mathcal{M} : \tilde{\mathcal{X}}^n \to \mathcal{A}$ be a randomized mechanism. Then, $\mathcal{M} \circ f : \mathcal{X}^n \to \mathcal{A}$ is (ϵ, δ, Π)-smoothed DP if $\mathcal{M}$ is $\big(\epsilon, \delta, f(\Pi)\big)$-smoothed DP.*

Proof. According to the definition of smoothed DP, for any fixed database $x \in \mathcal{X}^n$, we have that

$$\delta_{\epsilon,\mathcal{M}\circ f}(x) \triangleq \max\Big(0, \max_{\mathcal{S},x':||x-x'||_1\leq 1}\big(d_{\mathcal{M}\circ f,\mathcal{S},\epsilon}(x, x')\big), \max_{\mathcal{S},x':||x-x'||_1\leq 1}\big(d_{\mathcal{M}\circ f,\mathcal{S},\epsilon}(x', x)\big)\Big),$$

where $d_{\mathcal{M}\circ f,\mathcal{S},\epsilon}(x, x') = \big(\Pr\left[\mathcal{M}\big(f(x)\big) \in \mathcal{S}\right] - e^{\epsilon} \cdot \Pr\left[\mathcal{M}\big(f(x')\big) \in \mathcal{S}\right]\big)$.

Similarly, for mechanism $\mathcal{M}$ and any fixed database $a \in \tilde{\mathcal{X}}^n$, we have,

$$\delta_{\epsilon,\mathcal{M}}(a) \triangleq \max\Big(0, \max_{\mathcal{S},a':||a-a'||_1\leq 1}\big(d_{\mathcal{M},\mathcal{S},\epsilon}(a, a')\big), \max_{\mathcal{S},a':||a-a'||_1\leq 1}\big(d_{\mathcal{M},\mathcal{S},\epsilon}(a', a)\big)\Big),$$

where $d_{\mathcal{M},\mathcal{S},\epsilon}(a,a') = \big(\Pr[\mathcal{M}(a)\in\mathcal{S}] - e^{\epsilon}\cdot\Pr[\mathcal{M}(a')\in\mathcal{S}]\big)$. For any fixed database x, we let $a = f(x)$ and have,

$$\begin{aligned}
&\max_{\mathcal{S},x':||x-x'||_1\le 1}\big(d_{\mathcal{M}\circ f,\mathcal{S},\epsilon}(x,x')\big)\\
=&\max_{\mathcal{S},x':||x-x'||_1\le 1}\Big(\Pr\big[\mathcal{M}\big(f(x)\big)\in\mathcal{S}\big] - e^{\epsilon}\cdot\Pr\big[\mathcal{M}\big(f(x')\big)\in\mathcal{S}\big]\Big)\\
=&\max_{\mathcal{S},x':||x-x'||_1\le 1}\Big(\Pr[\mathcal{M}(a)\in\mathcal{S}] - e^{\epsilon}\cdot\Pr\big[\mathcal{M}\big(f(x')\big)\in\mathcal{S}\big]\Big)
\end{aligned}$$

Note that f is an deterministic function. Thus, for any database x and x' such that different on no more than one entry, we know that $f(x)$ and $f(x')$ will not have more than one different entries. Then, we have,

$$\begin{aligned}
\max_{\mathcal{S},x':||x-x'||_1\le 1}\big(d_{\mathcal{M}\circ f,\mathcal{S},\epsilon}(x,x')\big) &\le \max_{\mathcal{S},a':||a-a'||_1\le 1}\Big(\Pr[\mathcal{M}(a)\in\mathcal{S}] - e^{\epsilon}\cdot\Pr[\mathcal{M}(a)\in\mathcal{S}]\Big)\\
&= \max_{\mathcal{S},a':||a-a'||_1\le 1}\big(d_{\mathcal{M},\mathcal{S},\epsilon}(a,a')\big).
\end{aligned}$$

By symmetry, we have,

$$\max_{\mathcal{S},x':||x-x'||_1\le 1}\big(d_{\mathcal{M}\circ f,\mathcal{S},\epsilon}(x',x)\big) \le \max_{\mathcal{S},a':||a-a'||_1\le 1}\big(d_{\mathcal{M},\mathcal{S},\epsilon}(a',a)\big).$$

Combining the above two inequalities with definition of smoothed DP, for any fixed x and $a = f(x)$, we have

$$\delta_{\epsilon,\mathcal{M}\circ f}(x) \le \delta_{\epsilon,\mathcal{M}}(a).$$

When $x\sim\vec{\pi}$, we know that $a\sim f(\vec{\pi})$. Then,

$$\begin{aligned}
\delta_{\mathcal{M}} &= \max_{\vec{\pi}_a\in f^n(\Pi)}\Big(\mathbb{E}_{a\sim\vec{\pi}_a}\big[\delta_{\epsilon,\mathcal{M}}(a)\big]\Big) = \max_{\vec{\pi}_x\in\Pi^n}\Big(\mathbb{E}_{x\sim\vec{\pi}_x}\big[\delta_{\epsilon,\mathcal{M}}(f(x))\big]\Big)\\
&\ge \max_{\vec{\pi}_x\in\Pi^n}\Big(\mathbb{E}_{x\sim\vec{\pi}_x}\big[\delta_{\epsilon,\mathcal{M}\circ f}(x)\big]\Big) = \delta_{\mathcal{M}\circ f},
\end{aligned}$$

where $\delta_{\mathcal{M}}$ (or $\delta_{\mathcal{M}\circ f}$) is the smoothed DP parameter for $\mathcal{M}$ (or $\mathcal{M}\circ f$). □

B.6.4 Proof of Proposition 5

Proposition 5 (Distribution reduction). *Given any $\epsilon,\delta\in\mathbb{R}_+$ and any Π_1 and Π_2 such that $\mathrm{CH}(\Pi_1) = \mathrm{CH}(\Pi_2)$, a anonymous mechanism $\mathcal{M}$ is (ϵ,δ,Π_1)-smoothed DP if and only if $\mathcal{M}$ is (ϵ,δ,Π_2)-smoothed DP.*

Proof. We let $\Pi^* = \{\pi_1^*, \cdots, \pi_p^*\}$ denote the vertices of $\mathrm{CH}(\Pi_1)$ or $\mathrm{CH}(\Pi_2)$. Then, for any distribution $\pi \in \Pi_1$, $\pi = \sum_{j=1}^{p} \alpha_j \cdot \pi_j^*$, where $\sum_{j=1}^{p} \alpha_j = 1$ and $\alpha_j \in [0,1]$ for any $j \in [p]$. Let $\vec{\pi}_{-i}$ to denote the distribution of the agents other than the i-th agent. Then, we know

$$\Pr[x|\vec{\pi}] = \Pr[x_{-i} \cup \{X_i\}|\vec{\pi}] = \Pr[x_{-i}|\vec{\pi}_{-i}] \cdot \Pr[X_i|\pi_i] = \Pr[x_{-i}|\vec{\pi}_{-i}] \cdot \left(\sum_{j=1}^{p} \alpha_{j,i} \cdot \Pr[X_i|\pi_j^*] \right),$$

where $\sum_{j=1}^{p} \alpha_{j,i} = 1$ and $\alpha_{j,i} \in [0,1]$ for any $j \in [p]$. Considering that the above decomposition works for any database, by induction, we have,

$$\begin{aligned}
\Pr[x|\vec{\pi}] &= \prod_{i=1}^{n} \sum_{j=1}^{p} \left(\alpha_{j,i} \cdot \Pr[X_i|\pi_j^*] \right) = \prod_{i=1}^{n} \sum_{j=1}^{p} \left(\alpha_{j,i} \cdot \Pr[x|\pi_j^*] \right) \\
&= \sum_{j_1, \cdots, j_n \in [p]} \left[\left(\prod_{i=1}^{n} \alpha_{j_i,i} \right) \cdot \Pr[x|(\pi_{j_1}^*, \cdots, \pi_{j_n}^*)] \right].
\end{aligned}$$

The above inequality shows that any for $\vec{\pi} \in \Pi_1$,

$$\begin{aligned}
\mathbb{E}_{x \sim \vec{\pi}}[\delta_{\epsilon,\mathcal{M}}(x)] &= \sum_{j_1, \cdots, j_n \in [p]} \left[\left(\prod_{i=1}^{n} \alpha_{j_i,i} \right) \cdot \mathbb{E}_{x \sim (\pi_{j_1}^*, \cdots, \pi_{j_n}^*)}[\delta_{\epsilon,\mathcal{M}}(x)] \right] \\
&\leq \max_{j_1, \cdots, j_n \in [p]} \left(\mathbb{E}_{x \sim (\pi_{j_1}^*, \cdots, \pi_{j_n}^*)}[\delta_{\epsilon,\mathcal{M}}(x)] \right) \\
&= \max_{\vec{\pi} \in \Pi^{*n}} \left(\mathbb{E}_{x \sim \vec{\pi}} \left[\delta_{\epsilon,\mathcal{M}}(x) \right] \right).
\end{aligned}$$

Considering that the above inequality holds for any $\vec{\pi} \in \mathrm{CH}(\Pi_1)$, then,

$$\max_{\vec{\pi} \in \Pi_1^n} \left(\mathbb{E}_{x \sim \vec{\pi}} \left[\delta_{\epsilon,\mathcal{M}}(x) \right] \right) \leq \max_{\vec{\pi} \in \Pi^{*n}} \left(\mathbb{E}_{x \sim \vec{\pi}} \left[\delta_{\epsilon,\mathcal{M}}(x) \right] \right).$$

Also note that $\Pi^* \subseteq \Pi_1$, we have,

$$\max_{\vec{\pi} \in \Pi_1^n} \left(\mathbb{E}_{x \sim \vec{\pi}} \left[\delta_{\epsilon,\mathcal{M}}(x) \right] \right) \geq \max_{\vec{\pi} \in \Pi^{*n}} \left(\mathbb{E}_{x \sim \vec{\pi}} \left[\delta_{\epsilon,\mathcal{M}}(x) \right] \right).$$

Then, by symmetry, we have,

$$\max_{\vec{\pi} \in \Pi_1^n} \left(\mathbb{E}_{x \sim \vec{\pi}} \left[\delta_{\epsilon,\mathcal{M}}(x) \right] \right) = \max_{\vec{\pi} \in \Pi^{*n}} \left(\mathbb{E}_{x \sim \vec{\pi}} \left[\delta_{\epsilon,\mathcal{M}}(x) \right] \right) = \max_{\vec{\pi} \in \Pi_2^n} \left(\mathbb{E}_{x \sim \vec{\pi}} \left[\delta_{\epsilon,\mathcal{M}}(x) \right] \right).$$

□

B.7 Missing Proofs in Section 4.5: The Mechanisms of Smoothed DP

B.7.1 The Missing Proof for Theorem 8

Theorem 8 (The DP and Smoothed-DP for SHM). *Using the notations in Algorithm 7, given any strictly positive set of distribution* Π*, any finite set* $\mathcal{A}$ *and any* $n, T \in \mathbb{Z}_+$*, we have the following properties on* $\mathcal{M}_H$*,*

1°(Smoothed DP upper bound) $\mathcal{M}_H$ is $\Big(\epsilon, \exp\big(-\Theta(\frac{(n-T)^2}{n})\big), \Pi\Big)$-smoothed DP for any $\epsilon \geq \ln\big(\frac{2n}{n-T}\big)$.

