THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Mathématiques et Informatique
Unité de recherche : Laboratoire d'Informatique de Grenoble

Algorithmes privés et équitables pour une prise de décision séquentielle fiable

Private and Fair Algorithms for Trustworthy Sequential Decision-Making

Présentée par :

Alexandre RIO

Direction de thèse :

Sihem AMER-YAHIA DIRECTRICE DE RECHERCHE, CNRS DELEGATION ALPES	Directrice de thèse
Marta SOARE MAITRESSE DE CONFERENCE, Université d'Orléans	Co-encadrante de thèse
Merwan BARLIER IGR, HUAWEI	Co-encadrant de thèse
Igor COLIN IGR, HUAWEI	Co-encadrant de thèse

Rapporteurs :

EMILIE KAUFMANN
CHARGEE DE RECHERCHE HDR, CENTRE INRIA DE L'UNIVERSITE DE LILLE
LIVA RALAIVOLA
DOCTEUR EN SCIENCES HDR, CRITEO

Thèse soutenue publiquement le **30 juin 2025**, devant le jury composé de :

KIM THANG NGUYEN, PROFESSEUR DES UNIVERSITES, GRENOBLE INP - UGA	Président
SIHEM AMER-YAHIA, DIRECTRICE DE RECHERCHE, CNRS DELEGATION ALPES	Directrice de thèse
EMILIE KAUFMANN, CHARGEE DE RECHERCHE HDR, CENTRE INRIA DE L'UNIVERSITE DE LILLE	Rapporteure
LIVA RALAIVOLA, DOCTEUR EN SCIENCES HDR, CRITEO	Rapporteur
MATTHIEU GEIST, DOCTEUR EN SCIENCES HDR, EARTH SPECIES PROJECT	Examinateur
AURELIEN BELLET, DIRECTEUR DE RECHERCHE, CENTRE INRIA D'UNIVERSITE COTE D'AZUR	Examinateur

REMERCIEMENTS

Je tiens tout particulièrement à remercier ma directrice de thèse Sihem Amer-Yahia, ainsi que mes encadrants Marta Soare, Merwan Barlier et Igor Colin. Cela a été un grand plaisir de travailler avec vous, et j'ai énormément appris auprès de chacun d'entre vous.

Un grand merci également à Emilie Kaufmann et Liva Ralaivola pour avoir rapporté ma thèse, ainsi qu'au reste du jury : Kim Thang Nguyen, Matthieu Geist et Aurélien Bellet. Je suis reconnaissant d'avoir pu soutenir ma thèse devant un jury d'une telle qualité scientifique.

J'ai aussi eu la chance d'être si bien entouré au Noah's Ark Lab de Huawei, au sein duquel j'ai réalisé la plus grande partie de ma thèse. J'y ai rencontré des personnes fantastiques, sans lesquelles ces années n'auraient pas eu la même saveur.

Je remercie du fond du coeur mes parents, mon oncle et ma grand-mère pour leur soutien sans faille depuis le début. Cette thèse est l'aboutissement d'un long parcours que je n'aurais pu mener sans vous. Je vous serai toujours reconnaissant.

Enfin, merci à ma compagne Olivia pour m'avoir si bien épaulé au quotidien durant ces trois années. Tu as vécu cette thèse aussi intensément que moi, et j'ai beaucoup de chance de t'avoir à mes côtés.

ABSTRACT

Algorithms are increasingly driving decision-making in the real world, impacting people's lives and society at large. This has unsurprisingly sparked ethical concerns, such as privacy and fairness, which are becoming ever more urgent to address. Tackling ethical issues in this context is challenging and often conflicts with the primary performance-oriented goals. However, increasingly stringent legal regulations and strategic interests provide strong incentives for technology organizations to invest in the development of trustworthy solutions.

In this context, this thesis contributes to the development of algorithms that integrate ethical issues, especially privacy and fairness, without sacrificing performance in sequential decision-making — a setting that covers a wide range of problems where a series of decisions is taken over time. This effort is addressed through the lens of three sequential decision-making problems with many practical applications.

The thesis first addresses the problem of best arm identification in multi-agent multi-armed bandits, a framework used to optimize distributed large-scale systems. In this setting, we identify inter-agent communications as a critical privacy vulnerability. We therefore develop methods that rely on private communications without compromising the benefits of multi-agent collaboration.

The second part of this thesis deals with privacy-preserving deep reinforcement learning. While reinforcement learning is increasingly deployed in real-world scenarios with sensitive data, the field lacks scalable private solutions. This thesis aims to fill this gap by proposing deep reinforcement learning algorithms with formal differentially privacy guarantees. It first proposes a private model-based approach for the offline setting, before investigating theoretically-grounded differentially private policy gradient methods.

Finally, the thesis investigates the integration of producer fairness in bundle recommendation, where users are served with sets of compatible and complementary items, and fairness of exposure is measured over entire recommendation sequences. After formalizing this challenging task, both exact and heuristic solutions are introduced, effectively balancing fairness of exposure and bundle quality.

Résumé

Les algorithmes pilotent de plus en plus la prise de décision dans le monde réel, influençant la vie des individus et la société dans son ensemble. Cela a, sans surprise, suscité des préoccupations éthiques, telles que la confidentialité et l'équité, qui deviennent de plus en plus urgentes à traiter. Aborder ces problématiques éthiques dans le contexte de la prise de décision automatisée est complexe et entre souvent en conflit avec les objectifs principaux axés sur la performance. Toutefois, un cadre réglementaire de plus en plus strict et de nombreux enjeux stratégiques incitent fortement les organisations technologiques à investir dans le développement de solutions intégrant ces aspects éthiques.

Dans ce contexte, cette thèse contribue au développement d'algorithmes intégrant ces dimensions éthiques (en particulier la confidentialité et l'équité) sans compromettre la performance dans des problèmes de prise de décision séquentielle, un cadre englobant de nombreux scénarios où des décisions sont prises successivement au cours du temps. Ce sujet est abordé sous le prisme de trois problèmes de prise de décision séquentielle aux multiples applications pratiques.

La thèse aborde d'abord le problème de l'identification du meilleur bras dans des bandits multi-agent, un cadre utilisé pour optimiser des systèmes distribués à grande échelle. Dans ce contexte, nous identifions les communications entre agents comme une vulnérabilité critique en matière de confidentialité. Nous développons donc des méthodes reposant sur des communications privées, sans compromettre les avantages de la collaboration multi-agent.

La deuxième partie de cette thèse porte sur l'apprentissage par renforcement profond sur des données privées. Alors que l'apprentissage par renforcement est de plus en plus utilisé dans des scénarios réels impliquant des données sensibles, le domaine manque encore de solutions privées applicables à grande échelle. Cette thèse vise à répondre à cette problématique en proposant des algorithmes d'apprentissage par renforcement profond dotés de garanties formelles de confidentialité différentielle.

Enfin, la thèse explore l'intégration de l'équité dans la recommandation de paniers d'articles, où les utilisateurs se voient proposer des ensembles d'éléments compatibles et complémentaires, et où l'équité d'exposition est mesurée sur l'ensemble d'une séquence de recommandations. Après avoir formalisé cette tâche complexe, des solutions exactes et heuristiques sont proposées, permettant de concilier efficacement équité et qualité des paniers recommandés.

Table of Contents

Page

1 Introduction . . . 1
- 1.1 Ethical Concerns in Sequential Decision-Making . . . 1
  - 1.1.1 Privacy . . . 2
  - 1.1.2 Fairness . . . 3
- 1.2 The Cost of Privacy and Fairness . . . 4
- 1.3 Trustworthy Decision-Making at Huawei: Regulatory and Strategic Context . . . 5
- 1.4 Outline and Contributions . . . 6

2 Background: Trustworthy Machine Learning . . . 9
- 2.1 Differential Privacy . . . 9
  - 2.1.1 Formalism and Main Concepts . . . 9
  - 2.1.2 Achieving Differential Privacy . . . 12
  - 2.1.3 The Limitations of the DP Framework . . . 18
  - 2.1.4 Beyond Differential Privacy . . . 19
- 2.2 Algorithmic Fairness . . . 20
  - 2.2.1 Formalism and Main Concepts . . . 20
  - 2.2.2 Achieving Fairness . . . 27

3 Pure Exploration in Multi-Agent Bandits with Private Communications 31
- 3.1 Context . . . 31
- 3.2 Related Work . . . 33
- 3.3 Preliminaries . . . 34
  - 3.3.1 Multi-Agent Bandits . . . 34
  - 3.3.2 Multi-Agent BAI with Private Communications . . . 36
  - 3.3.3 Successive Elimination in Multi-Agent Bandits . . . 37
- 3.4 Multi-Agent Successive Elimination with Differentially Private Communications 39
  - 3.4.1 Differentially Private Successive Elimination . . . 39
  - 3.4.2 A DP Algorithm for Multi-Agent BAI . . . 40
- 3.5 Best Arm Privacy in Multi-Agent BAI . . . 46
  - 3.5.1 Multi-Agent Successive Elimination with (α, η)-Private Communications 47
  - 3.5.2 Improved Privacy with Message Corruption . . . 47
- 3.6 Experiments . . . 53
  - 3.6.1 Experimental Setup . . . 53
  - 3.6.2 Empirical Findings . . . 54

3.7 Chapter Summary . . . 57

4 Privacy-Preserving Deep Reinforcement Learning . . . 58
4.1 Context . . . 59
4.2 Contributions . . . 60
4.2.1 Differentially Private Offline Reinforcement Learning . . . 61
4.2.2 Differentially Private Policy Gradient . . . 61
4.3 Related Work . . . 62
4.4 Differentially Private Reinforcement Learning . . . 63
4.4.1 Reinforcement Learning . . . 63
4.4.2 Differential Privacy in Reinforcement Learning . . . 67
4.5 Differentially Private Offline Model-Based Reinforcement Learning . . . 71
4.5.1 Model Learning with Differential Privacy . . . 72
4.5.2 Policy Optimization under a Private Model . . . 76
4.5.3 Experiments . . . 82
4.6 Differentially Private Policy Gradient . . . 90
4.6.1 Policy Gradient and Trust-Region Methods . . . 90
4.6.2 A Theoretical Private Policy Gradient . . . 91
4.6.3 Practical Algorithm . . . 99
4.6.4 Experiments . . . 103
4.7 Chapter Summary . . . 109

5 Promoting Producer-Fairness in Bundle Recommendation . . . 110
5.1 Context . . . 110
5.2 Related Work . . . 113
5.3 Preliminaries . . . 114
5.3.1 Bundles . . . 115
5.3.2 Exposure and P-Fairness . . . 115
5.4 The P-Fair Bundle Recommendation Problem . . . 117
5.4.1 General Problem . . . 117
5.4.2 A Relaxed Formulation . . . 118
5.5 Exact and Heuristic Solutions . . . 120
5.5.1 An Integer Linear Programming Formulation . . . 120
5.5.2 Heuristic Solutions . . . 120
5.6 Experiments . . . 122
5.6.1 Experimental Setup . . . 122
5.6.2 Empirical Findings . . . 127
5.7 Chapter Summary . . . 131

6 Conclusion . . . 132
6.1 Research Summary . . . 132
6.2 Perspectives . . . 133

References . . . 148

Chapter One

Introduction

Many problems require making a series of decisions over time, a process known as *sequential decision-making*. Using data and algorithms to automate sequential decision-making, in pursuit of greater efficiency, enhanced accuracy, and cost savings, has become ubiquitous [Rodríguez, 2022]. As a result, algorithm-driven decisions — sometimes devoid of any human supervision — have gradually become a part of people's lives, unavoidably raising ethical concerns over issues like privacy and fairness [Mökander et al., 2021]. This thesis addresses the development of trustworthy algorithms for sequential decision-making in practical, application-driven settings. It places particular emphasis on privacy-preserving methods in bandits and reinforcement learning, while also exploring the related issue of fairness in recommendation systems.

1.1 Ethical Concerns in Sequential Decision-Making

Examples of algorithm-driven sequential decision-making in the real world are plentiful. From social media to e-commerce, machine-learning-based recommendation systems have shaped the way we consume products and content [Ziyuan et al., 2024]. Automatic decision-making powered by multi-armed bandits and reinforcement learning (RL) also fuels everyday industrial technologies and software services. For instance, the usage of multi-armed bandits to optimize telecommunication networks and power grids [Spyrou et al., 2024] has been gaining ground, affecting people's access to energy digital services. Reinforcement learning is also increasingly deployed in the real world, spanning areas as diverse as content recommendation [Afsar et al., 2023], autonomous driving [Kiran et al., 2020], ride-sharing [Qin et al., 2021] and healthcare [Yu et al., 2019]. It is also at the core of the success of large language models (LLMs) and conversational agents. In particular, RL from human feedback (RLHF) [Kaufmann et al., 2023] has enabled the alignment of LLMs with human expectations, a key component in their democratization, and RL has recently shown its potential to unlock advanced reasoning capabilities in these models [DeepSeek-AI, 2025].

While automating decision-making based on powerful tools like reinforcement learning

provides undeniable benefits, its rising influence on society has sparked significant ethical concerns [Leslie, 2019]. Among these worries, **privacy** and **fairness** stand out as key issues.

1.1.1 Privacy

Automating decision-making processes typically requires large amounts of data [Duan et al., 2019]. Useful data can be publicly available (*e.g.* open-source datasets), but is often *private*. Private data is often understood as either *personal data* — data belonging to individuals and potentially containing sensitive information (*e.g.*, medical record, consumption behavior) — or as *proprietary data* — data owned by organizations and giving them a competitive advantage (*e.g.*, sensor measurements, optimal parameters of a system). Personal and proprietary data may of course overlap since a company can own personal data collected from its users. Using private data can be necessary to produce useful models: personal data is often needed to provide accurate predictions or personalized recommendations, and companies rely on their valuable proprietary data to gain a competitive edge.

Example 1. *Personalized recommendation.* To provide accurate and relevant recommendations to its users, a recommendation algorithm must be trained on historical data collected from past user interactions. For instance, an e-commerce recommendation system will extract patterns from customer purchases, product viewings, and demographic information.

Example 2. *LLM fine-tuning.* An LLM fine-tuned to provide personalized medical advice must be provided with real patient data including health conditions and medical records. In an industrial context, post-training an LLM for a specific use case, *e.g.*, document analysis or intelligent customer support, will require large amounts of valuable company data.

Example 3. *Advanced system optimization.* Using bandits or RL to optimize an advanced system, *e.g.*, a telecommunication network, requires collecting large amounts of system data. This data may contain valuable information that is a product of the company's expertise.

Unfortunately, machine learning models are known to retain and leak information about their training data. According to prior research, one of the main sources of privacy leakage is overfitting, when the model captures specific details of the training data samples [Yeom et al., 2018]. In particular, deep neural networks and larger architectures tend to be more prone to privacy leakage because of their tendency to overfit the training data. Many powerful privacy attacks have been developed to exploit such opportunities [Rigaki and García, 2024]. Among them, membership inference attacks (MIAs) [Shokri et al., 2017] aim to infer the presence of a target data record in the training dataset of a model. Multi-armed bandits and RL, core frameworks for sequential decision-making, are not immune to these threats [Dubey and Pentland, 2020, Gomrokchi et al., 2023]. In this emerging era of LLM-based decision-making, addressing privacy concerns in RL is becoming an increasingly pressing challenge, calling for dedicated solutions that remain largely underdeveloped [Fan et al., 2025].

A breach of personal data can harm people in many ways, including theft of sensitive information, targeted exploitation of the stolen data, and reputation damage for both the at-fault company and the individuals. On the other hand, the disclosure of important proprietary data can have serious impacts on a company, such as a loss of competitive advantage, financial losses, and legal implications.

Observation 1

Using machine learning for sequential decision-making makes vulnerable the private data used in training, with serious consequences for individuals and organizations alike.

1.1.2 Fairness

Algorithms are notoriously prone to biases and can make unfair decisions which, if left unchecked, can have serious consequences on individuals and society [Kordzadeh, 2022]. There are many causes of algorithmic unfairness, including the replication of existing biases from the training dataset, missing data, ill-defined algorithmic objectives, and the exploitation of proxy attributes for sensitive attributes [Pessach and Shmueli, 2020]. Examples of unfair algorithmic behavior are plentiful, particularly in recommendation systems.

Example 4. *Popularity bias in e-commerce.* Recommendation algorithms are notably subject to popularity bias, that is the tendency of the algorithm to over-recommend the most popular items. Indeed, they generally optimize for an average accuracy metric over the user population, encouraging recommendations that would satisfy the majority [Bellogín et al., 2017]. This phenomenon can create a "rich get richer" situation, where the popularity of the already most popular items is reinforced while making it harder for new and less popular items to be discovered by users. On e-commerce platforms, such unfair behavior is detrimental to every stakeholder: the consumers, who have fewer opportunities to access niche content and face reduced diversity in the recommendations, the producers, some of whom miss out on potential revenues, and the platform itself, as all parties get a worse experience overall.

Example 5. *Demographic bias in hiring.* Another striking example is the use of recommendation systems during a hiring process [Fabris et al., 2025]; for instance, to recommend a candidate to interview among a list of selected applicants. If men have been historically favored over women in recruitment decisions, a recommendation algorithm trained on historical data — with the goal to maximize the probability that the applicant selected for interview eventually gets hired — will tend to unfairly choose men over women, since men have had a higher likelihood of being hired in the past. This is a typical example of the replication of existing biases from algorithms with real consequences on the lives of individuals.

Observation 2

Letting algorithms and machine learning drive sequential decision-making can lead to biased and unfair decisions, leading to significant socioeconomic consequences and loss of opportunities for individuals and organizations alike.

Definition: Trustworthy Sequential Decision-Making

Focusing solely on performance-oriented goals — *e.g.*, accuracy, reward, or engagement — while neglecting ethical concerns like privacy and fairness can yield harmful outcomes for both individuals and organizations. Addressing these ethical questions is central to **trustworthy sequential decision-making**: the development of methods for automating sequential decision-making that are not only useful and efficient but also can be trusted by the various stakeholders and are aligned with societal values.

1.2 The Cost of Privacy and Fairness

Unfortunately, guaranteeing privacy and fairness introduces unique challenges. In fact, these goals are often at odds with the primary performance objectives of the businesses or institutions that deploy automatic decision-making tools in the real world.

Protecting privacy generally involves removing or obscuring information from the data, leading to worse-performing models. *Differential privacy* (DP) [Dwork, 2006], the gold standard approach to privacy protection in machine learning, adds random noise at different steps in the learning pipeline, therefore perturbing the model convergence or predictions. Obtaining algorithms that enjoy both high performance and strong privacy guarantees is an ongoing challenge in sequential decision-making. RL — already notoriously unstable and challenging in non-private settings — becomes an especially difficult task under privacy constraints, and numerous obstacles remain before privacy-preserving solutions can be deployed in real-world settings [Fan et al., 2025]. Moreover, in the case of DP, it is not fully understood what level of privacy should be aimed for in practice, as its theoretical framework is deemed overly restrictive for deep learning applications [Ponomareva et al., 2023].

Encouraging fair decisions may also be to the detriment of accuracy [Liu and Vicente, 2022]. For instance, in recommendation systems, a fairness-aware algorithm may willingly recommend suboptimal items to a user to satisfy fairness requirements, trading off consumer satisfaction for producer fairness. It may also harm algorithm efficiency, as adding fairness constraints to the optimization process can lead to more computations.

Observation 3

Addressing trustworthy sequential decision-making may therefore not align with standard performance-oriented objectives and generally involves a trade-off.

1.3 Trustworthy Decision-Making at Huawei: Regulatory and Strategic Context

Organizations — the primary drivers of automatic decision-making technologies — aiming to integrate privacy-preserving or fairness-aware mechanisms must therefore be willing to compromise on traditional views of performance and value, which may be challenging to incentivize. However, we can identify several powerful incentives that make it worthwhile for an organization like Huawei, as well as other companies and institutions, to engage with this issue:

- **Legal compliance:** Depending on where they operate, businesses may have to adhere to regulations enforcing them to protect data privacy and ensure fairness. In Europe, the General Data Protection Regulation (GDPR) regulates the usage, storage, and processing of personal data, particularly when aimed at automatic decision-making. Comparable policies exist, for instance, in individual US states, such as the California Consumer Privacy Act (CCPA), and in China, with the Personal Information Protection Law (PIPL). Differential privacy aligns with the spirit of these laws, while providing even stronger guarantees. Fairness is also concerned by regulations, albeit in narrower scopes. For instance, the Civil Rights Act of 1964 in the US prohibits discrimination on the basis of protected attributes. Businesses failing to comply may receive significant sanctions, incentivizing them to implement best ethical practices.
- **Consumer power, trust, and reputation:** More aware consumers in today's world increasingly consider ethics when buying a product [Luttrell and Petty, 2021]. As they drive the markets, this incentivizes companies to value privacy and fairness to remain competitive. Ethical principles are also becoming key aspects of a company's trust and reputation [Ateeq and Milhem, 2024].
- **Long-term strategy:** Optimizing for performance only can be shortsighted, and considering other dimensions like privacy and fairness can enhance long-term user experience and satisfaction [Martínez-Navalón et al., 2021, Yang and Lee, 2024], ultimately benefiting the company. Businesses may also possess valuable proprietary data or engineering expertise that should be protected from competitors, warranting the implementation of privacy-preserving mechanisms.

Observation 4

Engaging with ethical considerations such as privacy and fairness is not only desirable for society, but also of strategic importance for organizations like Huawei.

	BAI in Multi-Agent Bandits	**Deep Reinforcement Learning**	**Bundle Recommendation**
Sequential aspect	Sequence of interactions with the environment	Markov Decision Process	Product exposure across a sequence of consumers
Decision aspect	Automatic parameter or option selection	Action (*e.g.*, treatment dose, prompt completion)	Recommendation of a set of products
Ethical concerns	Privacy	Privacy	Fairness
Consequences	Theft of personal or company data	Theft of personal or company data	Loss of revenues for producers
Solution	Best arm privacy, DP	DP	Producer-fairness of exposure
Trade-off	Decreased accuracy or increased sample complexity	Decreased performance or increased training time	Decreased recommendation quality

Table 1.1: Contexts, challenges and contributions across the sequential decision-making problems addressed in this thesis.

1.4 Outline and Contributions

Despite significant challenges, a company like Huawei therefore has compelling incentives to develop trustworthy sequential decision-making algorithms that meet its specific operational needs and ethical considerations, without sacrificing performance. In this thesis, we address this problem from the perspective of three specific sequential decision-making tasks: best arm identification in multi-armed bandits, deep reinforcement learning, and bundle recommendation. Each of these problems is of practical relevance for Huawei (see examples below) and yet is subject to privacy or fairness concerns. For each problem, we design efficient solutions that align with practical requirements —– taking into account the problem structure and real-world constraints —– while ensuring privacy or fairness without compromising performance. Table 1.1 summarizes the contexts, challenges, and contributions across these three tasks. We perform experiments on toy benchmarks selected to provide fundamental insights and understanding of our methods that can later generalize to real-world applications without compromising Huawei's internal data. By lowering the cost of integrating privacy and fairness and proposing solutions tailored to specific use cases, our aim is thus to encourage the adoption of ethics-aware approaches in sequential decision-making.

The outline of this manuscript is as follows. **Chapter 2** provides general background regarding privacy — with a focus on *differential privacy*, the primary privacy framework employed in this thesis — and fairness. Subsequent chapters address each of the aforementioned problems. **Chapter 3** first tackles best arm identification in multi-agent multi-armed bandits in private settings. **Chapter 4** then addresses the important yet overlooked problem of private deep reinforcement learning. Finally, **Chapter 5** investigates the unexplored topic of fairness in bundle recommendation. In the following, we provide more details on the specific context for each problem as well as the corresponding contributions.

Private Pure Exploration in Multi-Agent Bandits

Best arm identification in multi-agent bandits can model scenarios where the goal is to find, among a set of candidate parameters or items, the solution that maximizes a reward, and where the relevant data is distributed by nature over multiple locations. An example of interest for Huawei is the optimization of large-scale antenna networks. The antennas are parameterized by a large set of parameters that influence network quality and are difficult to tune manually. The multi-agent bandit model allows to automatically find the global parameter maximizing some quality measure (*e.g.*, data throughput) while leveraging the local data of each antenna in the network.

To this end, the different agents have to communicate information based on their local data, for instance through a central server. These communications may carry information about the network state and the parameters that is valuable to Huawei and should not be revealed to competitors. Meanwhile, it can be easy for an adversary to intercept such communications. Designing a learning approach that keeps the communications private without sacrificing the initial performance objective is thus an important topic.

Contributions. In **Chapter 3**, we develop methods to protect communications in multi-agent bandits. We identify two types of information that the network manager may want to conceal: the input data (*e.g.*, network state variables) of the bandit algorithm and the output best arm (*i.e.*, the optimal parameters), calling for two different privacy notions. We show that both methods provide strong theoretical guarantees and efficiently balance sample complexity and privacy guarantees in toy environments.

Corresponding paper:

- RIO, Alexandre, BARLIER, Merwan, COLIN, Igor, and SOARE, Marta. Multi-Agent Best Arm Identification with Private Communications. In: *Proceedings of ICML* (2023).

Privacy-Preserving Deep Reinforcement Learning

RL is being increasingly deployed in the real world for personalized recommendation (e.g., in e-commerce, healthcare, and finance) or the control of autonomous systems. It is also becoming a cornerstone of the post-training of LLMs, at first in LLM alignment with RLHF, and more and more for enhancing the reasoning capabilities of LLMs. All these tasks are data-intensive and typically operate on sensitive data, raising privacy concerns.

Huawei, as a leading global technology company, is particularly interested in developing cutting-edge LLMs fitting their specific use cases, which entails using RL to train LLMs on valuable proprietary data. Recently, the firm released a series of LLMs named *Pangu* [Christianos et al., 2023], whose pipelines heavily rely on RL. Unfortunately, both RL and LLM fine-tuning have been shown vulnerable to privacy attacks. Meanwhile, private RL is a very

challenging topic that remains largely confined to theoretical work, and existing solutions are not suitable for large-scale problems such as the post-training of billion-parameter LLMs. Designing scalable privacy-preserving deep RL algorithms is therefore a pressing problem.

Contributions. **Chapter 4** takes a first step in this direction by developing the first deep RL algorithms with formal DP guarantees. It is organized around two distinct contributions. First, we tackle the offline setting with a model-based approach. Then, we develop a DP policy gradient for the online setting. We lead conclusive experiments on diverse tasks previously unaddressed in the literature, paving the way toward scalable DP RL.

Corresponding papers:

- RIO, Alexandre, BARLIER, Merwan, COLIN, Igor, and THOMAS, Albert. Differentially Private Deep Model-Based Reinforcement Learning. In: *European Workshop on Reinforcement Learning (EWRL)* (2024). Submitted to: *IJCAI* (2025).
- RIO, Alexandre, BARLIER, Merwan, and COLIN, Igor. Differentially Private Policy Gradient. Submitted to: *ICML* (2025).

Producer-Fairness in Bundle Recommendation

Media content platforms and e-commerce websites commonly present products to the user in the form of a set of items, or *bundles*. Recommending bundles instead of single items can enhance user satisfaction and boost the platform and producer revenues, but introduce additional challenges. Indeed, in addition to being relevant to the user, items within a bundle must be highly compatible, diverse or complementary, possibly under cardinality or budget constraints. Considering such pairwise item interactions and structural constraints is difficult from both a formalization and computational perspective.

These challenges make existing fairness-aware approaches for single-item recommendation unsuitable for this setting, hence the need to develop specific fairness-aware bundle recommender systems. As a provider of commercial recommendation solutions, Huawei has a vested interest in addressing the bundle recommendation problem while also engaging with fairness issues for reasons such as legal compliance and long-term customer satisfaction.

Contributions. In **Chapter 5**, we formalize the producer-fair bundle recommendation problem, where users are served in turn with sets of items, and fairness of exposure to producers is achieved over the whole recommendation sequence. We propose both exact and heuristic-based solutions, and a thorough empirical investigation on various datasets shows that our methods effectively balance recommendation quality and producer fairness.

Corresponding paper:

- RIO, Alexandre, SOARE, Marta, AMER-YAHIA Sihem. Producer-Fairness in Sequential Bundle Recommendation. Submitted to: *ECML* (2025).

Chapter Two

Background: Trustworthy Machine Learning

In this chapter, we introduce general concepts and definitions regarding the two aspects of trustworthy machine learning that we address in this thesis, namely **privacy** and **fairness**. The first section focuses on privacy, with an emphasis on *differential privacy* as the main privacy notion upon which the following chapters rely. The second section addresses algorithmic fairness, with a focus on fairness of exposure for recommendation systems. This chapter is not meant to be an exhaustive review of the vast fields of privacy and fairness, but rather provides the background knowledge that is relevant and directly useful for the development of the main contributions of this thesis.

2.1 Differential Privacy

This section first discusses the formalism and main concepts from *differential privacy*, how they are applied to practical settings, and how differential privacy is achieved in practice, from basic mechanisms to (deep) machine learning. It then investigates the limits of the theoretical differential privacy framework when applied to real-world problems, and concludes a brief overview of other privacy notions beyond differential privacy.