2°(DP lower bound) For any $\epsilon > 0$, there does not exist $\delta < \frac{T}{n}$ such that $\mathcal{M}_H$ is (ϵ, δ)-DP.

3°(Tightness of 1°) For any $\epsilon > 0$, there does not exist $\delta = \exp(-\omega(n))$ such that $\mathcal{M}_H$ is (ϵ, δ, Π)-smoothed DP.

Proof. We present our proof of 1° in the following three steps.

Step 0. Notations and Preparation.

Let $\vec{H} \triangleq \text{hist}(X_1, \cdots, X_n)$ (and $\vec{h} \triangleq \text{hist}(X_{j_1}, \cdots, X_{j_T})$) denote the histogram of the whole database (the T samples). H_i (and h_i) denote the i-th component of vector $\vec{H}$ (and $\vec{h}$). Let $m = |\mathcal{A}|$ be the size of the finite set $\mathcal{A}$. Note the sampling process in $\mathcal{M}_H$ is without replacement. Then,

$$\Pr\left[\vec{h}|\vec{H}\right] = \frac{1}{\binom{n}{T}} \cdot \prod_{i=1}^{m} \binom{H_i}{h_i}.$$

Then, we recall privacy loss when the output is $\vec{h}$. This loss, $\delta_{\mathcal{M}_H,\vec{h},\epsilon}(\vec{H}, \vec{H}')$, is mathematically defined as

$$\begin{aligned}\delta_{\mathcal{M}_H,\vec{h},\epsilon}(\vec{H}, \vec{H}') &\triangleq \Pr\left[\vec{h}|\vec{H}\right] - e^{\epsilon} \cdot \Pr\left[\vec{h}|\vec{H}'\right] \\ &= \Pr\left[\vec{h}|\vec{H}\right]\left(1 - e^{\epsilon} \cdot \frac{\Pr\left[\vec{h}|\vec{H}'\right]}{\Pr\left[\vec{h}|\vec{H}\right]}\right),\end{aligned}$$

where $\vec{H}$ and $\vec{H}'$ is different on at most one data. We sometimes write $\delta_{\mathcal{M}_H,\vec{h},\epsilon}(\vec{H}, \vec{H}')$ as $\delta_{\vec{h}}(\vec{H}, \vec{H}')$ when the context is clear. Also note that $\mathcal{S}$ can be any subset of the image space

of $\mathcal{M}_H$. Then, for any neighboring $\vec{H}$ and $\vec{H}'$, we have,

$$\max_{\mathcal{S}} \delta_{\mathcal{S}}(\vec{H}, \vec{H}') = \sum_{\vec{h}:\, \delta_{\vec{h}}(\vec{H},\vec{H}')>0} \delta_{\vec{h}}(\vec{H}, \vec{H}'). \tag{B.8}$$

Step 1. Bound $\delta(\vec{H})$ and $\max_{\mathcal{S}} \delta_{\mathcal{S}}(\vec{H}, \vec{H}')$.

W.l.o.g., we assume that $\vec{H}'$ has one more data of the i_1-th type while has one less data of the i_2-th type. Then,

$$\begin{aligned}
\frac{\Pr\left[\vec{h}|\vec{H}'\right]}{\Pr\left[\vec{h}|\vec{H}\right]} &= \prod_{i=1}^{m} \frac{\binom{H_i'}{h_i}}{\binom{H_i}{h_i}} = \frac{\binom{H_{i_1}-1}{h_{i_1}} \cdot \binom{H_{i_2}+1}{h_{i_2}}}{\binom{H_{i_1}}{h_{i_1}} \cdot \binom{H_{i_2}}{h_{i_2}}} \\
&= \left(1 - \frac{h_{i_1}}{H_{i_1}}\right) \cdot \left(1 + \frac{h_{i_2}}{H_{i_2} + 1 - h_{i_2}}\right) \\
&\geq 1 - \frac{h_{i_1}}{H_{i_1}}.
\end{aligned}$$

By the definition of $\delta_{\vec{h}}(\vec{H}, \vec{H}')$, we have,

$$\begin{aligned}
\delta_{\vec{h}}(\vec{H}, \vec{H}') &= \Pr\left[\vec{h}|\vec{H}\right] \left(1 - e^{\epsilon} \cdot \frac{\Pr\left[\vec{h}|\vec{H}'\right]}{\Pr\left[\vec{h}|\vec{H}\right]}\right) \\
&\leq \Pr\left[\vec{h}|\vec{H}\right] \left(1 - e^{\epsilon} \cdot \left(1 - \frac{h_{i_1}}{H_{i_1}}\right)\right).
\end{aligned}$$

Thus, when $\frac{h_{i_1}}{H_{i_1}} \geq 1 - e^{-\epsilon}$, we always have $\delta_{\vec{h}}(\vec{H}, \vec{H}') \leq 0$. We also note that $\delta_{\vec{h}}(\vec{H}, \vec{H}') \leq \Pr\left[\vec{h}|\vec{H}\right]$ for any $\vec{h}$. Then, we combine the above results with (B.8) and we have,

$$\begin{aligned}
\max_{\mathcal{S}} \delta_{\mathcal{S}}(\vec{H}, \vec{H}') &= \sum_{\vec{h}:\, \delta_{\vec{h}}(\vec{H},\vec{H}')>0} \delta_{\vec{h}}(\vec{H}, \vec{H}') \\
&\leq \sum_{\vec{h}:\, h_{i_1} > (1-e^{-\epsilon})H_{i_1}} \Pr\left[\vec{h}|\vec{H}\right] \\
&= \Pr\left[h_{i_1} > (1 - e^{-\epsilon})H_{i_1}|\vec{H}\right].
\end{aligned}$$

Note that $\mathbb{E}[h_{i_1}] = \frac{T}{n} \cdot H_{i_1}$. We apply Chernoff bound and have,

$$\max_{\mathcal{S}} \delta_{\mathcal{S}}(\vec{H}, \vec{H}') \leq \min\left(1,\ \exp\left(-\frac{1}{3} \cdot \left(\frac{1 - e^{-\epsilon} - \frac{T}{n}}{\frac{T}{n}}\right)^2 \cdot H_{i_1}\right)\right).$$

Letting $c = \epsilon - \ln\left(\frac{n}{n-T}\right)$, we have,

$$\max_{\mathcal{S}} \delta_{\mathcal{S}}(\vec{H}, \vec{H}') \leq \min\left(1,\ \exp\left(-\frac{(1-e^{-c})^2}{3} \cdot \left(\frac{n-T}{T}\right)^2 \cdot H_{i_1}\right)\right).$$

Using the same procedure, we have,

$$\max_{\mathcal{S}} \delta_{\mathcal{S}}(\vec{H}', \vec{H}) \leq \min\left(1,\ \exp\left(-\frac{(1-e^{-c})^2}{3} \cdot \left(\frac{n-T}{T}\right)^2 \cdot H_{i_2}\right)\right).$$

By the definition of smoothed DP, we have,

$$\begin{aligned}\delta(\vec{H}) &= \max\left(0, \max_{\mathcal{S},\vec{H}':\left|\left|\vec{H}-\vec{H}'\right|\right|_1 \leq 1} \delta_{\mathcal{S}}(\vec{H}', \vec{H}), \max_{\mathcal{S},\vec{H}':\left|\left|\vec{H}-\vec{H}'\right|\right|_1 \leq 1} \delta_{\mathcal{S}}(\vec{H}, \vec{H}')\right) \\ &\leq \min\left(1,\ \exp\left(-\frac{(1-e^{-c})^2}{3} \cdot \left(\frac{n-T}{T}\right)^2 \left(\min_i H_i\right)\right)\right).\end{aligned} \tag{B.9}$$

Step 2. The smoothed analysis to $\delta(\vec{H})$.