2.1.1 Formalism and Main Concepts

Differential privacy (DP), introduced in Dwork [2006], formalizes the protection of private data in algorithms. This framework has become the gold standard in privacy protection thanks to the strong formal guarantees it provides against privacy leakage and its theoretical properties that facilitate the construction of complex private algorithms.

Informally, differential privacy ensures that the presence of a particular data point (for instance, data related to an individual) in the input dataset does not change significantly

the algorithm's output. An adversary observing this output thus cannot determine with high confidence whether a given individual was in the input dataset, thereby protecting the individual's information. Its formal definition relies on a number of notions discussed below.

Definition 2.1.1. *Dataset and Mechanism.* A mechanism $\mathcal{M}$ is a randomized function of the data, that is a function:

$$\mathcal{M} : \mathcal{D} \mapsto \mathcal{O} \ ,$$

that takes as input a dataset $D = \{x_i\}_i \in \mathcal{D}$, with $x_i \in \mathcal{X}$ a data sample, and outputs a mathematical object $o \in \mathcal{O}$ (*e.g.*, a statistics of the data, a probability distribution). In machine learning, $\mathcal{M}$ is generally a training algorithm taking as input the training dataset and outputting a model. $\mathcal{M}$ is random (for instance due to the random sampling from the dataset during training or the random initialization of the model parameters), so that, given $\mathcal{E}$ a subset of $\mathcal{O}$, we consider the probability that $\mathcal{M}$'s output is in $\mathcal{E}$, *i.e.*, $\mathbb{P}(\mathcal{M}(D) \in \mathcal{E})$.

Differentially privacy prevents the mechanism's output from revealing too much information about a particular data point. Therefore, we must clearly define the notion of data point that we want to protect: this is the *privacy unit*. Indeed, a data point could be a single sample or a group of samples, for instance all samples related to a given individual. From the definition of the privacy unit, we can define *dataset adjacency*.

Definition 2.1.2. *Dataset Adjacency.* Two datasets D and D' are *add-or-remove adjacent* (or simply *adjacent* or *neighboring*) if D' can be obtained from D by adding or removing a single data point x. We denote dataset adjacency as $d(D, D') = 1$.

To better understand the implications of differential privacy, it is best to define the *threat model* that describes how an adversary is assumed to gain knowledge about the private data.

Definition 2.1.3. *Threat Model.* In the DP framework, an adversary is able to observe the mechanism's output $\mathcal{M}(D) = o$. From the observation of o, the adversary tries to gain information about a particular data point $x \in D$. The framework also assumes that the adversary may have any auxiliary knowledge about the situation.

We can now provide a formal definition of differential privacy:

Definition 2.1.4. (ϵ, δ)*-differential privacy.* Given $\epsilon > 0$, $\delta \in [0, 1)$, a *mechanism* $\mathcal{M}$ is (ϵ, δ)-differentially-private if for any pair of adjacent datasets D, D' and any subset $\mathcal{E} \subseteq \mathcal{O}$:

$$\mathbb{P}\left(\mathcal{M}(D) \in \mathcal{E}\right) \leq \exp(\epsilon) \cdot \mathbb{P}\left(\mathcal{M}(D') \in \mathcal{E}\right) + \delta \ .$$

ϵ and δ both control the strength of the privacy guarantees but play different roles.

Role of ϵ. To better understand the role of ϵ, sometimes called the *privacy budget*, it can be useful to assume $\delta = 0$ — a case referred to as *pure differential privacy* — and

also assume that $\mathcal{E}$ is reduced to a single output, *i.e.*, $\mathcal{E} = \{o\}$. If $\mathbb{P}(\mathcal{M}(D) = o) > 0$ and $\mathbb{P}(\mathcal{M}(D') = o) > 0$, Definition 2.1.4 is then equivalent to:

$$-\epsilon \leq \log \frac{\mathbb{P}(\mathcal{M}(D) = o)}{\mathbb{P}(\mathcal{M}(D') = o)} \leq \epsilon \ ,$$

as a consequence of the symmetry of the definition with respect to D and D'. Therefore, if $\mathcal{M}$ is $(\epsilon, 0)$-DP (or simply ϵ-DP) with ϵ close to zero, it ensures that $\mathbb{P}(\mathcal{M}(D) = o) \approx \mathbb{P}(\mathcal{M}(D') = o)$. In other words, an adversary observing the output o cannot tell whether the input dataset was D or D'; in particular, they cannot tell if a particular data point was or not in the dataset. When ϵ gets larger, the DP definition allows the two probabilities to diverge to a greater extent, making it easier for the adversary to distinguish between D and D'. Therefore, **the smaller ϵ, the stronger the privacy guarantees, and *vice versa***.

Role of δ. The case $\delta > 0$ is referred to as *approximate differential privacy*. It means that the bound $\mathbb{P}(\mathcal{M}(D) \in \mathcal{E}) \leq \exp(\epsilon) \cdot \mathbb{P}(\mathcal{M}(D') \in \mathcal{E})$ only holds with high probability $1 - \delta$. Therefore, with probability δ, the two probabilities may be arbitrarily different and privacy protection is no longer guaranteed. Approximate DP is therefore a relaxation of DP where we allow the ϵ-guarantee to be broken. In practice, we typically set $\delta < 1/N$, so that the probability of a privacy breach occurring is smaller than the likelihood of picking an individual data point at random.

Remark 2.1.5. *Relaxation of Differential Privacy.* The original definition of differential privacy provides a worst-case bound on the probability ratio $\frac{\mathbb{P}(\mathcal{M}(D)\in\mathcal{S})}{\mathbb{P}(\mathcal{M}(D')\in\mathcal{S})}$, holding for any measurable $\mathcal{E} \subseteq \mathcal{O}$. Several other definitions have been proposed that relax this definition. In particular, Rényi Differential Privacy (RDP) [Mironov, 2017] rather bounds $R_\alpha(\mathcal{M}(D), \mathcal{M}(D')) \leq \epsilon$, where R_α is the Rényi divergence of order α, which corresponds to bounding the moments of the privacy loss function $\log \frac{\mathbb{P}(\mathcal{M}(D)\in\mathcal{S})}{\mathbb{P}(\mathcal{M}(D')\in\mathcal{S})}$. As a consequence, using RDP instead of DP may lead to tighter (and therefore more advantageous) bounds on the privacy budget ϵ.

Remark 2.1.6. *Central* vs. *Local DP.* Another key aspect of the DP threat model concerns the location at which the DP mechanism is applied. In the central DP model, a trusted central server can access the raw data and process it. In the local DP model, a DP mechanism is applied to the data before the server accesses it, so that the server never sees raw data. While local DP can accommodate situations where the central server is not trusted, central DP generally requires much less noise and thus yields better privacy-performance trade-offs. **In this thesis, we focus on the central setting.**

2.1.1.1 Applications of the DP Framework

A major focus of this thesis is to provide privacy-preserving mechanisms suitable to practical sequential decision-making applications. This first requires properly defining the various

	Input data *What do we protect?*	**Privacy unit** *How do we define adjacency? ("granularity" of protection)*	**Mechanism** *What do we make DP?*	**Output** *What does the adversary observe and try gaining information from?*
Recommendation	User interaction history, *e.g.*, purchases, ratings, views	User data*	Training algorithm	Recommendation system or recommendations
Reinforcement Learning	States, actions, rewards	Trajectories or user data*	Training algorithm	Policy or actions
LLM	Training corpus, *e.g.*, prompt-completion pairs	Single examples or user data*	Training algorithm	Language model or completions

Table 2.1: Application of the DP framework for different domains. *: *user data refers to all examples related to a given user or individual.*

concepts introduced in the previous section, aligning with the practical requirements and threat model. Table 2.1 provides several examples illustrating the application of the differential privacy framework on different tasks. This table highlights the questions one has to ponder when defining differential privacy in a specific use case.

This process has crucial implications for the practical protection that the private algorithm will provide. Consider the case of training a private large language model (LLM) on medical data. If we define the privacy unit at the level of a single example —– meaning that the presence or absence of a single prompt-completion pair in the training data has a limited effect on the final model — then each example is protected in isolation. However, a single patient may be associated with multiple training examples. Since the DP guarantee applies only at the single-example level, it does not provide guarantees about the effect of including or excluding all of a patient's data collectively. This means that while each prompt-completion pair is individually protected, the aggregate influence of all examples related to a patient could still be significant, potentially leading to privacy leakage about that patient in the trained model.

When addressing privacy issues in multi-agent bandits (Chapter 3) and deep reinforcement learning (Chapter 4), we therefore carefully consider the different aspects of the DP framework to align with the practical privacy vulnerabilities.

2.1.2 Achieving Differential Privacy

Having introduced the framework and main concepts of differential privacy, we now address the topic of how to achieve differential privacy in practice. We first review the basic DP mechanisms and the properties that are useful to build more complex privacy-preserving algorithms, before delving into the training of privacy-preserving machine learning models.

2.1.2.1 Basic Mechanisms

Consider a (non-private) function of the data, $f : \mathcal{D} \mapsto \mathcal{O}$ and $\|\cdot\|$ a norm on $\mathcal{O}$. When adding or removing a data point x from D, resulting in an adjacent dataset D', the output of f changes by at most (assuming the corresponding quantity is bounded):

$$\Delta_{\|\cdot\|}(f) := \max_{d(D,D')=1} \|f(D) - f(D')\| \ .$$

$\Delta_{\|\cdot\|}(f)$ is the *sensitivity* of f with respect to $\|\cdot\|$. In particular, when $\mathcal{O} \subseteq \mathbb{R}^d$, we denote $\Delta_1(f)$ and $\Delta_2(f)$ the sensitivity of f with respect to the L_1- and L_2-norms, respectively. This quantity reflects how f's output is affected by adding or removing a data point in D.

In essence, achieving differential privacy requires obscuring the contribution of a given data point to the mechanism's output. A natural approach is thus to perturb f's output with a noise proportional to $\Delta_{\|\cdot\|}(f)$. The magnitude of the perturbation should also reflect the strength of the privacy guarantees, scaling with $1/\epsilon$. Therefore, the standard approach to transform f into a differentially private mechanism $\mathcal{M}$ is to add a zero-mean random noise z to f's output whose magnitude σ scales with $\Delta_{\|\cdot\|}(f)/\epsilon$:

$$\mathcal{M}(D) := f(D) + z \ .$$

We now provide two classical DP mechanisms for real-valued functions $f : \mathcal{D} \mapsto \mathbb{R}^d$.

Definition 2.1.7. *Laplace mechanism.* Given $\epsilon > 0$, the Laplace mechanism adds a noise sampled from the centered Laplace distribution with scale $\Delta_1(f)/\epsilon$:

$$\mathcal{M}_{\text{Lap}}(D) := f(D) + \text{Lap}(0, \Delta_1(f)/\epsilon) \ .$$

The resulting mechanism $\mathcal{M}$ is ϵ-DP.

Definition 2.1.8. *Gaussian mechanism.* Given $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$, the Gaussian mechanism adds a noise sampled from the centered normal distribution with scale $\sigma = C(\delta) \cdot \Delta_2(f)/\epsilon$:

$$\mathcal{M}_{\text{Gauss}}(D) := f(D) + \mathcal{N}(0, \sigma^2) \ .$$

If $C(\delta) \geq \sqrt{2\log(1.25/\delta)}$, then the resulting mechanism is (ϵ, δ)-DP.

Both mechanisms are widely used and are the building blocks of many complex privacy-preserving algorithms. However, they have notable differences. First, the Laplace mechanism is based on L_1-sensitivity, whereas the Gaussian mechanism relies on L_2-sensitivity. This distinction makes the Gaussian mechanism generally more suitable for *high-dimensional settings*. The reason is that the L_2-norm grows sub-linearly with the number of dimensions, meaning the added noise increases more slowly compared to the L_1-norm, which scales linearly with dimension. As a result, in high-dimensional settings, the Gaussian mechanism

generally provides better performance while still ensuring privacy. On the other hand, the Gaussian mechanism only guarantees approximate differential privacy, unlike the Laplace mechanism. Finally, the Gaussian mechanism in theory only works for $\epsilon < 1$, although alternative Gaussian mechanisms have been studied for $\epsilon \geq 1$ (see Zhao et al. [2019]).

2.1.2.2 Toy Example: Membership Inference Attack

Examples 6 and 7 illustrate how such mechanisms can protect the leakage of private information. We consider the case of a membership inference attack, in which an adversary tries to infer the presence of an individual in a dataset. The mere knowledge that an individual is part of a dataset can itself be sensitive information; for example, if the dataset consists only of members of a certain political party or individuals with a particular disease.

Example 6. *Membership inference attack without DP.* Consider an anonymized dataset D containing information about $n = 100$ individuals from a political party. The dataset includes in particular monthly revenues and basic demographic information (*e.g.*, age, gender, zip code). The revenues range from €1,000 to €10,000. The dataset owner allows users to query information about the dataset in the form of aggregate statistics, using a public API. **Alex intends to use the API to discover whether or not Thomas is a member of the political party**. He remembers that Thomas disclosed that he earned €10,000 per month. He first queries the aggregated monthly revenue over the whole dataset D and obtains $f(D) = 300{,}000$. Using demographic information he has about Thomas, he now queries the aggregated monthly revenue on a data subset D' that he expects would exclude Thomas, and obtains $f(D') = 290{,}000$ with a count of $n' = 99$ individuals. He therefore knows that the left out individual earns $f(D) - f(D') = 10{,}000$ euros in monthly revenues. Combining all the information he has about Thomas, Alex concludes that it is likely that Thomas is in this dataset and is a member of the political party.

Example 7. *Membership inference attack with DP.* Now, consider that when querying the aggregated revenue, the API applies the Laplace mechanism with $\epsilon = 0.5$ to make the statistic differentially private. Because the revenue range is known by the dataset owner, they can compute f's sensitivity: $\Delta(f) = 9{,}000$. Alex provides the same queries as in Example 6, but now obtains $f(D) = 309{,}792$ and $f(D') = 288{,}754$. Computing $f(D) - f(D') = 21{,}038$, Alex has not gained much knowledge about Thomas' membership in the political party.

In Example 7, the Laplace mechanism effectively obscures Thomas' contribution to the dataset, preventing a successful membership inference attack. However, each query remains sufficiently close to the exact data to provide meaningful statistical insights.

2.1.2.3 Composition and Post-Processing

The above mechanisms are insufficient on their own for building complex private algorithms beyond simple statistical queries. Fortunately, the properties of differential privacy allow us

to efficiently compute the total privacy budget of the composition of multiple queries.

Parallel composition states that applying a DP mechanism multiple times on disjoint subsets of the data does not degrade the privacy guarantees. On the other hand, *sequential composition* properties state that if we repeatedly apply DP mechanisms on the same data, then the privacy budget accumulates.

Proposition 2.1.9. Parallel composition. *Let $\mathcal{M}$ be an (ϵ, δ)-DP mechanism and $D = \bigcup_{k=1}^{K} D_k$ a partition of the data. Then, the mechanism $(\mathcal{M}(D_1), ..., \mathcal{M}(D_K))$ which releases all outputs is (ϵ, δ)-DP.*

Proposition 2.1.10. Sequential composition. *Let $\mathcal{M}_1$ be an (ϵ_1, δ_1)-DP mechanism and $\mathcal{M}_2$ be an (ϵ_2, δ_2)-DP mechanism. Then the mechanism $(\mathcal{M}_1(D), \mathcal{M}_2(D))$ which releases both outputs is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-DP.*

Example 8. *Toy example from Section 2.1.2.1.* The sequential composition property reflects the fact that if Alex would perform the same queries multiple times, he would obtain a better estimate of the exact value of the aggregated revenue as the random noise averages out. This corresponds to a degradation of the privacy guarantees, *i.e.*, an increase in the total privacy budget ϵ. The API may prevent such a strategy by imposing an upper limit on the privacy budget that Alex can consume, each query consuming a part of this budget.

We now consider the more general case of k-fold adaptive composition, a type of sequential composition $\mathcal{M}_1 \circ ... \circ \mathcal{M}_k$ such that:

- Mechanism $\mathcal{M}_i$ may depend on the outputs of previous mechanisms $\mathcal{M}_1, ..., \mathcal{M}_{i-1}$;
- Mechanism $\mathcal{M}_i$ takes as input both the private dataset D and the outputs of previous mechanisms $\mathcal{M}_1, ..., \mathcal{M}_{i-1}$.

Proposition 2.1.11. Sequential composition. *Let $\mathcal{M}_1 \circ ... \circ \mathcal{M}_k$ be a k-fold composition of mechanisms, such that $\mathcal{M}_i$ is (ϵ_i, δ_i)-DP. Then the mechanism that releases all outputs of the k-fold composition is $(\sum_{i=1}^{k} \epsilon_i, \sum_{i=1}^{k} \delta_i)$-DP.*

While the bound on the privacy budget provided in Proposition 2.1.11 degrades linearly with the number of compositions, Proposition 2.1.12 below provides a tighter bound on ϵ under k-fold composition. The trade-off is a slightly larger δ. Even tighter bounds can be obtained using relaxed DP definitions like RDP (see Section 2.1.2.4)

Proposition 2.1.12. Advanced k-fold composition [Dwork et al., 2010]. *Let $\mathcal{M}_1 \circ ... \circ \mathcal{M}_k$ be a k-fold composition of mechanisms, such that each $\mathcal{M}_i$ is (ϵ, δ)-DP. Then, for any $\delta' > 0$, the mechanism that releases all outputs of the k-fold composition is $(\epsilon', k\delta + \delta')$-DP, with:*

$$\epsilon' = k\epsilon(e^{\epsilon} - 1) + \epsilon\sqrt{2k\log(1/\delta')} \ .$$

Finally, the *post-processing* property of DP states that applying a data-independent transformation to the output of a private mechanism does not affect the privacy budget. In particular, the privacy guarantees cannot be reversed by further processing the output.

Proposition 2.1.13. Post-processing. *Let $\mathcal{M}$ be an (ϵ, δ)-DP mechanism. For any data-independent function g, $g(\mathcal{M}(D))$ is (ϵ, δ)-DP.*

Together, these properties enable the design of complex private algorithms from simple building blocks such as the Laplace and Gaussian mechanisms, without the need to refer back to the core definition of differential privacy.

2.1.2.4 DP Machine Learning

One of the most important applications of the above principles is the training of differentially private machine learning models. A machine learning model can generally be represented as a vector $\theta \in \mathbb{R}^d$, *e.g.*, the parameters of a neural network, such that for an input example $x \in \mathcal{X}$, the model return $y = f_\theta(x)$. Let $\mathcal{M} : \mathcal{D} \mapsto \mathbb{R}^d$ be the mechanism representing the algorithm with which we train model θ, *e.g.*, the Stochastic Gradient Descent (SGD). $\mathcal{M}$ takes as input a dataset $D \in \mathcal{D}$ and releases the model $\theta \in \mathbf{R}^d$. We assume that a malicious adversary $\mathcal{A}$ can access the model, or, at least, is able for any input example x to obtain the corresponding model prediction $f_\theta(x)$ (*e.g.*, through a public API). Based on this, the adversary tries to gain information about individual examples in the training data D.

Membership inference attacks against ML models. Many studies have shown that such privacy attacks against machine learning models are feasible. In particular, many membership inference attacks have been proposed: given a target sample x_{target}, the adversary tries to infer the membership of x_{target} to D. Formally, $\mathcal{A}$ tries to estimate the probability $\mathbb{P}(x_{\text{target}} \in D)$ and classifies x_{target} as member of D if $\mathbb{P}(x_{\text{target}} \in D) \geq \tau$, where τ is a threshold balancing the precision and the recall of the attack. A membership inference attack can therefore be formalized as a binary classification task. A typical approach for $\mathcal{A}$ is thus to train a classifier $\mathcal{A}_\theta : \mathcal{X} \mapsto \{0, 1\}$ which predicts the binary label of x:

$$y = \begin{cases} 1, & \text{if } x \in D \\ 0, & \text{otherwise.} \end{cases} ,$$

with:

$$\mathbb{1}(\hat{y} \geq \tau) \ , \ \hat{y} = \widehat{\mathbb{P}}(x \in D) = \mathcal{A}_\theta(x) \ .$$

A typical approach consists in using *shadow models* [Shokri et al., 2017] —*i.e.*, models that are similar to the target model θ but trained on different data — to collect a dataset of example-membership pairs $D_{\text{attack}} = (x_j, y_j)_j$. Then, a classifier $\mathcal{A}_\theta$ is trained from D_{attack} in a supervised fashion.

Differentially Private Deep Learning. [Abadi et al., 2016] introduce DP-SGD, a differentially private training method for deep neural networks based on SGD. The underlying

idea is that, when training a neural network, the output model θ is characterized by the sample gradients computed during training. Such gradients are computed from the private examples in the dataset, and it is both necessary and sufficient to make them differentially private in order to obtain a final model that is itself differentially private. DP-SGD, described in Algorithm 1, therefore applies the Gaussian mechanism to the gradients before updating the model. Formally, let:

$$G_t : D \longrightarrow \bar{g}_t := \sum_{i \in L_t} \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$$

be the function that computes the stochastic gradient at training round t, and further denote $g_t(x_i) := \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$ the per-sample gradients. This is the function on which we apply the Gaussian mechanism. Since the gradients of neural networks are typically unbounded, preventing the computation of the sensitivity $\Delta_2(G_t)$, DP-SGD scales the gradients so that their norm does not exceed some constant $S \in \mathbb{R}$ (a stage usually referred to as *clipping*). Precisely, aligned with the DP definition, DP-SGD clips the per-sample gradients $g_t(x_i)$ for $x_i \in \mathcal{X}$ before aggregation (line 10), in order to appropriately obscure the contribution of each sample to the gradient. In the *add-or-remove* setting, the sensitivity of G_t is therefore bounded by S. S is typically treated as a hyperparameter.

Algorithm 1 DP-SGD [Abadi et al., 2016]

Require: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N} \sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise multiplier z, gradient norm bound S.

1: **Initialize** θ_0 randomly
2: **for** $t \in [T]$ **do**
3: Take a random sample L_t with sampling probability $\frac{L}{N}$
4: **Compute gradient**
5: **for** each $i \in L_t$ **do**
6: $g_t(x_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, x_i)$
7: **end for**
8: **Clip gradient**
9: **for** each $i \in L_t$ **do**
10: $\bar{g}_t(x_i) \leftarrow g_t(x_i) / \max\left(1, \frac{\|g_t(x_i)\|_2}{S}\right)$
11: **end for**
12: **Add noise**
13: $\tilde{g}_t \leftarrow \frac{1}{|L_t|} \left(\sum_i \bar{g}_t(x_i)\right) + \mathcal{N}(0, z^2 S^2 \mathbf{I})$
14: **Descent**
15: $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{g}_t$
16: **end for**
17: **Output** θ_T

A given sample may appear in different gradients throughout the model training, scaling the corresponding privacy budget under *sequential composition*. The number of gradients in which each sample x_i appears is random as x_i is sampled with probability $q = L/N$ in the batch L_t at each round — that is *Poisson sampling*. To compute the total privacy budget ϵ_{tot} consumed by Algorithm 1, Abadi et al. [2016] develop the *moments accountant* that computes

ϵ_{tot} from the hyperparameters q, T and z. The *moments accountant* leverages Rényi-DP and *privacy amplification by subsampling* [Balle et al., 2018] to provide a tighter bound on ϵ_{tot} (Proposition 2.1.14) than provided by the basic or advanced composition theorems.

Proposition 2.1.14. Privacy budget of DP-SGD (Moments accountant, [Abadi et al., 2016]). *Given $\delta \in (0,1)$, $q \in (0,1)$, $T \in \mathbb{N}^\star$ and $z > 0$, DP-SGD is (ϵ_{tot}, δ)-DP with:*

$$\epsilon_{tot} = \mathcal{O}\left(\frac{q\sqrt{T}}{z}\right) \quad .$$

2.1.3 The Limitations of the DP Framework

While appealing for its strong formal privacy guarantees, the differential privacy framework can be considered too restrictive for some practical applications. Ponomareva et al. [2023] extensively discuss the practicalities of deploying differential privacy in deep learning applications, and highlight what may be one of the main obstacles to its widespread adoption in industrial contexts: *the gap between a strong theoretical guarantee and a realistic privacy budget to target in practice.*

In Definition 2.1.4, for a given output $o \in \mathcal{O}$, the ratio $\frac{\mathbb{P}(\mathcal{M}(D)=o)}{\mathbb{P}(\mathcal{M}(D')=o)}$ is bounded by $\exp(\epsilon)$, which means that an ϵ-DP mechanism "allows" the output o to be $\exp(\epsilon)$ more likely for D' than for an adjacent dataset D. Consider for instance $\epsilon = 0.1$. $\exp(0.1) \approx 1.11$; therefore, o can only be up to 11% more likely for D' as compared to D, which clearly limits the ability of an adversary to distinguish between both, making $\epsilon = 0.1$ a strong theoretical guarantee. On the other hand, consider $\epsilon = 5$: $\exp(5) \approx 148$, which means that under 5-DP, using D' instead of D may make o up to 148 times more likely. Obviously, such ϵ does not provide strong privacy guarantees in theory. In fact, from a theoretical perspective, only $\epsilon < 1$ is deemed to provide significant privacy protection. **However, as pointed out in Ponomareva et al. [2023], such ϵ generally leads to exceedingly large performance drops in practical deep learning applications.**

The privacy guarantee ϵ in Definition 2.1.4 is only an upper bound on the privacy leakage, and may be very loose. In practice, the real privacy risk may therefore be significantly lower than the bound ϵ we are able to compute. In particular, composition theorems, which are often used in the computations of the privacy guarantees of a complex algorithm, can yield very loose bounds; although bounds computed using RDP (*e.g.*, using DP-SGD's accountant) are much tighter. Moreover, differential privacy is a worst-case definition in several aspects. Notably, it must hold for every pair of neighboring datasets and does not account for the actual dataset on which we apply the algorithm. Therefore, the sensitivity must be computed with respect to the worst-case dataset pairs, the one for which the function value changes the most, which can result in very large noise amounts. Moreover, differential privacy is designed to resist strong adversaries with significant background knowledge about the dataset and algorithm. For instance, even if the adversary has complete knowledge of the dataset except one individual, this individual's membership must still be protected. More

specifically, for iterative algorithms, it is assumed that all intermediate updates (*e.g.*, all gradients for DP-SGD) are released and accessed by the adversary.

Such assumptions may be unrealistic for a practical adversary. For instance, Feldman et al. [2018] show that, in convex problems, the privacy guarantees of an iterative training algorithm can be amplified if the intermediate updates are actually not released in practice. However, Cebere et al. [2025] demonstrate that it is not always true in non-convex settings. In sequential decision-making tasks — the focus of this thesis — where sensitive input data often appears as long, high-dimensional sequences, it may also be impossible for the adversary to identify precisely an entire target sequence for which they want to infer membership. Therefore, in practice, the adversary faces the very complex task of reconstructing a high-dimensional input with limited background knowledge.

For these reasons, Ponomareva et al. [2023] argue that even if it does not provide formal privacy guarantees, a privacy budget $1 \leq \epsilon \leq 10$ is a realistic and reasonable goal for achieving meaningful privacy guarantees in deep learning tasks. Also, any finite ϵ is an improvement over a model with no formal privacy guarantees and can already significantly reduce the chances of successful attacks. However, the last case should be supported at least by empirical privacy auditing — where the algorithm is assessed against privacy attacks — before deploying the model. Such privacy attacks can nevertheless be difficult to implement and rigorous benchmarks for privacy auditing are scarce (we can cite, for instance, Jagielski et al. [2020]), suggesting that there should be significant efforts in this direction if we want to accelerate the deployment of private models in the real world.

2.1.4 Beyond Differential Privacy

If its strong formal guarantees and desirable properties have made differential privacy the main focus of research in private machine learning, it should nevertheless be regarded as just one lever among several tools for protecting private data. We discuss below other privacy tools that could be used instead or in conjunction with differential privacy depending on practical requirements.

Cryptography. Cryptography refers to the ensemble of techniques that process information to make it undecipherable for external parties. When the data is encrypted, an adversary cannot extract any information from this data in a reasonable time frame without access to the decryption key(s), offering very strong protection. Examples of cryptographic techniques include *homomorphic encryption*, which allows computations directly on the encrypted data without decryption, and *secure multi-party computation*, a framework under which multiple parties can collaborate and perform joint computations without ever exposing raw data. However, cryptographic techniques are often computationally intensive and may pose challenges when applied to complex computations, limiting their applications.

Federated Learning. Federated learning [McMahan et al., 2017] has long been praised as a privacy protection tool. Indeed, federated learning enables the training of global models in distributed settings while the raw data never leaves the remote clients (*e.g.*, mobile devices). This intuitively reduces the likelihood of catastrophic privacy leakage as data is not stored in a single location and is never communicated in raw form. However, federated learning still requires to share local model updates between the different computation nodes. As the extensive literature on differential privacy has shown, model updates, *e.g.*, gradients, can leak considerable information about the input data. Therefore federated learning alone is not sufficient to protect data, but may rather be used in conjunction with differential privacy or cryptography to tackle distributed settings.