We note that Π is f-strictly positive, which means any distribution $\pi \in \Pi$'s PMF is no less than f. Here, f can be a function of m or n. f-strictly positive will reduce to the standard definition of strictly positive when f is a constant function. Given the f-strictly positive condition, by the definition of smoothed DP and for any constant $c_2 \in (0, 1)$, we have,

$$\begin{aligned}&\delta = \mathbb{E}_{\vec{H}}\left[\delta(\vec{H})\right] \\ \leq\ &\Pr\left[\min_i H_i \geq (1-c_2)\cdot f(m,n)\cdot n\right] \cdot \mathbb{E}\left[\delta(\vec{H}) | \min_i H_i \geq (1-c_2)\cdot f(m,n)\cdot n\right] \\ &\quad + \Pr\left[\min_i H_i < (1-c_2)\cdot f(m,n)\cdot n\right] \cdot \mathbb{E}\left[\delta(\vec{H}) | \min_i H_i < (1-c_2)\cdot f(m,n)\cdot n\right] \\ \leq\ &\mathbb{E}\left[\delta(\vec{H}) | \min_i H_i \geq (1-c_2)\cdot f(m,n)\cdot n\right] + \Pr\left[\min_i H_i < (1-c_2)\cdot f(m,n)\cdot n\right].\end{aligned}$$

Then, we apply Chernoff bound and union bound on the second term.

$$\begin{aligned}&\Pr\left[\min_i H_i < (1-c_2)\cdot f(m,n)\cdot n\right] \leq \sum_i \Pr\left[H_i < (1-c_2)\cdot f(m,n)\cdot n\right] \\ \leq\ &m \cdot \exp\left(-\frac{(c_2)^2}{2}\cdot f(m,n)\cdot n\right) = \exp\left(-\frac{(c_2)^2}{2}\cdot f(m,n)\cdot n + \ln m\right).\end{aligned}$$

For the first term, we directly apply inequality (B.9) and have,

$$\begin{aligned}&\mathbb{E}\left[\delta(\vec{H})|\min_i H_i \geq (1-c_2)\cdot f(m,n)\cdot n\right]\\ \leq\ &\min\left(1,\ \exp\left(-\frac{(1-e^{-c})^2}{3}\cdot\left(\frac{n-T}{T}\right)^2\cdot(1-c_2)\cdot f(m,n)\cdot n\right)\right).\end{aligned}$$

Combining the above bounds, we have

$$\delta \leq \exp\left(-\frac{(1-e^{-c})^2}{3}\cdot\left(\frac{n-T}{T}\right)^2\cdot(1-c_2)\cdot f(m,n)\cdot n\right)+\exp\left(-\frac{(c_2)^2}{2}\cdot f(m,n)\cdot n+\ln m\right).$$

Then, the 1° part of Theorem 8 follows by letting f be a constant function, $c=\ln 2$ and $c_2=0.5$.

2°. **The DP lower bound**.

We consider a pair of neighboring databases that x contains n data of the same type and x' contains $n-1$ of the same type while the rest data is in a different type. We use X_i to denote the different data in the two databases. Then, we have

$$\delta \geq \delta(x,x') \geq \Pr[X_i \text{ is sampled}] = \frac{T}{n}.$$

Then, the 2° part of Theorem 8 follows.

3°. **The tightness of** 1°.

In smoothed-DP, $\delta \geq \Pr[x]\cdot\delta(x)$. For the database x in the proof 2°, we have that

$$\Pr[x] \leq c_2^n = \exp\big(-\Theta(n)\big).$$

Then, we have,

$$\delta \geq \Pr[x]\cdot\delta(x) \geq \frac{T}{n}\cdot\exp\big(-\Theta(n)\big) = \exp\big(-\Theta(n)\big).$$

Then, the 3° part of Theorem 8 follows. □

B.7.2 The Proof for Theorem 9

Theorem 9 (The smoothed DP for continuous sampling average). *Using the notations above, given any set of strictly positive distribution over* $[0,1]$*, any* $T, n \in \mathbb{Z}_+$ *and any* $\epsilon \geq 0$*, there does not exist any* $\delta < \frac{T}{n}$ *such that* $\mathcal{M}_A$ *is* (ϵ, δ, Π)*-smoothed DP.*

Proof. Using the same notations as Algorithm 8, we let database $x = \{X_1, \cdots, X_n\}$. W.l.o.g., we assume the different data in x' is X_n and let $x' = \{X_1, \cdots, X_{n-1}, X'_n\}$. To simplify notations, we let $x_S = \{X_{j_1}, \cdots, X_{j_T}\}$ (or x'_S) as the sampled T data from x (or x'). To approach the "worst-case" of privacy loss, we set X'_n in the form that for all $\{i_1, \cdots, i_T\} \subseteq [n]$ and $\{i'_1, \cdots, i'_{T-1}\} \subseteq [n-1]$,

$$X'_n \neq \left(\sum_{j\in[T]} X_{i_j}\right) - \left(\sum_{j\in[T-1]} X_{i'_j}\right).$$

In other words, we let X'_n in the way that all subset containing X'_n will have a different average with any subset without X'_n. Because X'_n has a continuous support while T is a finite number, we can always find an X'_n to meet the above requirement. Then, we set

$$\mathcal{S} = \left\{\bar{x} : \bar{x} = X'_n + \sum\nolimits_{j\in[T-1]} X_{i'_j}, \text{ where } \{i'_1, \cdots, i'_{T-1}\} \subseteq [n-1]\right\}.$$

Then, for any database x and any $\epsilon \geq 0$, we have,

$$\begin{aligned} d_{\mathcal{M}_A,\mathcal{S},\epsilon}(x', x) &= \Pr\left[\mathcal{M}(x') \in \mathcal{S}\right] - e^{\epsilon} \cdot \Pr\left[\mathcal{M}(x) \in \mathcal{S}\right] \\ &= \Pr\left[\mathcal{M}(x') \in \mathcal{S}\right] = \Pr[X'_n \text{ is sampled}] = \frac{T}{n} \end{aligned}$$

We note that $\frac{T}{n}$ is a common bound for all x. Then, the theorem follows by the definition of smoothed DP. □

B.8 Smoothed DP and DP for Sampling with Replacement

Theorem 15 (DP and smoothed DP for SHM with replacement)**.** *Using the notations in algorithm 7, given any strictly positive set of distribution* Π *and any* $T, n \in \mathbb{Z}_+$ *such that* $T = o(n)$*, the counting algorithm* $\mathcal{M}_C$ *with replacement is* $\left(1 + \frac{2T}{n}, \exp\left(-\Theta(n)\right), \Pi\right)$*-smoothed DP. However, given any* $\epsilon \geq 0$*, there does not exist any* $\delta \leq \frac{T}{n+1}$ *such that* $\mathcal{M}_C$ *is*

(ϵ, δ)-*differentially private.*

Proof. For any counting algorithm $\mathcal{M}_C$, the order between balls does not change the value $\delta_{x,\epsilon,\mathcal{C}}$. Thus, $\delta_{x,\epsilon,\mathcal{C}}$ only depends on the total number of ball M. To simplify notations, we let δ_M to denote the tolerance probability when $\epsilon = 1 + \frac{2T}{n}$ and the database x contains M balls. Mathematically,

$$\delta_M \triangleq \delta_{1+\frac{2T}{n},\mathcal{M}_C}(x) \text{ when there are } M \text{ balls in } x.$$

We first prove the standard DP property of $\mathcal{M}_C$ by analysing δ_1. By Theorem 7, we have,

$$\begin{aligned}
\delta \geq \delta_1 &\geq \Pr\left[\mathcal{M}_C(M=1) \in \langle T\rangle \setminus \{0\}\right] - e^{\epsilon} \cdot \Pr\left[\mathcal{M}_C(M=0) \in \langle T\rangle \setminus \{0\}\right] \\
&= \Pr\left[\mathcal{M}_C(M=1) \in \langle T\rangle \setminus \{0\}\right] = 1 - \Pr\left[\mathcal{M}_C(M=1) = 0\right] \\
&= 1 - \left(1 - \frac{1}{n}\right)^T \geq \frac{T}{n+1}.
\end{aligned}$$

By now, we proved the standard DP part of theorem. In the next two steps, we will prove the smoothed DP part.

Step 1. Tight bound for δ_M.

To simplify notations, we define $P_{M,k} \triangleq \Pr(K=k)$ when there are n bins, M balls and T samples. When the sampling algorithm is with replacement,

$$P_{M,k} \triangleq \Pr(K=k) = \binom{T}{k}\left(\frac{M}{n}\right)^k \left(\frac{n-M}{n}\right)^{T-k}.$$

By the definition of δ_M, we have,

$$\begin{aligned}
\delta_M &= \max\left(0, \max_{\mathcal{S}, M' \in \{M-1, M+1\}} \left(\Pr\left[\mathcal{M}_C(M) \in \mathcal{S}\right] - e^{\epsilon} \cdot \Pr\left[\mathcal{M}_C(M') \in \mathcal{S}\right]\right)\right) \\
&= \max\left(0, \max_{\mathcal{S} \subseteq \langle T\rangle, M' \in \{M-1, M+1\}} \left[\sum_{\frac{k}{T} \in \mathcal{S}} \left(P_{M,k} - e^{\epsilon} \cdot P_{M',k}\right)\right]\right) \\
&= \max\left(0, \max_{\mathcal{S} \subseteq \langle T\rangle, M' \in \{M-1, M+1\}} \left[\sum_{\frac{k}{T} \in \mathcal{S}} P_{M,k}\left(1 - e^{\epsilon} \cdot \frac{P_{M',k}}{P_{M,k}}\right)\right]\right).
\end{aligned} \tag{B.10}$$

Step 1.1. Tight bound for $\frac{P_{M',k}}{P_{M,k}}$.

Using the above notation, we have

$$\begin{aligned}\frac{P_{M,k}}{P_{M-1,k}} &= \left(\frac{M}{M-1}\right)^k \left(\frac{n-M}{n-M+1}\right)^{T-k} \\ &= \left(1+\frac{1}{M-1}\right)^{(M-1)\cdot\frac{k}{M-1}} \cdot \left(1-\frac{1}{n-M+1}\right)^{(n-M+1)\cdot\frac{T-k}{n-M+1}} \\ &\leq \exp\left(\frac{k}{M-1} - \frac{T-k}{n-M+1}\right).\end{aligned} \tag{B.11}$$

Similarly, we have,

$$\begin{aligned}\frac{P_{M-1,k}}{P_{M,k}} &= \left(\frac{M-1}{M}\right)^k \left(\frac{n-M+1}{n-M}\right)^{T-k} \\ &= \left(1-\frac{1}{M}\right)^{M\cdot\frac{k}{M}} \cdot \left(1+\frac{1}{n-M}\right)^{(n-M)\cdot\frac{T-k}{n-M}} \\ &\leq \exp\left(\frac{T-k}{n-M} - \frac{k}{M}\right).\end{aligned} \tag{B.12}$$

Combining (B.11) and (B.12), we have

$$\begin{aligned}\exp\left(\frac{T-k}{n-M+1} - \frac{k}{M-1}\right) \leq \frac{P_{M-1,k}}{P_{M,k}} \leq \exp\left(\frac{T-k}{n-M} - \frac{k}{M}\right) \quad \text{and} \\ \exp\left(\frac{k}{M} - \frac{T-k}{n-M}\right) \leq \frac{P_{M+1,k}}{P_{M,k}} \leq \exp\left(\frac{k}{M+1} - \frac{T-k}{n-M-1}\right).\end{aligned} \tag{B.13}$$

Step 1.2. Upper bound for $P_{M,k}$ when $M < \min\{k, n/2\}$.

$$\begin{aligned}P_{M,k} =& \binom{T}{k}\left(\frac{M}{n}\right)^k \left(\frac{n-M}{n}\right)^{T-k} \leq \binom{T}{k}\left(\frac{M}{n}\right)^k < \left(\frac{e\cdot k}{k\cdot T}\right)^k \left(\frac{M}{n}\right)^k \\ &\leq \left(\frac{e\cdot k}{n}\right)^k \leq \left(\frac{e\cdot T}{n}\right)^k = O\left(\frac{T^k}{n^k}\right).\end{aligned}$$

Step 1.3. Bound δ_M.