***Ad hoc* privacy notions.** Existing privacy notions may not fit every practical use case, and more specific definitions may be more suitable. Differential privacy, for instance, is regularly revisited to fit specific threat models, often relaxing its worst-case requirements [Desfontaines and Pejó, 2020, Cyffers and Bellet, 2022]. In this thesis, we address the problem of *best arm privacy* in multi-agent bandits, which is motivated by real-world applications and calls for another privacy definition, all detailed in the corresponding chapter (**Chapter 3**).

2.2 Algorithmic Fairness

Decision-making tasks are particularly concerned with fairness issues as the corresponding decisions may significantly impact real people. In this section, we review different notions of algorithmic fairness, with an emphasis on automatic decision-making. It is impossible to provide a single, unifying fairness framework since fairness issues are highly dependent on the specific practical context. Therefore, this section is not meant to be an exhaustive review of all fairness definitions in the field, but rather to introduce the main concepts relevant to the treatment of fairness in this thesis.

2.2.1 Formalism and Main Concepts

In this section, we provide the main concepts and definitions used to formalize fairness in algorithmic decision-making, starting from broad legal definitions and progressing to more specific notions directly relevant to the context of fair sequential decision-making. We use the term **individual** to refer to a subject of the algorithm decision, *i.e.*, someone that is affected by this decision. This individual has **attributes**, such as their gender, age, or economic status. We qualify an attribute as **protected** if this attribute should not be used in the algorithm decisions on the basis of legal regulations or societal expectations, which is typically the case for the aforementioned attributes.

2.2.1.1 Legal Definitions

In the legal domain, unfairness in algorithms is usually defined according to two notions [Barocas and Selbst, 2016]:

- **Disparate treatment:** an algorithm is said to have *disparate treatment* if it explicitly treats an individual differently based on the individual's membership in a protected group (*i.e.*, a class defined with respect to protected attributes);
- **Disparate impact:** an algorithm is said to have *disparate impact* if it affects a member of a protected group negatively — even if the algorithm does not provide disparate treatment.

Disparate treatment is sometimes referred to as *direct discrimination* and disparate impact as *indirect discrimination*. Example 9 illustrates the difference between both notions.

Example 9. *Algorithmic-aided hiring.* Consider a company seeking to fill a vacant job position. After receiving hundreds of resumes, it conducted a screening stage to select $N = 20$ potential candidates to interview. Due to limited resources, the company uses an algorithm to select a few candidates to interview from the remaining 20 candidates. **Disparate treatment:** The algorithm directly considers the candidate's gender, a protected attribute, to base its decision — for instance, a linear regression taking as input the gender attribute and applying a non-zero coefficient to its value. **Disparate impact:** The algorithm does not take gender as input as a means to avoid gender bias. However, it takes other variables that are correlated to gender, and is trained on historical data where men have been unfairly advantaged. Thus, the algorithm is likely to indirectly favor men over women.

While *disparate treatment* is more easily addressed through legal regulations that prohibit the direct use of protected attributes in algorithmic decision-making, *disparate impact* is more challenging to identify and regulate.

2.2.1.2 Individual and Group Fairness

From these legal definitions, different definitions have been proposed to measure disparate treatment and disparate impact in algorithmic decisions, mainly from two perspectives:

- **Individual fairness**, which states that similar individuals with respect to a task (that is, excluding irrelevant protected attributes) should be treated similarly;
- **Group fairness**, which states that protected groups (defined based on protected attributes) should receive similar outcomes from the algorithm.

The key difference is that *group fairness* treats fairness at the aggregated level and focuses on protected attributes to the detriment of other — possibly relevant — individual features, while *individual fairness*, which is often based on a distance metric between individuals, takes into account other individual features beyond protected attributes. Group and individual

fairness may conflict: for instance, if an algorithm enforces strict parity between men and women in the number of candidates recommended for interview (*group fairness*), it may overlook significant differences in relevant attributes (*e.g.*, years of experience) between the two groups (*individual unfairness*). In this thesis, we focus on **group fairness**.

2.2.1.3 Outcome Fairness

As mentioned before, the definition of fairness is highly dependent on the task hand. Fairness is often defined with respect to outcome probabilities. Below we briefly review typical outcome-based fairness measures as described in Pessach and Shmueli [2020].

Given a training dataset D, auxiliary information A and a individual x, we consider an algorithm f making binary predictions or decisions $\hat{y} = f(x; D, A)$; for instance, an algorithm predicting whether or not a candidate should be interviewed, or whether or not a product should be recommended. In the context of group fairness and based on a protected attribute, we denote $S = 1$ the privileged group and $S = 0$ the unprivileged group. For each measure $\mathbf{M}$ provided below, a small value is associated with more fair decisions. A corresponding fairness constraint could be to ensure that $\mathbf{M} \leq \epsilon$ for a small ϵ.

Definition 2.2.1. *Demographic parity.* It ensures that a positive decision is attributed with the same probability to both groups, and is measured as:

$$\mathbf{M} = |\mathbb{P}(\hat{y} = 1|S = 1) - \mathbb{P}(\hat{y} = 1|S = 0)| \ .$$

Definition 2.2.2. *Disparate treatment.* It ensures that the ratio of positive decision probabilities between the unprivileged and privileged groups is close to 1, and is measured as:

$$\mathbf{M} = 1 - \frac{\mathbb{P}(\hat{y} = 1|S = 0)}{\mathbb{P}(\hat{y} = 1|S = 1)} \ .$$

Definition 2.2.3. *Equalized odds* and *equal opportunity.* These definitions assume the existence of a ground truth decision y and ensure that the true positive rate (TPR) and the false positive rates (FPR) are close between the two groups. *Equal opportunity* only considers the FPR, while *equalized odds* consider both the FPR and the TPR.

$$\mathbf{M}_{-} = |\mathbb{P}(\hat{y} = 1|S = 1, y = 0) - \mathbb{P}(\hat{y} = 1|S = 0, y = 0)| \ ,$$
$$\mathbf{M}_{+} = |\mathbb{P}(\hat{y} = 1|S = 1, y = 1) - \mathbb{P}(\hat{y} = 1|S = 0, y = 1)| \ .$$

In Example 9, we could ensure outcome-based fairness using *demographic parity* by enforcing the probability of recommending people of both genders to be equal up to ϵ. Example 10 provides another illustration of outcome-based fairness.

Example 10. *Next-movie recommendation in streaming platforms.* After a user finishes a movie, the platform uses a recommendation system to recommended the next movie that

the user should watch, based on past interactions. The algorithm's objective is to select the movie that the user is the most likely to watch, or has the best predicted rating. Due to a *popularity bias*, the algorithm tends to over-recommend the most popular movies (group $S = 1$), while the least popular options (group $S = 0$) are disadvantaged. A fairness-aware mechanism based on fair outcomes will ensure that the probability of recommending a movie from group $S = 0$ is close to the probability of recommending a movie from group $S = 1$.

The above definitions are widely used in the literature to measure and encourage group-fair algorithmic decisions, focusing on the final outcome. However, complex decision-making tasks involve several stages that may impact the fairness of the final decision earlier in the process. Other fairness notions, like *fairness of exposure* (discussed below), are therefore the topic of extensive research.

2.2.1.4 Fairness of Exposure

While outcome-based fairness focuses on the outcomes of a decision, exposure-based fairness seeks to ensure that individuals from different groups have equal access to opportunities within the decision process. Fairness of exposure therefore intervenes at an earlier stage of the decision process compared to outcome-based fairness. Examples 11 and 12 provide examples illustrating the difference between outcome-based fairness and fairness of exposure.

Example 11. *Fairness of exposure in hiring.* Consider again the problem of selecting applicants to interview described in Example 9. This problem can be framed as a two-stage decision process, where **1)** N applicants are screened from the initial pool of resumes (*screening stage*), and **2)** $n < N$ applicants are selected for interview from the screened applicants (*selection stage*). The fact that applicant i is selected for interview is the outcome of this process. If an algorithm directly selects the applicants to interview as in Example 9 (second stage), and we want to ensure that this algorithm provides fair decisions, then it is *outcome-based* fairness. On the other hand, consider an algorithm that intervenes in the first stage of the process, that is resume screening, proposing a list of N potential candidates to consider for an interview, but we let the hiring manager select the interviewed candidates. In this case, a suitable notion of algorithmic fairness is to ensure that the algorithm fairly balances the different protected groups it will show (or *expose*) to the hiring manager, *i.e.*, it provides the same opportunities to the different protected groups. This is *fairness of exposure*.

Example 12. *Fairness of exposure in e-commerce.* Consider an e-commerce platform with different producers $k = 1, ..., K$. Each producer k proposes an ensemble of items G_k. When a user visits the platform, a recommendation algorithm chooses different items to expose to the user. Recommendations are based on past interactions, with the aim to maximize the predicted relevance. However, items from producer $k = 1$ (*i.e.*, items from G_1) have historically been more popular. As a result, the algorithm tends to assign them higher relevance scores, and items from this producer receive disproportionate exposure. To be fair to the producers, the platform could ensure that all groups of items receive a fair exposure.

These examples show that, while fairness of exposure seeks to provide fair opportunities to the different groups, the outcome — the candidate selected for an interview in Example 11, the item purchased or viewed by the user in Example 12 — no longer depends on the algorithm, and final outcomes may not be fair in the sense of the definitions from Section 2.2.1.3.

Fairness of exposure is often associated with ranked lists in recommendation. When a platform recommends a top-n list of items, that is an ordered list $(i_1, ..., i_n)$ containing the n most relevant items for the current recommendation, it is assumed that the position of the item in the list influences its probability of being considered (*e.g.*, for being selected as a job applicant, for being watched as a movie, for being purchased a product). This is called the *position bias*, and we refer to how much the item is given visibility as the *item exposure*. The position bias is generally quantified using the Discounted Cumulative Gain (DCG) measure: the item of rank j in a top-n list receives an exposure equal to $1/\log(1+j)$. This models the fact that the item of rank j will be more visible than the item of rank $k > j$ and that first-ranked items receive disproportionally more attention than items of lower ranks (*winner-takes-all*).

However, fairness of exposure may be considered outside of ranked lists. For instance, when presenting items in the form of bundles where items are not competing but rather complementary — like a set of compatible and complementary accessories that could be bought together — it is reasonable to assume that all items in the bundle receive equal exposure.

From the perspective of group fairness, fairness of exposure ensures that the aggregated exposure of items from different groups $G_1, ..., G_K$ is fair concerning some criterion. As for outcome-based fairness, the groups $G_1, ..., G_K$ are defined based on protected attributes. Below, we define group exposure in the case of ranked lists and (un-ranked) sets of items.

Definition 2.2.4. *Group exposure of ranked lists.* Consider a ranked list of items $L = (i_1, ..., i_n)$. We denote:

$$E_k = \frac{1}{|G_k|} \cdot \sum_{j=1}^{n} \mathbb{1}(i_j \in G_k) \cdot v_j \ .$$

the exposure of group $k = 1, ..., K$ in list L, where v_j is the position bias of rank j.

Definition 2.2.5. *Group exposure of sets.* Consider a set of items $S = (i_1, ..., i_n)$ with no position bias (*e.g.*, a bundle). We denote:

$$E_k = \frac{1}{n} \cdot \sum_{j=1}^{n} \mathbb{1}(i_j \in G_k) \ .$$

the exposure of group $k = 1, ..., K$ in list S.

We can recover Definition 2.2.5 from Definition 2.2.4 by setting $v_j = |G_k|/n$ for all $j = 1, ..., n$ and $k = 1, ..., K$.

From the above definitions, we now provide two popular examples of definitions of fairness of exposure, based on the broader definitions of demographic parity and disparate treatment, as stated in Singh and Joachims [2018].

Definition 2.2.6. *Demographic parity of exposure.* It ensures that all groups receive equal exposure:

$$E_k = E_\ell \ , \quad \forall k, \ell = 1, ..., K \ .$$

Definition 2.2.7. *Disparate treatment in exposure.* It ensures that the exposure of each group is proportional to its average relevance:

$$\frac{E_k}{U_k} = \frac{E_\ell}{U_\ell} \ , \quad \forall k, \ell = 1, ..., K \ ,$$

where $U_k = |G_k|^{-1} \cdot \sum_{j=1}^{n} \mathbb{1}(i_j \in G_k) \cdot r_j$ is the average relevance of group G_k.

In light of these two definitions, we can provide a more general definition of fairness of exposure: given a group weight ρ_k, we seek to ensure that the exposure of each group is proportional to its weight:

$$\frac{E_k}{\rho_k} = \frac{E_\ell}{\rho_\ell} \ , \quad \forall k, \ell = 1, ..., K \ .$$

We recover Definitions 2.2.6 and 2.2.7 using, for every k, $\rho_k = 1$ and $\rho_k = U_k$ respectively. However, this definition may encompass other notions of fairness of exposure.

These definitions enforce equality between all the (weighted) exposures, which is generally impossible to achieve. In practice, we aim to achieve these definitions up to some tolerance. In the case of two groups G_0, G_1, we can for instance measure unfairness as follows:

$$\mathbf{M} = \left| 1 - \frac{E_0/\rho_0}{E_1/\rho_1} \right| \ .$$

In case of perfect fairness, $\mathbf{M} = 0$, but $\mathbf{M} \leq \epsilon$ may be tolerated. If we have more than two groups, then any deviation measure (*e.g.*, standard deviation) on the series $(E_1/\rho_1, ..., E_k/\rho_k)$ may be considered to measure fairness.

2.2.1.5 Amortized Fairness over Sequences

So far, we considered fairness of exposure only from the perspective of a single ranked list or set of items. However, enforcing fairness constraints on a limited number of items is challenging, if not infeasible. Indeed, allocation opportunities are limited, meaning only a few slots are available to distribute exposure fairly among different groups. If one really wants to enforce strict fairness constraints in a small number of items, it may therefore be at

the expense of highly relevant items, possibly leading to an unreasonable drop in the overall quality of the recommendation.

To circumvent this issue, Biega et al. [2018] introduced the notion of *amortized fairness* in the context of individual fairness in rankings. The idea is simple: if fairness is impossible or too difficult to achieve for a single ranking (*i.e.*, for a single user recommendation), then we may rather measure fairness of exposure over a sequence of rankings. In fact, this notion aligns well with practical considerations in two-sided marketplaces (*e.g.*, e-commerce websites, streaming services, review websites), where consumers and producers meet, as illustrated in Example 13.

Example 13. *Amortized fairness in marketplaces.* Consider an e-commerce platform where producers from several groups $G_1, ..., G_K$ sell products and consumers visit and seek products to purchase. While fairer exposure of products can benefit both sides, the consequences of promoting fairness are not symmetrical. On the producer side, it is undeniably important to ensure fairness of exposure among the items from different groups of producers, since each producer should be granted fair opportunities to access the market and earn revenues based on the quality of their products. On the other hand, consumers may benefit from discovering more products, especially those looking for niche items. However, the primary concern for consumers is obtaining accurate recommendations, while the fairness of producer exposure in their own recommendations is unlikely to be a central consideration. If enforcing fairness constraints in a single recommendation results in a severe drop in relevance — which is especially likely for the average users who do not seek niche products —, a fairer exposure of products benefits producers far more than consumers. However, if we instead ensure fairness of exposure over a sequence of rankings, distributing the fairness constraints over time, each consumer is likely to be less impacted in terms of recommendation quality, while benefits remain the same for producers, better aligning with the objectives of both parties.

In order to amortize fairness of exposure over time, it suffices to measure exposure and enforce corresponding fairness constraints over recommendation sequences. Assume that, at each step $t = 1, ..., T$, the algorithm proposes a list or set $L_t = (i_{t,1}, ..., i_{t,n})$ to user u_t, defining a T-step recommendation sequence $\mathcal{R}_T$. A possible definition of amortized exposure over $\mathcal{R}_T$ is thus:

$$E_k = \frac{1}{T|G_k|} \cdot \sum_{t=1}^{T} \sum_{j=1}^{n} \mathbb{1}(i_{t,j} \in G_k) \cdot v_j \ ,$$

or, in the case of sets:

$$E_k = \frac{1}{Tn} \cdot \sum_{t=1}^{T} \sum_{j=1}^{n} \mathbb{1}(i_{t,j} \in G_k) \ .$$

Amortized fairness of exposure may therefore be a suitable notion of fairness for various sequential decision-making tasks, advantageously balancing the fairness constraints and the performance of the decision algorithm. It may also be seen as an allocation task, aiming to distribute exposure across different item groups over many users while minimizing disruption to individual user recommendation quality.

	Pre-processing	**In-processing**	**Out-processing**
Acts on...	Input data	Decision Algorithm	Output decisions
Strengths	Algorithm-agnostic and less dependent on specific fairness definitions	Specific and better fairness-performance trade-off	Algorithm-agnostic, effective at removing a specific bias
Weaknesses	Cannot influence algorithm internal biases, may distort patterns in the original dataset	Algorithm-specific	Worse trade-off between performance and fairness in general
Examples	Chen et al. [2023], Fang et al. [2022], Rastegarpanah et al. [2019], Salimi et al. [2019]	Yao and Huang [2017], Do et al. [2021], Biswas et al. [2022], Singh and Joachims [2018]	Naghiaei et al. [2022], Li et al. [2021b], Wu et al. [2021]

Table 2.2: Comparison of fairness-aware mechanisms.

2.2.2 Achieving Fairness

The last section focused on the key concepts and definitions of fairness in sequential decision-making. In the present section, we aim to provide a brief overview of mechanisms for achieving fairness in decision-making problems, and in particular fairness of exposure.

2.2.2.1 Typology of Fairness-Aware Mechanisms

Achieving fairness of exposure has been approached at different stages of the decision-making process, allowing us to group fairness mechanisms into three main categories: *pre-processing*, *in-processing* and *post-processing*.

- **Pre-processing** methods intervene directly on the input data, alleviating existing biases that the algorithm may inherit.
- **In-processing** methods modify the algorithm itself to incorporate fairness during the optimization process.
- **Post-processing** methods process the outputs of the algorithm to make them fairer.

Table 2.2 highlights the strengths and weaknesses of each approach.

2.2.2.2 Pre-Processing Approaches

A key reason behind the existence of bias in algorithms is thought to be the reproduction, or even the amplification of pre-existing bias in the training data [Pessach and Shmueli, 2020]. This is a serious concern, as the algorithm can perpetuate these biases or even reinforce them through its influence on real-world outcomes. Gender unfairness in hiring processes, illustrated in Example 9, as well as popularity bias, described for instance in Example 10, are examples of existing biases that may contaminate recommendation algorithms.

A natural approach to fairness is therefore to act directly on the data, removing or at least mitigating its biases. To this end, several works [Chen et al., 2023, Fang et al., 2022, Rastegarpanah et al., 2019] have proposed methods to inject new data D' into or in addition to the original training data D, resulting in a new dataset D_{fair} that is supposed to be less biased. This approach is referred to as *data augmentation* or *antidote data*. Salimi et al. [2019] take a slightly different approach by framing the problem as an optimization problem named *database repair*: starting from the initial dataset D, the goal is to find the closest dataset D_{fair} satisfying a set of constraints Γ ensuring the fairness of the input data.

One of the main advantages of pre-processing approaches is that they need not assume a specific fairness notion or measure and are generally agnostic to the recommendation algorithm, hence generalizing better than fairness interventions at later stages of the process.

2.2.2.3 In-Processing Approaches

Contrary to pre-processing methods, in-processing methods intervene directly during the optimization or learning process, incorporating fairness mechanisms that encourage fair decisions, even when the input data is biased.

In recommendation, in-processing methods generate ranked lists or sets directly from the set of available items. A natural approach is to introduce, besides the initial performance objective (*e.g.*, maximizing ranking relevance), an additional fairness objective or constraint that will encourage the optimization process to produce fairer recommendations. Consider the original, "unfair" recommendation problem:

$$\min \mathbf{J} \ ,$$

then, we can generally transform it into a fairness-aware recommendation problem as:

$$\min \mathbf{J} + \mathbf{M} \quad \text{or} \quad \min \mathbf{J} \text{ s.t. } \mathbf{M} \leq \epsilon.$$

$\mathbf{M}$ could be, for instance, one of the measures described in Sections 2.2.1.3 and 2.2.1.4. Here, fairness can also be seen as a penalty or regularization that drives the decision-making process towards fairer decisions. For example, Yao and Huang [2017] tackles fair collaborative filtering — an approach to recommendation that relies on user-item interaction data to infer preferences — with matrix factorization by adding an (sub-differentiable) "unfairness" measure to the original matrix factorization objective.

Some works formulate the problem of generating rankings as the computation of stochastic matrices or tensors. This approach allows to transform a very expensive combinatorial problem into a linear or convex optimization problem. Singh and Joachims [2018] see rankings as an optimization problem over the space doubly stochastic matrices, where the

objective is to maximize a linear objective of the form:

$$\max_{P \text{ doubly stochastic}} u^T P v \ ,$$

where a solution P is a matrix (P_{ij}) such that P_{ij} is the probability that item i is recommended at position j, u is a vector representing the items' expected relevance and v is a vector accounting for position bias. To ensure producer fairness of exposure, they further add a linear fairness constraint of the form $f^T P g = h$, where the vectors f, g and scalar h can encompass different notions of fairness. The resulting linear program can be efficiently solved to produce probabilistic rankings. Deterministic rankings are then inferred from the matrix P using the Birkhoff-von Neumann method. Similarly, Do et al. [2021] — which also take into account consumer fairness —, see rankings as three-way tensors P where P_{uij} denotes the probability that item i is recommended to user u at rank j. This formulation allows them to perform efficient convex optimization over the space of tensors, using the Frank-Wolfe algorithm. They frame fairness in terms of Lorenz dominance, leading to Pareto-optimal rankings that provide the best allocation of overall ranking quality and exposure over consumers and producers, respectively.

Biswas et al. [2022] also address the problem of two-sided recommendation and frame the task as an allocation of the producers' exposure over consumers, amortizing fairness over many rankings — although, similarly to Do et al. [2021], they are not exactly in a sequential setting since the whole batch of consumers is a system input.

By introducing fairness directly into the decision-making optimization process, in-processing approaches can be really effective at achieving a specific trade-off between performance and fairness. On the downside, these approaches are algorithm-specific, making them less generally applicable than pre- and post-processing methods.

2.2.2.4 Post-Processing Approaches

Finally, post-processing methods modify the output of the decision algorithm.

In the context of rankings, post-processing is often understood as *re-ranking*, that is either permuting or extracting items in the original ranked list L to obtain a list with fairer exposure, L_{fair}.

In the context of two-sided fairness with producer fairness of exposure, where all users $u \in \mathcal{U}$ are known in advance to the system, Naghiaei et al. [2022] formulate the problem as follows: given top-N recommendation lists provided by a non-fair recommendation algorithm for each user $u \in \mathcal{U}$ and the associated relevance scores $(S_{ui})_{u \in \mathcal{U}, i=1,\dots,N}$, they seek to extract fair top-K lists, with $K < N$. The algorithm decisions are represented as a binary matrix

$A = (A_{ui})_{u \in \mathcal{U}, i=1,\dots,N}$ such that:

$$A_{ui} = \begin{cases} 1, & \text{if } i \text{ is recommended to } u \text{ in the fair } K\text{-list} \\ 0, & \text{otherwise.} \end{cases} .$$

The task therefore consists in finding the matrix A that maximizes recommendation relevance with *ad hoc* consumer and producer fairness penalties. Li et al. [2021b] propose a very similar approach, introducing fairness as a constraint instead of a penalty.

As pre-processing methods, post-processing methods have the advantage of being agnostic to the decision-making algorithm. Applied later in the decision process, their action is easy to understand and they are effective at removing decision bias; however, having less agency over the whole decision process, they also tend to have a worse impact on accuracy.

Chapter Three

Pure Exploration in Multi-Agent Bandits with Private Communications

After providing a general background on privacy and fairness in the previous chapter, this chapter presents the first contribution of this thesis. It addresses multi-agent multi-armed bandits, a powerful framework for the optimization of large-scale distributed systems. Precisely, we study the problem of fixed-confidence best arm identification in this setting, also called *pure exploration*, with applications ranging from parameter-tuning in telecommunication networks to pharmaceutical drug selection. We identify that in such multi-agent systems, inter-agent communications, while essential for large-scale collaboration, are a key privacy vulnerability, as they can be readily intercepted by malicious adversaries seeking to steal valuable data. In response to this threat, we propose novel methods for multi-agent best arm identification based on private communication protocols.

This contribution led to a publication at *ICML 2023* [Rio et al., 2023].

3.1 Context

The process of gathering data to identify the best option in a stochastic multi-armed bandit is known as *pure exploration* or *best arm identification* (BAI). The goal for the learner is to suggest the most valuable option given the data collected through sequential interaction with the system. In the specific *fixed-confidence* setting, we seek to achieve this goal using as few interactions as possible, while guaranteeing that the actual best option is returned with fixed, high probability. Motivating examples include automatic parameter tuning of telecommunication antennas [Vannella, 2023] and pharmaceutical drug selection [Varatharajah et al., 2018]. Antenna performance depends on a large number of parameters, making it difficult and time-consuming for engineers to manually explore the space of configurations and find the one that will optimize the user experience. Selecting the best drug among multiple candidates is also a complex and costly process. Re-framing these problems as best arm identification in multi-armed bandits and relying on automatic strategies can be very

effective [Shang et al., 2019].

Such problems usually involve multiple agents that share and interact with the same environment, generating as many data streams. Clinical trials involve many participants to select the most appropriate drug for the majority of the population. Antennas are generally distributed over networks spanning a specific region. Moreover, in many situations, data is distributed by nature, and cannot be consolidated and processed on a single server. It is therefore interesting for the companies to model such problems as multi-agent systems and try to exploit the data collected by each agent (i.e., *locally*) to learn the optimal decision at the system level (i.e., *globally*). We therefore refer to this problem as *multi-agent best arm identification*. Collaboration among agents is made effective as each agent broadcasts messages, reflecting the feedback they receive from the system, to a central coordinator involved in the decision process.

Large amounts of data can be handled during such learning processes, and some of them are potentially sensitive (e.g., personal data or highly valued industrial information). Hence, it is critical to design multi-agent BAI methods that ensure the privacy of the data identified as sensitive. In many applications, one may want to protect the rewards used as inputs of the learning algorithm, especially since they reflect the preferences of the agents. For instance, confidential medical data may be at risk during a drug selection study. In this situation, the *differential privacy* framework ensures good guarantees against privacy leakage. Some other cases may require protecting the algorithm's output, that is the global optimal action. In our motivating examples, both the optimal antenna configuration and the best candidate drug are indeed of great value, as the product of the expertise of the engineers. Protecting it from competing firms is therefore of high importance. These concerns call for another notion of privacy, like the lesser known (α, η)-privacy from Féraud et al. [2019].

However, if it may be possible to protect the data held by the different participants (the central coordinator and the agents), the communications may still be easily intercepted by an adversary. Moreover, they cannot always be encrypted, especially in the multi-agent setting where the computational overhead would increase significantly and where the exchange of keys is not obvious. Consequently, whereas communications from agents are a crucial building block of multi-agent algorithms, they are often a weak link in terms of privacy. Indeed, if these messages carry enough information, they can leak sensitive data. Designing multi-agent solutions that protect sensitive information from adversaries observing the communications is thus crucial.

Contributions. We refer to this problem as *multi-agent best arm identification with private communications*. In this chapter, we investigate solutions to this problem, considering two scenarios based on the type of information that we want to protect: 1) the rewards, i.e. the input data, and 2) the estimated best arm, i.e., the output data. This leads us to question the most appropriate privacy notion to use in each scenario. In the first scenario, we propose DP-MASE, a multi-agent version of *DP Successive Elimination* [Sajed and Sheffet,

2019]. DP-MASE is the first attempt to protect the input data in the multi-agent setting with sensitive communications, using differential privacy. To address the second scenario, we rely on the notion of (α, η)-privacy, introduced in Féraud et al. [2019], and propose Corrupted Elimination that improves over their method by decoupling privacy and local BAI accuracy, leading to better sample complexity and providing flexibility to adjust these two essential aspects of the problem. We provide a theoretical analysis for the privacy and performance of both algorithms. In particular, we show that multi-agent collaboration leads to greater accuracy than independent agents, and derive associated sample complexity bounds. Experiments on toy environments support this theoretical findings.

3.2 Related Work

Multi-armed bandits. In multi-armed bandits (MAB), the reference theoretical framework for sequential decision-making [Thompson, 1933, Robbins, 1952, Lattimore and Szepesvári, 2020]), the problem of identifying the option (called *arm*) with the highest mean is known as *Pure Exploration* or *Best Arm Identification* (BAI). Unlike the problem of regret minimization, BAI assumes that we incur no cost in exploring suboptimal options and are just interested in returning the best arm for further exploitation. BAI has been thoroughly studied in two different settings: *fixed-budget* [Audibert and Bubeck, 2010] and *fixed-confidence* [Even-Dar et al., 2006]. In the former, we aim at returning the best arm in a given amount of time while minimizing the probability of failure, whereas in the latter we want to output the best arm as quickly as possible with a fixed probability of failure. In this paper, we focus on the fixed confidence setting, for which Even-Dar et al. [2006] formulate and analyze two standard BAI algorithms: *Successive Elimination* and *Median Elimination*. The theoretical analysis is conducted in the *Probably Approximately Correct* (PAC) framework, where we accept to only find an α-optimal arm with high probability. More advanced analysis and algorithms have been discussed, for instance in Jamieson and Nowak [2014].