By symmetry, we know that $\delta_M = \delta_{n-M}$. Thus, in all discussion of this step, we assume that $M \leq \lfloor n/2 \rfloor$. We note again that we assume $T = o(n)$. Thus, in (B.13), for any k and M, we have $\frac{T-k}{n-M-1} = o(1)$. Then, we discuss the value of $P_{M,k} - e^{\epsilon} \cdot P_{M',k}$ by the two cases of M'. In all following proofs, we set $\epsilon = \frac{2T}{n} = o(1)$.

When $M' = M + 1$, for any M and k, using the results in (B.10) and (B.13), we have,

$$\begin{aligned}
&\max_{\mathcal{S}} \left(\Pr\left[\mathcal{M}_C(M) \in \mathcal{S}\right] - e^{\epsilon} \cdot \Pr\left[\mathcal{M}_C(M+1) \in \mathcal{S}\right] \right) \\
= &\max_{\mathcal{S} \subseteq \langle T \rangle} \left[\sum_{\frac{k}{T} \in \mathcal{S}} P_{M,k} \left(1 - e^{\epsilon} \cdot \frac{P_{M+1,k}}{P_{M,k}} \right) \right] \quad \text{(using (B.10))} \\
\leq &\max_{\mathcal{S} \subseteq \langle T \rangle} \left[\sum_{\frac{k}{T} \in \mathcal{S}} P_{M,k} \left(1 - \exp\left(1 + \frac{2T}{n}\right) \cdot \exp\left(\frac{k}{M} - \frac{T-k}{n-M} \right) \right) \right] \quad \text{(using (B.13))}.
\end{aligned}$$

Note again that $M \leq n/2$ and we have,

$$\max_{\mathcal{S}} \left(\Pr\left[\mathcal{M}_C(M) \in \mathcal{S}\right] - e^{\epsilon} \cdot \Pr\left[\mathcal{M}_C(M+1) \in \mathcal{S}\right] \right) \leq \max_{\mathcal{S} \subseteq \langle T \rangle} \left[\sum_{\frac{k}{T} \in \mathcal{S}} P_{M,k} \left(1 - \exp\left(0\right)\right) \right] = 0.$$

When $M' = M - 1$, Following exactly the same procedure as the $M' = M - 1$ case, we know that for any $k \leq M - 1$ (or $k < M$),

$$P_{M,k} - e^{\epsilon} \cdot P_{M-1,k} \leq 0.$$

When $k \geq M$, from step 1.2, we have,

$$P_{M,k} - e^{\epsilon} \cdot P_{M-1,k} \leq P_{M,k} \leq \left(\frac{e \cdot T}{n} \right)^k.$$

Combining the above two inequalities, we have,

$$\begin{aligned}
&\max_{\mathcal{S}} \left(\Pr\left[\mathcal{M}_C(M) \in \mathcal{S}\right] - e^{\epsilon} \cdot \Pr\left[\mathcal{M}_C(M-1) \in \mathcal{S}\right] \right) \\
= &\max_{\mathcal{S} \subseteq \langle T \rangle} \left[\sum_{\frac{k}{T} \in \mathcal{S}} P_{M,k} \left(1 - e^{\epsilon} \cdot \frac{P_{M+1,k}}{P_{M,k}} \right) \right] \quad \text{(using (B.10))} \\
\leq &\sum_{k \geq M} \left(\frac{e \cdot T}{n} \right)^k \quad \text{(using the above two inequalities)} \\
\leq &\frac{\left(\frac{e \cdot T}{n}\right)^M}{1 - \frac{e \cdot T}{n}} = O\left(\frac{T^M}{n^M} \right).
\end{aligned}$$

Combining the above two cases with the analysis for standard DP, for any $M \leq \lfloor n/2 \rfloor$, we

have,

$$\delta_M = \begin{cases} O\left(\frac{T^M}{n^M}\right) & \text{if } M \geq 2 \\ \Theta\left(\frac{T}{n}\right) & \text{if } M = 1 \\ 0 & \text{otherwise} \end{cases}$$

Step 2. The smoothed analysis for δ_M.

By the definition of strictly positive distributions, we know that there must exist constant c such that any events happens with probability at least c. Then, by Chernoff bound, we have,

$$\Pr\left[M \leq \frac{c \cdot n}{2}\right] \leq \exp\left(-\frac{c \cdot n}{8}\right) \quad \text{and} \quad \Pr\left[M \geq 1 - \frac{c \cdot n}{2}\right] \leq \exp\left(-\frac{c \cdot n}{8}\right).$$

By the definition of smoothed DP, we have,

$$\begin{aligned} \delta &= \max_{\vec{\pi}} \left(\mathbb{E}_{x \sim \vec{\pi}} \left[\delta_M\right] \right) \\ &\leq \Pr\left[M \in (0, cn/2) \cup (1 - cn/2, 1)\right] \cdot \delta_1 + \Pr[M \in (cn/2, 1 - cn/2)] \cdot \delta_{cn/2} \\ &\leq \exp\big(- \Theta(n)\big) + 1 \cdot \end{aligned}$$

Then, theorem 15 follows by combining the above two bounds. □

B.9 The Detailed Settings of Experiments and Extra Experimental Results

B.9.1 The Detailed Settings of Experiments

The election experiment. Similar with the motivating example, we only consider the top-2 candidates in each congressional district. The top-2 candidates are Trump and Bidden in any congressional districts. In the discussion of our main text, DC refers to Washington, D.C. and NE-2 refers to Nebraska's 2nd congressional district. The distribution of each congressional district is calculated using the election results of the 2020 US presidential election. For example, in DC, each agent votes Bidden with 94.46% probability while votes Trump with 5.54% probability.

The SGD experiment. We formally define the 8-bit quantized Gaussian distribution $\mathcal{N}_{\text{8-bit}}$, which is used in the set of distributions Π of our SGD experiment. To simplify notations, for any interval $I = (i_1, i_2]$, we define $p_{I,\mu,\sigma}$ as the probability of Gaussian distribution

$\mathcal{N}(\mu, \sigma^2)$ on I. Formally,

$$p_{\mu,\sigma}(I) = \int_{i_1}^{i_2} \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2} \cdot \left(\frac{x-\mu}{\sigma}\right)^2\right) \mathrm{d}x. \tag{B.14}$$

Then, we formally define the 8-bit quantized Gaussian distribution as follows,

Definition 13 (8-bit quantized Gaussian distribution)**.** *Using the notations above, Given any μ and σ, the probability mass function of $\mathcal{N}_{8\text{-}bit}(\mu, \sigma^2)$ is*

$$\mathrm{PMF}(x) = \begin{cases} p_{,\mu,\sigma}\left(\left(\frac{i-128}{256}, \frac{i-127}{256}\right]\right) / Z_{\mu,\sigma} & \text{if } x = \frac{i-128}{256} \text{ where } i = \{0, \cdots, 255\} \\ 0 & \text{otherwise} \end{cases}, \tag{B.15}$$

where the normalization factor $Z_{\mu,\sigma} = p_{,\mu,\sigma}\big((-0.5, 0.5]\big)$.

By replacing the Gaussian distribution with Laplacian distribution, we defined 8-bit quantized Laplacian distribution $\mathcal{L}_{8\text{-bit}}(\mu, \sigma^2)$, which will be used in our extra experiments for the SGD with quantized gradients.

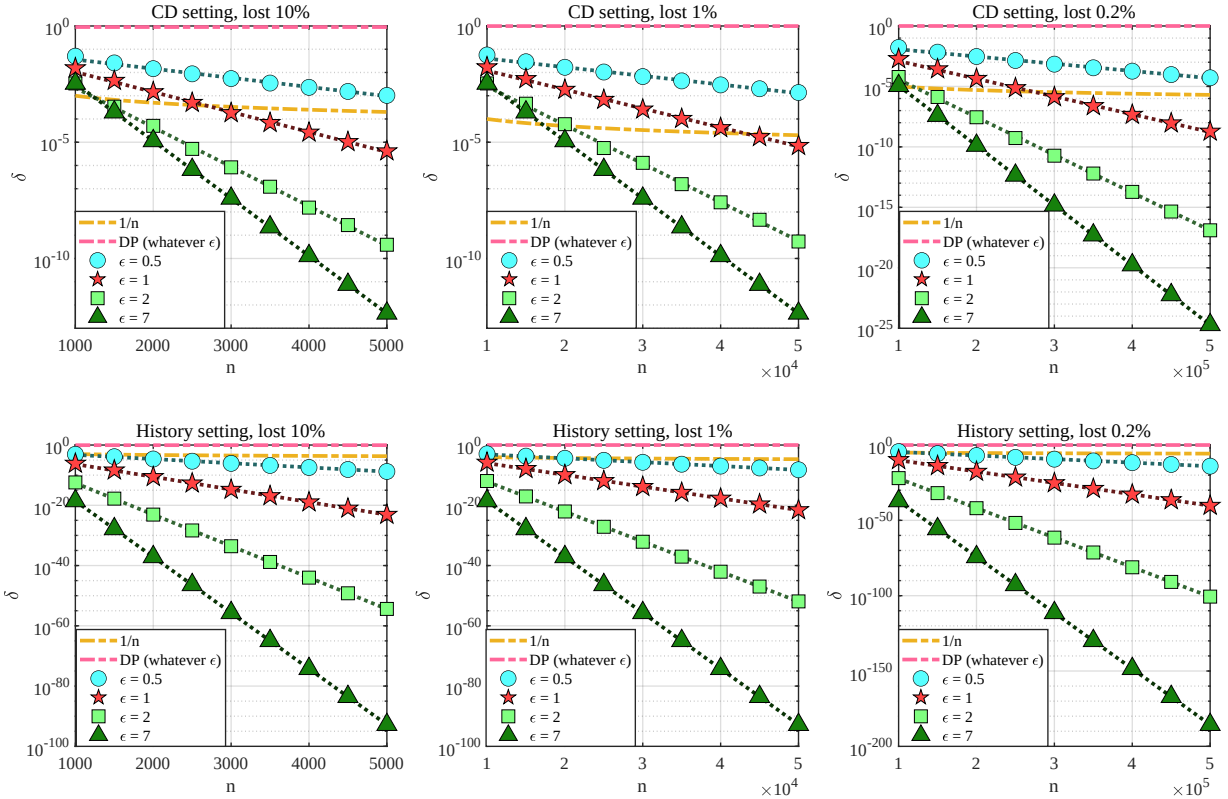

Figure B.2: Smoothed DP and DP for elections under congressional districts (CD) setting and history setting.