Multi-agent MAB. Recently, much attention has been paid to multi-agent multi-armed bandits, where multiple agents collaborate to solve the same MAB instance. First, this approach can allow to take advantage of multiple computing nodes to speed up the learning process [Hillel et al., 2013, Szörényi et al., 2013]. More importantly, it accurately models real-world problems where multiple agents combine their efforts to solve a given learning task. A traditional example is the device cloud, where a possibly wide range of remote smart devices, each producing and owning their data, are solicited for the learning of a common model. However, Madhushani and Leonard [2020] point out the limits of natural extensions of traditional bandit algorithms like UCB to the multi-agent setting. A significant part of the research has been directed toward networked agents. Sankararaman et al. [2019] address the regret minimization problem by initially sharing arms among the agents and propagating the best arm using gossip protocols, much like Chawla et al. [2020] and Agarwal et al. [2021]. Kumar Kolla et al. [2018] design UCB-like algorithms where statistics are aggregated over

graph neighborhoods; a work recently extended to BAI by Jha et al. [2022]. Other works consider a slightly different setting, named multi-bandit, where the agents have access to different bandit instances, with different arms [Gabillon et al., 2011, Scarlett et al., 2019] or different preferences [Shi et al., 2021, Réda et al., 2022].

Privacy in bandits. In the multi-agent setting, privacy is more than ever a concern, as sensitive user data may be leaked through the messages sent by the participants. Building private algorithms is therefore a major issue. While the very definition of privacy may vary according to the context, there are two main families of techniques to design such algorithms in the MAB framework: *cryptography* [Ciucanu et al., 2022, Garcelon et al., 2022], and *differential privacy* (DP) [Dubey and Pentland, 2020, Li and Song, 2022]. Encrypting the data usually has the advantage of being less harmful to the accuracy of the algorithm at the cost of additional computations, while differential privacy, consisting of adding noise in the data, requires little computation but may reduce accuracy. Differential privacy is more widely used in multi-agent settings. Tossou and Dimitrakakis [2016] and Ren et al. [2020] discuss the practical design of differentially-private MAB algorithms for the regret minimization problem, using additional Laplace noise. They also provide theoretical regret bounds that exhibit the cost of this mechanism compared to equivalent non-private algorithms. Dubey and Pentland [2020] and Li and Song [2022] use DP mechanisms to address privacy preservation in the specific multi-agent setting, where communications between agents add an additional layer of risk. Privacy guarantees for pure exploration have been less investigated. Sajed and Sheffet [2019], for instance, propose a differentially private version of *Successive Elimination*, first described in Dwork et al. [2009], to eliminate suboptimal arms while guaranteeing privacy. Féraud et al. [2019], on their part, address privacy in multi-agent best arm identification with a different approach. In contrast to *differential privacy*, the goal is no longer to protect the reward information that serves as input to the learning algorithm, but rather to protect its output, i.e., the identified optimal arm. Arms are eliminated both locally and globally with a voting system while having weak (not accurate) local agents ensures that an adversary cannot infer individual agent preferences by simply observing communications.

3.3 Preliminaries

3.3.1 Multi-Agent Bandits

We model our problem using the stochastic multi-armed bandit framework, stated in Definition 3.3.1 and illustrated in Figure 3.1.

Definition 3.3.1. *Stochastic Multi-Armed Bandit.* A stochastic multi-armed bandit (MAB) with K arms, or K-armed bandit, is a set of K stochastic real distributions $(R_1, ..., R_K)$, bounded in $[0, 1]$, with means $\mathbb{E}[R_k] = \mu_k$. When a user interacts with the MAB, the user *pulls* arm $k \in \mathcal{K} = \{1, ..., K\}$ and receives a *reward* $r \sim R_k$.

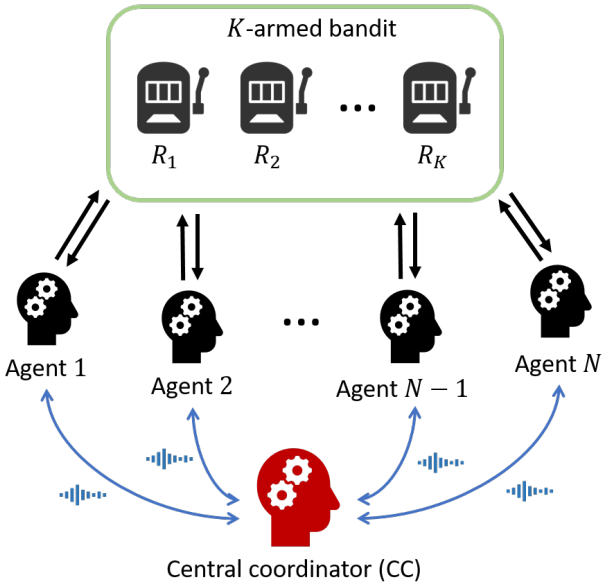


Figure 3.1: Illustration of the multi-agent multi-armed bandit framework.

We consider a multi-agent setting where a set $\mathcal{N} = \{1, ..., N\}$ of N agents collaborates to identify the best arm in a common K-armed bandit instance. When an agent $n \in \mathcal{N}$ interacts with the system at time $t \in \mathbb{N}^*$ and pulls an arm $k \in \mathcal{K}$, it receives a stochastic reward $r_k^n(t)$ drawn from some unknown distribution R_k with mean μ_k. Without loss of generality, we assume $\mu_1 \geq ... \geq \mu_K$. The objective for the system is to return an α-optimal arm, *i.e.*, an arm $k^\star$ such that $\mu_{k^\star} \geq \mu_1 - \alpha$ with high probability.

Assumption 3.3.2. *Active agent.* At any round t, only one agent $n_t \in \mathcal{N}$, the *active agent*, can interact with the system.

Assumption 3.3.3. *Fixed agent distribution.* The active agent at time t is sampled from a fixed distribution $P_{\mathcal{N}}$ on the agents $\mathcal{N}$, given by the environment.

Once active, an agent pulls every arm in his local active set (that is, the set of arms that the agent still considers to be potentially optimal). Over the course of the algorithm, arm k's mean is estimated through the following quantity:

$$\hat{\mu}_k^n(t) = \frac{1}{t^n} \sum_{\tau=1}^{t} r_k^n(\tau) \mathbb{1}\left[n_\tau = n\right] \quad ,$$

where t^n is the number of rounds agent n has been active up to time t, such that $t^n := t^n(t) = \sum_{\tau=1}^{t} \mathbb{1}\left[n_\tau = n\right]$. Assumption 3.3.2 simply models asynchronous interactions with the system, corresponding to several real-world applications (e.g., telecommunication networks).

We consider a centralized setting so that agents collaborate by sending messages to a central coordinator. This is equivalent to considering a decentralized setting where agents are connected with a complete graph. The messages contain the indices of the arms they

have eliminated locally. The messages sent by agent n up to time t can be summarized as a vector $(\lambda_k^n(t))_{k\in[K]}$ where $\lambda_k^n(t) = 1$ if agent n has already eliminated arm k at time t, and $\lambda_k^n(t) = 0$ otherwise.

Assumption 3.3.4. *Communication between agents.* Each time the active agent n_t eliminates an arm locally, he sends its index k to the central coordinator.

Notations. Let $\mathcal{M}_n(t) = (\lambda_k^n(t))_{k\in[K]}$ denote the messages sent by agent $n \in \mathcal{N}$ up to time t, and $\mathcal{M}(t) = (\mathcal{M}_1(t), ..., \mathcal{M}_N(t))$. Let also $\mathcal{M}_n$ denote the set of all messages sent by agent $n \in \mathcal{N}$ and $\mathcal{M} = (\mathcal{M}_1, ..., \mathcal{M}_N)$.

3.3.2 Multi-Agent BAI with Private Communications

In the context of *multi-agent BAI*, we are concerned with withholding two types of data one may want to protect : 1) the input data, namely the rewards, and 2) the output of the BAI algorithm, namely the identity of the optimal option. Moreover, we assume that the privacy of these data may be compromised through the messages sent by the agents. We thus propose privacy notions that aim at protecting the privacy of either the inputs or the output of multi-agent BAI algorithms from adversaries observing the communications.

3.3.2.1 Protecting the Input with Differential Privacy

The first situation is typically handled using *differential privacy* (see Chapter 2). A BAI algorithm learns the best arm based on the sequence of rewards collected through interactions with the system: these rewards may reveal sensitive information as they reflect the preferences of the agents. Since we consider that an adversary may observe the communicated messages (and not the best arm itself), we seek differential privacy with respect to the mechanism that outputs the set of messages sent by one agent. We then guarantee that the presence of one particular reward in the input dataset does not affect the message distribution significantly.

We give in Definition 3.3.5 a statement of DP adapted to our particular setting. Let $D_{1:t}$ be the set of all the rewards received by the N agents up to time t. $D_{1:t}$ and $D'_{1:t}$ are considered neighbors if they only differ by one reward. We consider the mechanism $\mathcal{Q}^n$ that takes as input a dataset $D_{1:t}$ and outputs $\mathcal{M}_n$.

Definition 3.3.5. *ϵ-differential privacy for communications.* Given $\epsilon > 0$, for any agent $n \in \mathcal{N}$, the mechanism $\mathcal{Q}_n$ is ϵ-differentially private if for any time t, any set of messages $\mathcal{M}_n(t) \in \{0,1\}^{[K]}$, and any pair of neighboring datasets $D_{1:t}$, $D'_{1:t}$, the following holds:

$$\mathbb{P}\left(\mathcal{Q}_n(D_{1:t}) = \mathcal{M}_n(t)\right) \leq e^{\epsilon}\mathbb{P}\left(\mathcal{Q}_n(D'_{1:t}) = \mathcal{M}_n(t)\right) \quad .$$

3.3.2.2 Protecting the Output with (α, η)-Privacy

Some applications may require protecting the algorithm's output, i.e., the global optimal action. This information can indeed be of high value, for instance when it is the product of the engineers' expertise in industrial applications. Protecting it from competing firms is therefore of critical importance. However, in multi-agent systems, it is difficult to prevent adversaries from observing the messages sent by the different participants. If those messages carry enough information about the optimal action, their identity can leak—even if it is well protected at the system level.

We are interested in designing methods intended to protect against this eventuality, which is formalized through the notion of (α, η)-privacy, first introduced in Féraud et al. [2019]. (α, η)-privacy, stated in Definition 3.3.6, ensures that an adversary observing the messages sent by one agent cannot infer an α-best arm with high confidence. Limiting the awareness of the adversary to the messages from a single agent is realistic in many applications, as we can imagine installing a bug on one particular device. However, this could be easily extended to the setting where the adversary may watch up to c agents, given a loss in the privacy guarantees.

Definition 3.3.6. *(α, η)-privacy.* An algorithm $\mathcal{A}$ is considered (α, η)-private for finding an α-optimal arm if, for any agent $n \in \mathcal{N}$, an adversary that knows $\mathcal{A}$ and has access to the set of messages $\mathcal{M}_n$ cannot infer what arm is α-optimal for agent n with probability higher than $1 - \eta$.

In particular, if $1 - \eta \leq 1/K$, the adversary cannot make better than a random guess.

3.3.3 Successive Elimination in Multi-Agent Bandits

In multi-agent algorithms, messages are signals that reflect the feedback an agent receives through his interactions with the environment. Sharing the signals with the other agents enables collaboration and allows better decision-making at the global level, compared to the situation where each agent learns independently. In multi-agent bandits, one could imagine a variety of such signals. However, in many applications, the amount of information contained in the messages is constrained by the capacity of the communication channels. In this work, we consider simple messages with integer values, fitting situations where communication is heavily constrained.

BAI algorithms based on successive arm elimination are particularly suited to this setting. These methods work by maintaining an active set of arms that is progressively reduced as the agent eliminates suboptimal arms. An arm is eliminated when it is estimated suboptimal with enough confidence, based on the rewards collected during interactions with the system. The algorithm stops when the active set is reduced to a single arm, which is the estimated

Algorithm 2 Multi-Agent Successive Elimination (MASE)

Input: $\alpha > 0$, $\delta \in [0,1]$, voting threshold $M \in [\![1, N]\!]$
Output: Estimated best arm in $\mathcal{K}$
Active sets $\mathcal{K} = \mathcal{K}^n = [K]$; global votes $(\lambda_k = 0)_{k\in[K]}$; local epochs $e^n = 0$; local round $r^n = 0$
while $|\mathcal{K}| > 1$ **do**
  Active agent n is sampled: $n \sim P_{\mathcal{N}(t)}$
  n interacts with system and updates mean estimates
  $t^n := t^n + 1$
  if condition for local elimination is met by k **then**
    Remove k from local active set: $\mathcal{K}^n := \mathcal{K}^n \setminus \{k\}$
    $\lambda_k := \lambda_k + 1$
    $e^n := e^n + 1$
    if $\lambda_k > M$ **then**
      Remove k from global active set: $\mathcal{K} := \mathcal{K} \setminus \{k\}$
      Remove k from all local active sets: $\mathcal{K}^n := \mathcal{K}^n \setminus \{k\}$ for all $n \in \mathcal{N}$
    end if
  end if
end while

best arm. As an important example, Successive Elimination from Even-Dar et al. [2006] guarantees to output an α-optimal arm with confidence $1-\delta$ in approximately $\frac{K}{\alpha^2}\log\frac{1}{\delta}$ steps.

It is quite straightforward to use successive elimination schemes within multi-agent systems and enable collaboration through integer-valued messages and a voting process. At a high level, each agent n independently runs a local BAI algorithm, managing his own local set of arms $\mathcal{K}^n$, while a central coordinator separately maintains a global set of arms $\mathcal{K}$. Every time an agent eliminates an arm locally, he sends its index to the central coordinator, hence "voting against" this arm. If the number of votes λ_k against a specific arm exceeds some predefined threshold M, it is removed from the global active set as well as from all local active sets. This multi-agent successive elimination (MASE) scheme at both local and global levels is described in Algorithm 2. In the following, we propose methods based on this multi-agent successive elimination scheme that ensures differential privacy and (α, η)-privacy.

Sample Complexity of MASE. We expect the sample complexity, computed as the total number of overall interactions with the system, to scale up with the number of agents. Indeed, in the asynchronous case (Assumption 3.3.2), more agents means more timesteps before a given agent has interacted enough with the system to output the best arm with high confidence.

Formally, the per-agent counts $(t^1, ..., t^N)$ follow a multinomial distribution with parameters $(t, (p_1, ..., p_N))$, where $p_i = P_{\mathcal{N}}(i)$. Moreover, the algorithm stops at step t if $\max_M \left(t^1, ..., t^N\right) > T(\eta)$, where $\max_M$ denotes the M-th largest value. For a given T, the probability $\mathbb{P}\left(\max_M (T_1, ..., T_N) > T(\eta)\right)$ decreases with N, which necessarily links the

sample complexity to N.