B.9.2 Extra Experimental Results

Smoothed DP in elections. Figure B.2 presents the numerical results for election under different settings. The congressional districts (CD) setting is the same setting of Π as Figure 4.2 (left), which is the distributions of all 57 congressional districts in 2020 presidential election. In the history setting, the set of distributions Π includes the distribution of all historical elections since 1920. All results in Figure B.2 shows similar trend as Figure 4.2 (left). No matter what is the set of distributions Π and no matter what is the ratio of votes got lost, the δ parameter of smoothed DP is always exponentially small in the number of agent n.

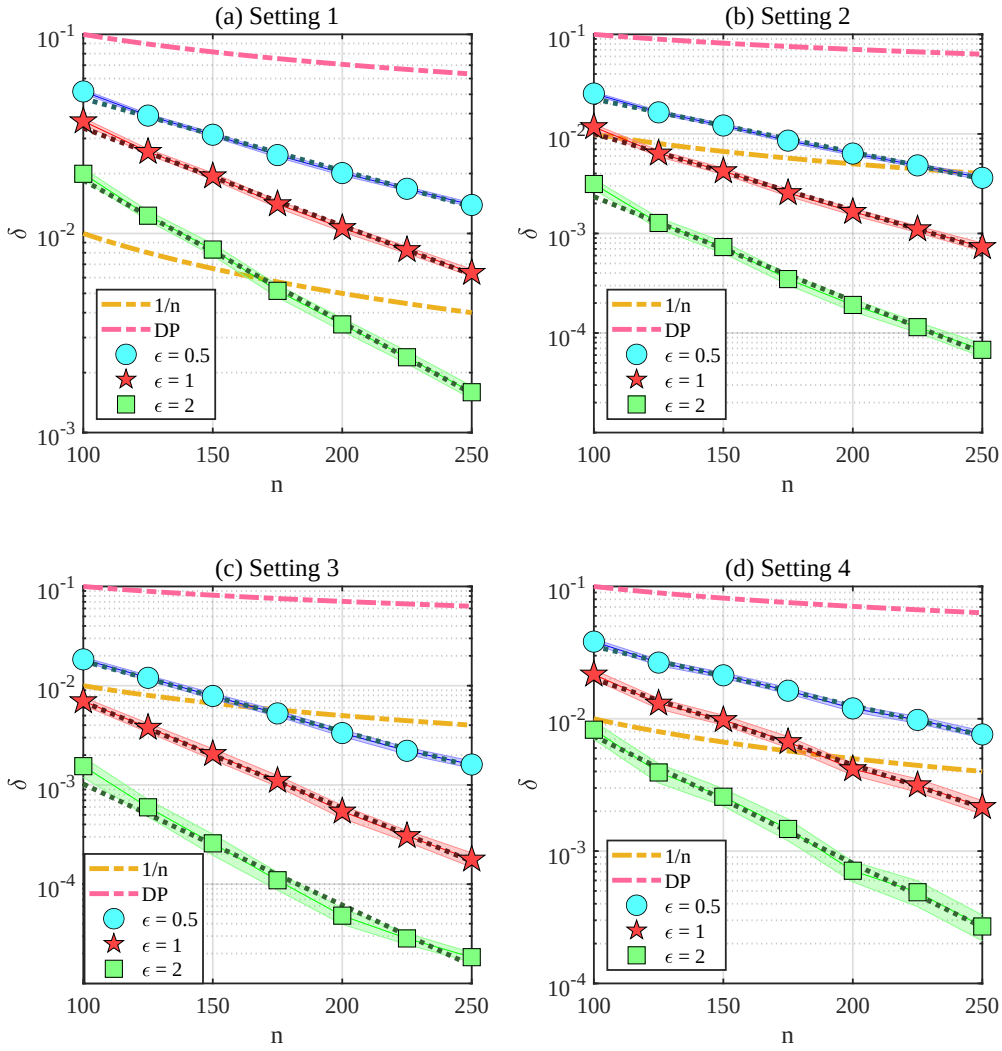

Figure B.3: Smoothed DP and DP for SGD with 8-bit gradient quantization under different settings. The shaded region shows the 99% confidence interval of δ's.

SGD with 8-bit gradient quantization. Figure B.3 presents the SGD with 8-bit gradient quantization experiment under different settings. All four settings in Figure B.3 shares the same setting as Figure 4.2 (right), except the set of distributions Π. The detailed settings of Π are as follows.

- Setting 1: $\Pi = \{\mathcal{N}_{\text{8-bit}}(0, 0.15^2),\ \mathcal{N}_{\text{8-bit}}(0.2, 0.15^2)\}$,
- Setting 2: $\Pi = \{\mathcal{N}_{\text{8-bit}}(0, 0.1^2),\ \mathcal{N}_{\text{8-bit}}(0.2, 0.1^2)\}$,
- Setting 3: $\Pi = \{\mathcal{N}_{\text{8-bit}}(0, 0.12^2),\ \mathcal{L}_{\text{8-bit}}(0, 0.12^2)\}$,
- Setting 4: $\Pi = \{\mathcal{L}_{\text{8-bit}}(0, 0.12^2),\ \mathcal{L}_{\text{8-bit}}(0.2, 0.12^2)\}$,

where $\mathcal{L}_{\text{8-bit}}(\mu, \sigma^2)$ is the 8-bit quantized Laplacian distribution defined in Appendix B.9.1. All results in Figure B.3 shows similar information as 4.2 (right). It says that the δ of smoothed DP is exponentially small in the number of agents n. As the number of iterations in Figure B.3 is just $\sim 10\%$ of the number of iteration of Figure 4.2 (right), we observe some random fluctuations in Figure B.3.

APPENDIX C
APPENDIX FOR CHAPTER 5

C.1 Table of Notations

Table C.1: Table of notations for Chapter 5.

Notation	Description	Firstly Appeared
d	dimension of norm	Sec. 5.1
k	number of interested pixels	Sec. 5.1
$\boldsymbol{x}$	(original) image	Sec. 5.2
$\mathcal{C}$	the set of labels	Sec. 5.2
$\boldsymbol{g}(\cdot,\cdot)$	interpretation algorithm	Sec. 5.2
$\tilde{\boldsymbol{x}}$	perturbed image	Sec. 5.2
$(\cdot)_i$	the i^{th} pixel of $(\cdot)$	Sec. 5.2
L	the constraint of ℓ_d-norm attacks	Sec. 5.2
$T_k(\cdot)$	the top-k components of $(\cdot)$	Sec. 5.2
$V_k(\boldsymbol{x},\tilde{\boldsymbol{x}})$	the top-k overlap ratio between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$	Sec. 5.2
β	robustness parameter in β-top-k robustness	Sec. 5.2
$\grave{\boldsymbol{g}}(\cdot)$	a randomized interpretation algorithm	Sec. 5.2
$R_\alpha(P\|\|Q)$	α-degree Rényi robustness of P on Q	Sec. 5.2
T	the number of noisy interpretation maps	Sec. 5.3
$\Gamma(\cdot)$	gamma function	Sec. 5.3
$\mathcal{G}$	generalized normal distribution	Sec. 5.3
b	the shape factor of generalized normal distribution	Sec. 5.3
σ	the standard deviation of a distribution	Sec. 5.3
μ	the expectation of a distribution	Sec. 5.3
$\boldsymbol{v}$	the scoring vector	Sec. 5.3
n	the number of pixels	Sec. 5.3
d_{prior}	the prior knowledge from the defender	Sec. 5.3
n	the number of pixels	Sec. 5.3
$\boldsymbol{m}$	the expected interpretation map of original image	Sec. 5.4.1
$\tilde{\boldsymbol{m}}$	the expected interpretation map of perturbed image	Sec. 5.4.1
k_0	$\lfloor(1-\beta)k\rfloor+1$	Sec. 5.4.1
$\mathcal{S}$	$\{k-k_0,\cdots,k+k_0+1\}$	Sec. 5.4.1
ϵ_{robust}	$-\ln\left(2k_0\left(\frac{1}{2k_0}\sum_{i^*\in\mathcal{S}}(m_i^*)^{1-\alpha}\right)^{\frac{1}{1-\alpha}}+\sum_{i^*\notin\mathcal{S}}m_i^*\right)$	Sec. 5.4.1

Portions of this chapter have previously appeared as: A. Liu, X. Chen, S. Liu, L. Xia, and C. Gan."Certifiably robust interpretation via Rényi differential privacy," *Artif. Intell.* vol. 313, Dec. 2022, Art. no. 103787.

Notation	Description	Firstly Appeared
$f_{\boldsymbol{v}}$	re-scaling function with scoring vector $\boldsymbol{v}$	Sec. 5.4.1
$\mathbb{1}(\cdot)$	the indicator function	Sec. 5.4.1
$\epsilon_\alpha\left(\frac{L}{\sigma}\right)$	$\frac{1}{\alpha-1}\ln\left[\frac{\alpha}{\alpha-1}\exp\left(\frac{(\alpha-1)L)}{\sqrt{2}\sigma}\right)+\frac{\alpha-1}{2\alpha-1}\exp\left(\frac{-\alpha L}{\sqrt{2}\sigma}\right)\right]$	Sec. 5.4.1
$\epsilon_b\left(\frac{L}{\sigma}\right)$	$\frac{1}{\Gamma(1/b)}\sum_{i=1}^{b/2}\binom{2i}{b}\left(\frac{L}{\sigma^*}\right)^{2i}\Gamma\left(\frac{b+1-2i}{b}\right)$	Sec. 5.4.1
σ^*	$\sqrt{\frac{\Gamma(1/b)}{\Gamma(3/b)}}\,\sigma$	Sec. 5.4.1
$\hat{\epsilon}_{robust}$	the estimated ϵ_{robust} using T samples	Sec. 5.4.2
B	the set of top-k components' subscripts	Sec. 5.6
η and k^*	the shape parameters of scoring vector $\boldsymbol{v}$	Sec. 5.6
lr	learning rate	Sec. 5.6

C.2 Additional Proofs

C.2.1 A Readable Proof for Theorem 11

Proof of Theorem 11. Mathematically, we calculate the minimum change in Rényi divergence to violet the requirement of β-top-k robustness. That is, calculate

$$\min_{\mathbb{E}[\tilde{\boldsymbol{m}}]\ s.t.\ V_k(\mathbb{E}[\boldsymbol{m}],\mathbb{E}[\tilde{\boldsymbol{m}}])\leq\beta} R_\alpha(\mathbb{E}[\boldsymbol{m}]\,||\,\mathbb{E}[\tilde{\boldsymbol{m}}]).$$

In all remaining proof of Theorem 11 we slightly abuse the notation and let m_i and $\tilde{m}_i$ to represent $\mathbb{E}[m_i]$ and $\mathbb{E}[\tilde{m}_i]$ respectively. W.L.O.G., we assume $m_1 \geq \cdots \geq m_n$. Then, we show that we must have $\tilde{m}_1 \geq \cdots \geq \tilde{m}_{k-k_0-1} \geq \tilde{m}_{k-k_0} = \cdots = \tilde{m}_{k+k_0+1} \geq \tilde{m}_{k+k_0+2} \geq \cdots \geq \tilde{m}_n$ to reach the minimum of Rényi divergence. To simplify notation, we let $s(\mathbb{E}[\boldsymbol{m}]\,||\,\mathbb{E}[\tilde{\boldsymbol{m}}]) = \sum_{i=1}^n m_i\left(\frac{\tilde{m}_i}{m_i}\right)^\alpha$. One can see that $s(\mathbb{E}[\boldsymbol{m}]\,||\,\mathbb{E}[\tilde{\boldsymbol{m}}])$ reaches the minimum on the same condition as $R_\alpha(\mathbb{E}[\boldsymbol{m}]\,||\,\mathbb{E}[\tilde{\boldsymbol{m}}])$. To outline our proof, we prove the following claims one by one (See section C.2.2 for the detailed proofs).

(i) [Natural] To reach the minimum, there are exactly k_0 different components in the top-k of $\mathbb{E}[\boldsymbol{m}]$ and $\mathbb{E}[\tilde{\boldsymbol{m}}]$.

(ii) To reach the minimum, $\tilde{m}_{k-k_0}, \cdots, \tilde{m}_k$ are not in the top-k of $\mathbb{E}[\tilde{\boldsymbol{m}}]$.

(ii') To reach the minimum, $\tilde{m}_{k+1}, \cdots, \tilde{m}_{k+k_0+1}$ must appear in the top-k of $\mathbb{E}[\tilde{\boldsymbol{m}}]$.

(iii') [Proved in [123]] To reach the minimum, we must have $\tilde{m}_i \geq \tilde{m}_j$ for any $i \leq j$.

One can see that the above claims on $\mathbb{E}[\tilde{\boldsymbol{m}}]$ are equivalent to the following KKT

condition:

$$\begin{aligned}
&\min_{\tilde{m}_1\cdots,\tilde{m}_n} \sum_{i=1}^{n} m_i \left(\frac{\tilde{m}_i}{m_i}\right)^{\alpha} \\
\text{subject to} \quad &\sum_{i=1}^{n} \tilde{m}_i = 1 \\
\text{subject to} \quad &\tilde{m}_j - \tilde{m}_i \le 0,\ \forall i < k \\
\text{subject to} \quad &-\tilde{m}_i \le 0,\ \forall i \in [k] \\
\text{subject to} \quad &\tilde{m}_j - \tilde{m}_i = 0,\ \forall i, j \in \mathcal{S}
\end{aligned}$$

Solving it, we know $s(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}])$ reaches minimum when

$$\forall i \in \mathcal{S},\ \tilde{m}_i = \frac{\breve{m}}{2k_0\breve{m} + \sum_{i\notin\mathcal{S}} m_i} \quad \text{and} \quad \forall i \notin \mathcal{S},\ \tilde{m}_i = \frac{m_i}{2k_0\breve{m} + \sum_{i\notin\mathcal{S}} m_i},$$

where $\breve{m} = \left(\frac{1}{2k_0}\sum_{i\in\mathcal{S}}(m_i)^{1-\alpha}\right)^{\frac{1}{1-\alpha}}$. By plugging in the above condition, we have,

$$\min_{\mathbb{E}[\tilde{\boldsymbol{m}}]\ s.t.\ V_k(\mathbb{E}[\boldsymbol{m}],\mathbb{E}[\tilde{\boldsymbol{m}}])\le\beta} R_\alpha(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}]) = -\ln\left(2k_0\left(\frac{1}{2k_0}\sum_{i\in\mathcal{S}}(m_i)^{1-\alpha}\right)^{\frac{1}{1-\alpha}} + \sum_{i\notin\mathcal{S}} m_i\right).$$

Then, we know β-top-k robustness condition will be filled if the Rényi divergence do not exceed the above value. □

C.2.2 Proofs to the Claims Used in the Proof of Theorem 11

(i) [Natural] To reach the minimum, there is exactly k_0 different components in the top-k of $\mathbb{E}[\boldsymbol{m}]$ and $\mathbb{E}[\tilde{\boldsymbol{m}}]$.

Proof. Assume that $i_1, \cdots, i_{k_0+j}$ are the components not in the top-k of $\boldsymbol{m}$ but in the top-k of $\tilde{\boldsymbol{m}}$. Similarly, we let $i'_1, \cdots, i'_{k_0+j}$ to denote the components in the top-k of $\boldsymbol{m}$ but not in the top-k of $\tilde{\boldsymbol{m}}$. Consider we have another $\tilde{\boldsymbol{m}}^{(2)}$ with the same value with $\tilde{\boldsymbol{m}}$ while $\tilde{m}_{i_{k_0+j}}$ is replaced by $\tilde{m}_{i'_{k_0+j}}$. In other words, there is $k_0 + j$ displacements in the top-k of $\tilde{\boldsymbol{m}}$ while

there is $k_0 + j - 1$ displacements in the top-k of $\tilde{\boldsymbol{m}}^{(2)}$. Thus,

$$\begin{aligned} & s\left(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}^{(2)}]\right) - s\left(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}]\right) \\ = & \left(\frac{\left(\tilde{m}_{i'_{k_0+j}}\right)^{\alpha}}{\left(m_{i_{k_0+j}}\right)^{\alpha-1}} + \frac{\left(\tilde{m}_{i_{k_0+j}}\right)^{\alpha}}{\left(m_{i'_{k_0+j}}\right)^{\alpha-1}} \right) - \left(\frac{\left(\tilde{m}_{i_{k_0+j}}\right)^{\alpha}}{\left(m_{i'_{k_0+j}}\right)^{\alpha-1}} + \frac{\left(\tilde{m}_{i'_{k_0+j}}\right)^{\alpha}}{\left(m_{i_{k_0+j}}\right)^{\alpha-1}} \right). \end{aligned}$$

Because $m_{i_{k_0+j}} \geq m_{i'_{k_0+j}}$ and $\tilde{m}_{i_{k_0+j}} \leq \tilde{m}_{i'_{k_0+j}}$, we know $s\left(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}^{(2)}]\right) - s\left(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}]\right) \leq 0$. Thus, we know reducing the number of misplacement in top-k can reduce the value of $s\left(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}]\right)$. If requires at least k_0 displacements, the minimum of $s\left(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}]\right)$ must be reached when there is exactly k_0 displacements. □

(ii) To reach the minimum, $\tilde{m}_{k-k_0}, \cdots, \tilde{m}_k$ are not in the top-k of $\mathbb{E}[\tilde{\boldsymbol{m}}]$.

Proof. Assume that $i_1, \cdots, i_{k_0}$ are the components not in the top-k of $\boldsymbol{m}$ but in the top-k of $\tilde{\boldsymbol{m}}$. Similarly, we let $i'_1, \cdots, i'_{k_0}$ to denote the components in the top-k of $\boldsymbol{m}$ but not in the top-k of $\tilde{\boldsymbol{m}}$. Consider we have another $\tilde{\boldsymbol{m}}^{(2)}$ with the same value with $\tilde{\boldsymbol{m}}$ while $\tilde{m}_{i_j}$ is replaced by $\tilde{m}_{j'}$, where $\tilde{m}_{j'}$ is in the top-k of $\tilde{\boldsymbol{m}}$ and $m_{j'} \geq m_{i_j}$. In other words, $\tilde{m}^{(2)}_{j'}$ is no longer in the top-k components of $\tilde{\boldsymbol{m}}$ while $\tilde{m}^{(2)}_{i_j}$ goes back to the top-k of $\tilde{m}^{(2)}_{i_j}$. Note again that $m_{j'} \geq m_{i_j}$ and $j' \leq i_j$. Thus,

$$\begin{aligned} & s\left(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}^{(2)}]\right) - s\left(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}]\right) \\ = & \left(\frac{\left(\tilde{m}_{i_j}\right)^{\alpha}}{\left(m_{j'}\right)^{\alpha-1}} + \frac{\left(\tilde{m}_{j'}\right)^{\alpha}}{\left(m_{i_j}\right)^{\alpha-1}} \right) - \left(\frac{\left(\tilde{m}_{j'}\right)^{\alpha}}{\left(m_{j'}\right)^{\alpha-1}} + \frac{\left(\tilde{m}_{i_j}\right)^{\alpha}}{\left(m_{i_j}\right)^{\alpha-1}} \right). \end{aligned}$$

Note that $\tilde{m}_{i_j} \leq \tilde{m}_i$, we have $s\left(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}^{(2)}]\right) - s\left(\mathbb{E}[\boldsymbol{m}] \,||\, \mathbb{E}[\tilde{\boldsymbol{m}}]\right) \geq 0$. Thus, we know that "moving a larger component out from top-k while moving a smaller component back to top-k" will make $s(\cdot||\cdot)$ larger. Then, (ii) follows by induction. □

(ii') To reach the minimum, $\tilde{m}_{k+1}, \cdots, \tilde{m}_{k+k_0+1}$ must appear in the top-k of $\mathbb{E}[\tilde{\boldsymbol{m}}]$, which holds according to the same reasoning as (ii).

C.2.3 Proof of Lemma 7

Lemma 7 For any positive even shape factor b, letting $x \sim \mathcal{G}(0,\sigma,b)$ and $\tilde{x} \sim \mathcal{G}(L,\sigma,b)$, we have

$$\lim_{\alpha\to 1^+} D_\alpha(x \,||\, \tilde{x}) = \epsilon_b\big(L/\sigma\big).$$

Proof of Lemma 7. By the definition of Rényi divergence, we have,

$$\lim_{\alpha\to 1} D_\alpha(x||x') = \lim_{\alpha\to 1}\left(\frac{1}{\alpha-1}\cdot\ln\int_{-\infty}^{\infty}\frac{b\cdot\exp\left(-(\frac{x-\mu}{\sigma^*})^b\right)}{2\sigma^*\Gamma(1/b)}\frac{\exp\left(-\alpha(\frac{x}{\sigma^*})^b\right)}{\exp\left(-\alpha(\frac{x-\mu}{\sigma^*})^b\right)}\mathrm{d}x\right).$$

Because we interested in the behaviour of D_α when $\alpha\to 1$, to simplify notation, we let $\delta=\alpha-1$. when $\alpha-1\to 0$, we apply first-order approximation to $\exp(\cdot)$ and have,

$$\lim_{\alpha\to 1} D_\alpha(x||x') = \lim_{\delta\to 0^+}\left(-\delta^{-1}\cdot\ln\int_{-\infty}^{\infty}\frac{b\cdot\exp\left(-(\frac{x}{\sigma^*})^b\right)}{2\sigma^*\Gamma(1/b)}\left(1-\delta(\frac{x}{\sigma^*})^b+\delta(\frac{x-\mu}{\sigma^*})^b\right)\mathrm{d}x\right).$$

Because $\frac{b\cdot\exp\left(-(\frac{x}{\sigma^*})^b\right)}{2\sigma^*\Gamma(1/b)}$ is the PDF of $\mathcal{G}(0,\sigma,b)$, we have,

$$\lim_{\alpha\to 1} D_\alpha(x||x') = \lim_{\delta\to 0^+}\left(-\delta^{-1}\cdot\ln\left[1+\delta\int_{-\infty}^{\infty}\frac{b\cdot\exp\left(-(\frac{x}{\sigma^*})^b\right)}{2\sigma^*\Gamma(1/b)}\left(-(\frac{x}{\sigma^*})^b+(\frac{x-\mu}{\sigma^*})^b\right)\mathrm{d}x\right]\right).$$

By applying first-order approximation to $\ln(\cdot)$, we have,

$$\lim_{\alpha\to 1} D_\alpha(x||x') = \int_{-\infty}^{\infty}\frac{b\cdot\exp\left(-(\frac{x}{\sigma^*})^b\right)}{2\sigma^*\Gamma(1/b)}\left((\frac{x-\mu}{\sigma^*})^b-(\frac{x}{\sigma^*})^b\right)\mathrm{d}x.$$

Then, we expand $\left(\frac{x-\mu}{\sigma^*}\right)^b$ and have,

$$\lim_{\alpha\to 1} D_\alpha(x||x') = \int_{-\infty}^{\infty}\frac{b\cdot\exp\left(-(\frac{x}{\sigma^*})^b\right)}{2\sigma^*\Gamma(1/b)}\sum_{i=1}^{b}\binom{i}{b}\left(\frac{\mu}{\sigma^*}\right)^i\left(\frac{x}{\sigma^*}\right)^{b-i}\mathrm{d}x.$$

When $b-i$ is odd, $\exp\left(-(\frac{x}{\sigma^*})^b\right)\left(\frac{x}{\sigma^*}\right)^{b-i}$ is an odd function and the integral will be zeros. When $b-i$ is even, it will become an even function. Thus, we have,

$$\lim_{\alpha\to 1} D_\alpha(x||x') = \int_{0}^{\infty}\frac{b\cdot\exp\left(-(\frac{x}{\sigma^*})^b\right)}{\sigma^*\Gamma(1/b)}\sum_{i=1}^{b/2}\binom{2i}{b}\left(\frac{\mu}{\sigma^*}\right)^{2i}\left(\frac{x}{\sigma^*}\right)^{b-2i}\mathrm{d}x.$$

Through substituting $\left(\frac{x}{\sigma^*}\right)^b$, we have,

$$\lim_{\alpha\to 1} D_\alpha(x||x') = \int_0^\infty \frac{\exp(-y)}{\Gamma(1/b)} \sum_{i=1}^{b/2} \binom{2i}{b} \left(\frac{\mu}{\sigma^*}\right)^{2i} y^{(1-2i)/b} \mathrm{d}y.$$

Finally, Lemma 7 follows by the definition of Gamma function. □

C.2.4 Formal Version of Theorem 13 with Proof

Before presenting Theorem 13, we first provide a technical lemma for the concentration bound for $\phi(\boldsymbol{m}) = 2k_0 \left(\frac{1}{2k_0}\sum_{i\in\mathcal{S}}(m_i^*)^{1-\alpha}\right)^{\frac{1}{1-\alpha}} - \sum_{i\in\mathcal{S}} m_i^*$. Considering that $||\boldsymbol{m}||_1 = 1$, we know $\epsilon_{\text{robust}} = -\ln(1+\phi(\boldsymbol{m}))$. To simplify notation, we let $\psi(\boldsymbol{m}) = \left(\frac{1}{2k_0}\sum_{i\in\mathcal{S}}(m_i^*)^{1-\alpha}\right)^{\frac{1}{1-\alpha}}$. Thus, we have $\phi(\boldsymbol{m}) = 2k_0\psi(\boldsymbol{m}) - \sum_{i\in\mathcal{S}} m_i^*$

Lemma 15. *Using the notations above, we have:*

$$\Pr\left[\phi(\boldsymbol{m}) \le \phi(\hat{\boldsymbol{m}}) + \delta\right] \le 1 - n\cdot\exp\left(-2T\delta^2\left[\psi^\alpha(\hat{\boldsymbol{m}})\left(\sum_{i\in\mathcal{S}}\hat{m}_{i^*}^{-\alpha}\right) - 2k_0\right]^2\right).$$

Proof. We first prove the following statement: if for all $i \in [n]$, $|\hat{m}_i - m_i| \le \delta$, we always have:

$$\phi(\boldsymbol{m}) - \phi(\hat{\boldsymbol{m}}) \le \delta\left[\psi^\alpha(\hat{\boldsymbol{m}})\left(\sum_{i\in\mathcal{S}}\hat{m}_{i^*}^{-\alpha}\right) - 2k_0\right]. \tag{C.1}$$

Because $\phi(\cdot)$ is a concave function when all components of the input vector are positive. Thus, if $\phi(\boldsymbol{m}) \ge \phi(\hat{\boldsymbol{m}})$ we have,

$$\phi(\boldsymbol{m}) - \phi(\hat{\boldsymbol{m}}) \le \sum_{i\in\mathcal{S}}(m_i^* - \hat{m}_i^*)\cdot\left.\frac{\partial\phi(\boldsymbol{m})}{\partial m_i^*}\right|_{\boldsymbol{m}=\hat{\boldsymbol{m}}}.$$

Then, inequality (C.1) follows by the fact that $\frac{\partial\phi(\boldsymbol{m})}{\partial m_i^*} = (m_i^*)^{-\alpha}\cdot\psi^\alpha(\boldsymbol{m})$.

Then, according to Hoeffding bound, we have:

$$\Pr\left[\forall i\in[n],\ |\hat{m}_i - m_i| \le \delta\right] \ge 1 - n\cdot\exp\left(-2T\delta^2\right).$$

Because $[\forall i \in [n],\ |\hat{m}_i - m_i| \le \delta]$ is a special case of $[\phi(\boldsymbol{m}) \le \phi(\hat{\boldsymbol{m}}) + \delta]$, Lemma 15 follows. □

Then, we formally present Theorem 13:

Theorem 13 (formal) *Using the notations above, we have,*

$$\begin{aligned}&\Pr\left[\epsilon_{robust} \geq \hat{\epsilon}_{robust} - \delta\right] \\ &\leq 1 - n \cdot \exp\left(-2T(e^{-\delta}-1)^2(\phi(\hat{\boldsymbol{m}})-1)^2\left[\psi^{\alpha}(\hat{\boldsymbol{m}})\left(\sum_{i\in\mathcal{S}}\hat{m}_{i^*}^{-\alpha}\right) - 2k_0\right]^2\right).\end{aligned}$$

Proof. Theorem 13 follows by applying $\epsilon_{\text{robust}} = -\ln(1+\phi(\boldsymbol{m}))$ to Lemma 15. □

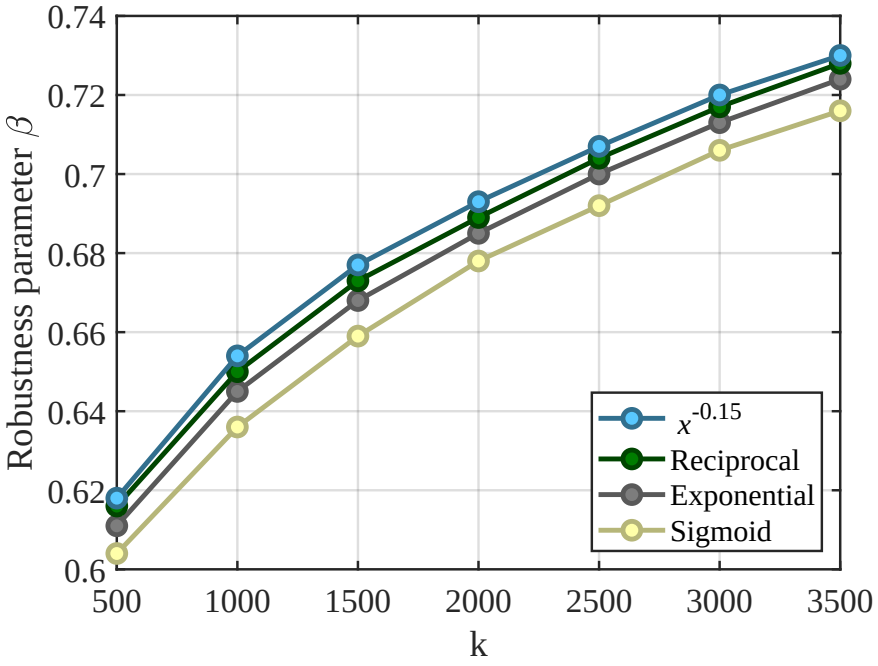

Figure C.1: The β-top-k robustness of Rényi-Robust-Smooth (RRS) in comparison with Sparsified SmoothGrad (SSG) of ResNet-50. (E) and (T) corresponds to the experimental and theoretical robustness, respectively.

C.3 Additional Experimental Results

C.3.1 The Robustness of Rényi-Robust-Smooth Under Different Scoring Vectors

Figure C.1 Compares Rényi-Robust-Smooth under different Scoring vectors $\boldsymbol{v}$. We note that the theoretical robustness guarantee of Rényi-Robust-Smooth does not depend on a specific form of $\boldsymbol{v}$. Thus, we select the scoring vectors according to the experimental ro-

bustness. Experimentally, we found that the most robust setting is $\boldsymbol{v} = (v_1, \cdots, v_n)$ where $v_i = Z^{-1} \cdot i^{-0.15}$ and $Z = \sum_{i=1}^{n} i^{-0.15}$.

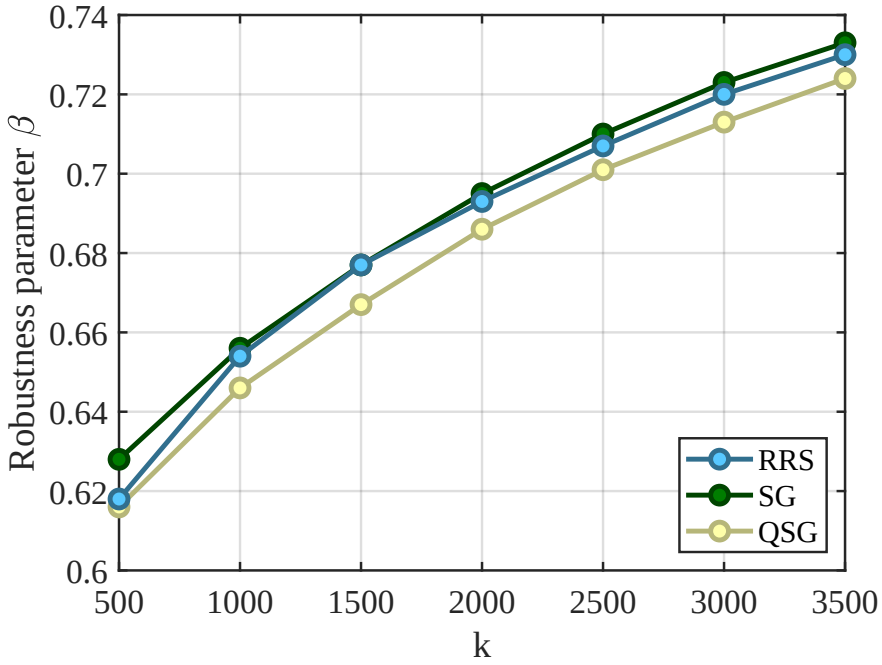

Figure C.2: The β-top-k robustness of Rényi-Robust-Smooth (RRS) in comparison with Quadratic SmoothGrad (QSG) and SmoothGrad(SG).

C.3.2 Compare SSG with SmoothGrad and Quadratic SmoothGrad

Figure C.2 shows that our RRS is ∼1% more robust than the widely-used Quadratic SmoothGrad[112]. However, another widely-used algorithm, SmoothGrad, is slightly (∼0.3%) more robust than Rényi-Robust-Smooth. It is an interesting future direction to prove the theoretical guarantees of SmoothGrad (if it has). The standard deviations of noises in Quadratic SmoothGrad and SmoothGrad are set as the same value as the standard deviation of noise in RRS. All other experimental settings of Figure C.2 are the same as Figure 5.6.

C.3.3 Additional Experiments on ResNet-50

Figure C.3 compares the β-top-k robustness of Rényi-Robust-Smooth with Sparsified SmoothGrad when the backbone of the neural network is ResNet-50. All other settings of C.3 are the same as Figure 5.6. The experimental robustness curves of ResNet-50 are similar to the curves for VGG. However, the theoretical bound becomes less tight for the ResNet backbone.

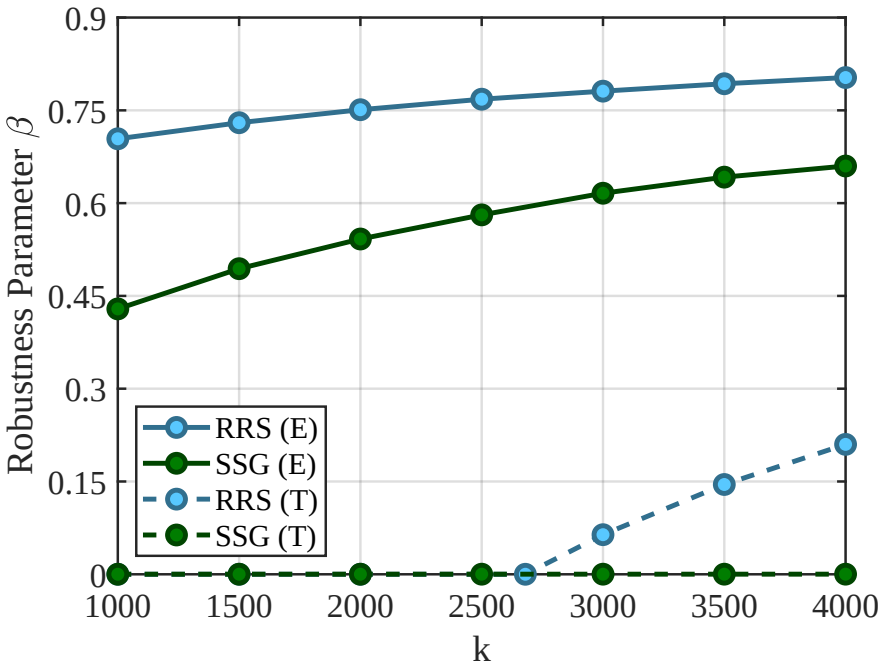

Figure C.3: The β-top-k robustness of Rényi-Robust-Smooth (RRS) in comparison with Sparsified SmoothGrad (SSG) for ResNet-50. (E) and (T) corresponds to the experimental and theoretical robustness, respectively.

Figure C.4 shows the computational efficiency-robustness tradeoff when the backbone of neural network is ResNet-50. All other settings of Figure C.4 are the same as Figure 5.8. The robustness parameter β in Figure C.4 (left) is similar as Figure 5.8 (left). Figure C.4 (right) shows that the GPU consumption of RRS is almost the same as SSG for ResNet-50 backbone.

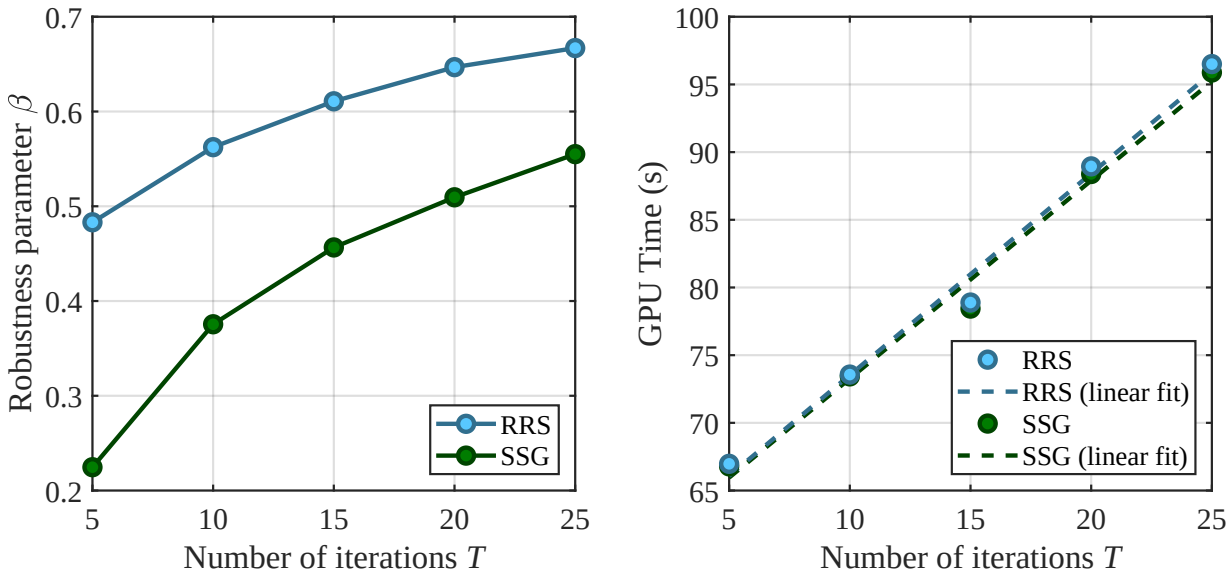

Figure C.4: Computational Efficiency-Robustness Tradeoff for ResNet-50 backbone. Left: the β-top-k robustness when the computational resources is highly constrained. Right: GPU consumption of RRS and SSG ($\sim 1\%$ less than RRS). The dashed lines show the linear fittings of RRS's and SSG's GPU consumption.