For a lower bound, we consider the extreme, non-private case, where agents share all their raw reward samples with the central coordinator (the set of messages that can be sent is then larger than what is described in Section 3.3.1). Up to negligible communication delays, the problem becomes equivalent to the single-agent setting. It is then relevant to compare our algorithms with single-agent Successive Elimination, for which lower bounds exist (see, for instance, Even-Dar et al. [2002]), although such approach does not offer any privacy.

3.4 Multi-Agent Successive Elimination with Differentially Private Communications

In this section, we introduce DP-MASE (DP Multi-Agent Successive Elimination). While learning the best option in the bandit instance, DP-MASE guarantees that the communications are differentially private with respect to the rewards. In essence, DP-MASE follows the multi-agent successive elimination scheme described in Algorithm 2, and uses DP Successive Elimination [Sajed and Sheffet, 2019] as a local BAI routine.

3.4.1 Differentially Private Successive Elimination

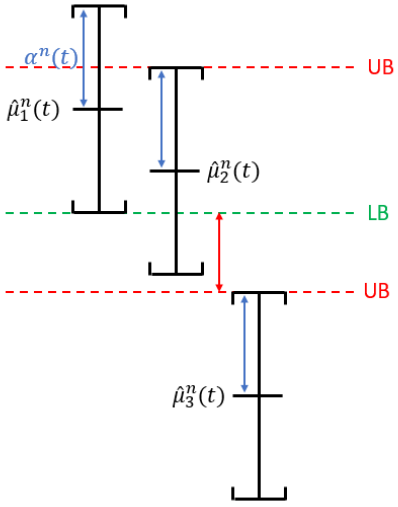


Figure 3.2: Illustration of Successive Elimination [Even-Dar et al., 2006].

In Successive Elimination [Even-Dar et al., 2006], illustrated in Figure 3.2, we consider the uncertainty over the value of each arm using a confidence interval of decreasing radius around the empirical mean. More specifically, at step t, we want the true mean of

arm k to lie within the following confidence interval, for a well-chosen $\alpha(t) \in \mathbb{R}$, with high probability:

$$\mu_k \in [\hat{\mu}_k(t) - \alpha(t); \hat{\mu}_k(t) + \alpha(t)] \quad .$$

If we set $\alpha(t) = \sqrt{\frac{\log(cKt^2/\beta)}{t}}$ for some $c > 0$, we know this holds with high probability $1 - \frac{2\beta}{cKt^2}$ due to Hoeffding's inequality. An arm k is eliminated as soon as its upper bound is less than the lower bound of the current best arm estimate, i.e.:

$$\max_{l \in \mathcal{K}} \hat{\mu}_l(t) - \hat{\mu}_k(t) > 2\alpha(t) \quad . \tag{3.1}$$

From then on, we refer to (3.1) as *suboptimality evaluation*. This procedure returns an ϵ-optimal best arm with probability at least $1 - \beta$, in at most $\mathcal{O}\left(\frac{K}{\epsilon^2} \log \frac{1}{\beta}\right)$ steps.

Making Successive Elimination differentially private is not straightforward. Naively, one could simply release noisy values for the means instead of raw values (for instance using the Laplace mechanism). However, this approach raises practical challenges. Indeed, a reward sample from arm k is involved in every subsequent empirical mean $\hat{\mu}_k$, making the privacy cost scale linearly with T, the total number of steps. To circumvent this, Sajed and Sheffet [2019] only perform suboptimality evaluation after a given number of steps (*i.e.* an epoch). Empirical means are then reset and samples are discarded. Therefore, each reward sample is only consumed in a single mean estimation. This approach also controls the global sensitivity of the queries and the scale of the Laplace noise appropriately.

Thanks to parallel composition, Sajed and Sheffet [2019] prove that DP Successive Elimination is ϵ-differentially private. It also effectively returns the best arm as the epoch length $R(e)$ grows properly over time, each time allowing for more accurate mean estimates. Indeed, $R(e)$ is such that every arm with a suboptimality gap greater than 2^{-e} is eliminated at epoch e with high probability.

3.4.2 A DP Algorithm for Multi-Agent BAI



Figure 3.3: Illustration of DP-MASE.

Our objective is now to propose a multi-agent version of DP Successive Elimination, based on the MASE scheme. We obtain DP-MASE, a new algorithm for multi-agent BAI with private communications (see Figure 3.3), depicted in Algorithm 3.

In DP-MASE, each agent independently runs a DP Successive Elimination instance on the shared multi-armed bandit. To deal with the asynchronous interactions, every agent

Algorithm 3 DP-MASE

1: **Input:** Privacy level ϵ, global failure probability δ, local failure probability β, voting threshold $M = \lceil \frac{\log \delta}{\log \beta} \rceil$
2: **Output:** Estimated best arm in $\mathcal{K}$
3: Global and local active sets $\mathcal{K} = \mathcal{K}^n = [K]$; global votes $(\lambda_k = 0)_{k\in[K]}$; local epochs $e^n = 0$; local round $r^n = 0$; global time $t = 0$
4: **while** $|\mathcal{K}| > 1$ **do**
5: Active agent n is sampled: $n \sim P_{\mathcal{N}}$
6: Sample all arms in $\mathcal{K}^n$ and update mean estimates
7: $r^n := r^n + 1$
8: **if** $r^n = R(e^n)$ **then**
9: $\bar{k}(e^n) = \text{LocalElimination}\,(\epsilon, e^n, (\hat{\mu}_k^n(e^n))_k, \beta)$
10: $\mathcal{K}^n = \mathcal{K}^n \setminus \bar{k}(e^n)$
11: $\lambda_k := \lambda_k + 1$ for any $k \in \bar{k}(e^n)$
12: **if** $|\mathcal{K}^n| = 1$ **then**
13: $\mathcal{N} \leftarrow \mathcal{N} \setminus \{n\}$ {Remove agent if done}
14: **else**
15: $e^n := e^n + 1$
16: Reset $(\hat{\mu}_k^n(e^n) = 0)_k$, $r^n = 0$
17: Compute next epoch length $R(e^n)$
18: $\mathcal{K}^n := \mathcal{K}^n \bigcap \mathcal{K}$ {Remove the latest globally eliminated arms from local active set}
19: **end if**
20: **for** $k \in \bar{k}(e^n)$ **do**
21: $\mathcal{K} := \mathcal{K} \setminus \{k\}$ **if** $\lambda_k > M$
22: **end for**
23: **end if**
24: t := t + 1
25: **end while**

keeps track of his current epoch e^n and current round $r^n \in [\![0, R(e^n)]\!]$ within this epoch. At the end of their epoch, an agent performs local elimination as in Sajed and Sheffet [2019] (see Algorithm 4), returning a set $\bar{k}(e^n)$ of eliminated arms. If it is not empty, arms in $\bar{k}(e^n)$ are immediately removed from the local active set $\mathcal{K}^n$, and their global voting counts λ_k's are updated (lines 9-11). Empirical means are then reset for the next epoch whose length is also computed using the same formula as in Sajed and Sheffet [2019]:

$$\max\left(\frac{32\log\left(8|\mathcal{K}^n|e^{n2}/\beta\right)}{2^{-2e^n}}; \frac{8\log\left(4|\mathcal{K}^n|e^{n2}/\beta\right)}{\epsilon 2^{-e^n}}\right) \quad .$$

Moreover, $\mathcal{K}^n$ is synchronized with the global active set $\mathcal{K}$, meaning that arms that have been globally eliminated during epoch e^n are now also removed from $\mathcal{K}^n$ (lines 15-18). Eventually, if votes against arms in $\bar{k}(e^n)$ have reached the voting threshold, they are removed from global active set $\mathcal{K}$ (lines 19-20).

We now provide different theoretical guarantees for DP-MASE. We first show that DP-

Algorithm 4 DP-MASE Local Elimination

Input: Privacy level ϵ, current local epoch e^n, estimated means $(\hat{\mu}_k^n(e^n))_{\mathcal{K}^n}$, local failure probability β
Output: A set of arms $\bar{k}(e^n)$ to eliminate locally at agent n
$\bar{k}(e^n) = \{\}$
Set $h(e^n) = \sqrt{\frac{\log(8|\mathcal{K}^n(e^n)|e^{n\,2}/\beta)}{2R(e^n)}}$
Set $c(e^n) = \frac{\log\left(4|\mathcal{K}^n(e^n)|e^{n\,2}/\beta\right)}{\epsilon R(e^n)}$
Perturb means: $\tilde{\mu}_k^n = \hat{\mu}_k^n(e^n) + \mathrm{Lap}\left(1/\epsilon R(e^n)\right)$
for all $k \in \mathcal{K}^n$ **do**
  if $\max_{l\in\mathcal{K}^n} \tilde{\mu}_l^n - \tilde{\mu}_k^n > 2\left(h(e^n) + c(e^n)\right)$ **then**
    $\bar{k}(e^n) := \bar{k}(e^n) \cup \{k\}$ {Add k to the set of arms to eliminate if elimination condition is met}
  end if
end for

MASE is ϵ-differentially private for communications (Proposition 3.4.1), in the sense of Definition 3.3.5, and also derive guarantees over its accuracy for finding an optimal arm (Proposition 3.4.2).

Proposition 3.4.1. Privacy of DP-MASE. *Algorithm 3, DP-MASE, is ϵ-differentially private for communications.*

Proof. Let $D(n, e^n)$ be the sequence of all rewards collected by agent n at epoch e^n and $\tilde{\mathcal{Q}}_{e^n}$ be the mechanism that outputs the indices of the arms eliminated by agent n at epoch e^n. We fix the global step t, and denote $\mathcal{T}^n$ the set of agent n's local epochs occurring up to time t. Therefore, $D_{1:t} = \bigcup_{n\in\mathcal{N}, e^n\in\mathcal{T}^n} D(n, e^n)$.

An agent n has knowledge of the latest globally eliminated arms only at the end of their current epoch, so that the reward samples in $D(n, e^n)$ have no effect on the local elimination process for any other agent m. Thus, $\tilde{\mathcal{Q}}_{e^n}$ operates only on $D(n, e^n)$. In other words, each reward sample only plays a role for a single agent, at a single epoch.

At epoch e^n, for every arm k in the active set $\mathcal{K}^n$, the empirical mean $\hat{\mu_k}(e^n)$ is computed using $R(e^n)$ samples. The sensitivity of $\hat{\mu_k}(e^n)$ is therefore $1/R(e^n)$. Moreover, each noisy mean is computed only once before suboptimality evaluations. Therefore, the local elimination mechanism which uses Laplace noise with scale $\frac{1}{\epsilon R(e^n)}$ is ϵ-DP. $\tilde{\mathcal{Q}}_{e^n}$ is thus ϵ-DP.

Consider the composed mechanism $\tilde{Q}$ defined as:

$$\tilde{Q}(D_{1:t}) = \left(\tilde{\mathcal{Q}}_{e^n}(D(n, e^n))\right)_{n\in\mathcal{N}, e^n\in\mathcal{T}^n} .$$

Thanks to the parallel composition theorem (see Theorem 4 in McSherry [2009]), $\tilde{Q}$ is ϵ-DP. However, $\mathcal{Q}_n$ can be seen as a deterministic function g of $\left(\tilde{\mathcal{Q}}_{e^n}\right)_{n,e^n}$, *i.e.*, $\mathcal{Q}_n = g\left(\left(\tilde{\mathcal{Q}}_{e^n}\right)_{n,e^n}\right)$. So $\mathcal{Q}_n$ is also ϵ-DP by post-processing (see Proposition 2.1 in Dwork and Roth [2014]). □

Proposition 3.4.2. Failure Probability of DP-MASE. *With voting threshold* $M = \lceil \frac{\log \delta}{\log \beta} \rceil$, *DP-MASE fails to return the best arm with probability at most* δ.

Proof. Sajed and Sheffet [2019] show that using DP Successive Elimination, the optimal arm $k^\star$ is never eliminated with probability at least $1 - \beta$. Following Property 1 from Féraud et al. [2019] applied to successive elimination algorithms, the probability of an agent eliminating $k^\star$ at given local epoch is therefore at most β/K.

An arm k is eliminated when the events $\{$agent n has voted against $k\}$ occur for $M = \lfloor \frac{\log \delta}{\log \beta} \rfloor$ players. By design, there exists a time $t = \sum_{n=1^N}$ such that for $K-1$ arms, there are $\lfloor \frac{\log \delta}{\log \beta} \rfloor$ voting players.

Consider the event that one of those arms is $k^\star$, $E = \cup_{l=1}^{K} \{l$-th eliminated arm is $k^\star\}$. By the union bound:

$$\mathbb{P}(E) \leq \sum_{l=1}^{K} \mathbb{P}(\{l\text{-th eliminated arm is } k^\star\})$$

.

Let $\mathcal{S}$ be the set of M players that have voted against the eliminated arm at global elimination round l, and $V_i^l \in \{0, 1\}$ the random variable denoting that this vote is against the best arm $k^\star$.

$$\begin{aligned}
\mathbb{P}(\{l\text{-th eliminated arm is } k^\star\}) &= \mathbb{P}(\cap_{i \in \mathcal{S}} \{V_i^l = 1\}) \\
&= \prod_{i \in \mathcal{S}} \mathbb{P}(\{V_i^l = 1\}) \\
&\leq \prod_{i \in \mathcal{S}} \beta/K = (\beta/K)^M
\end{aligned}$$

So:

$$\mathbb{P}(E) \leq \sum_{l=1}^{K} (\beta/K)^M = K(\beta/K)^M = \beta^M / K^{M-1} \leq \beta^M \quad .$$

□

Eventually, we derive in Proposition 3.4.3 a problem-dependent sample complexity bound for DP-MASE that holds with high probability.

Proposition 3.4.3. Sample Complexity of DP-MASE. *With probability at least* $(1-\delta) \times (1 - I_{1-p^\star}(T - T(\beta), 1 + T(\beta)))^M$, *DP-MASE has sample complexity:*

$$\mathcal{O}\left(\tfrac{1}{p^\star} T(\beta) + \tfrac{1}{4p^{\star 2}} \log \tfrac{1}{\delta}\right) \quad ,$$

with:

$$T(\beta) = \log(K/\beta) + \log\log(1/\Delta_2)\left(\frac{1}{\Delta_2^2} + \frac{1}{\epsilon\Delta_2}\right) \ .$$

Above, $I_p(a,b)$ is the regularized incomplete beta function evaluated at p with parameters (a,b), and $p^\star$ is the probability of the M-th most likely agent.

To prove Proposition 3.4.3, we first prove the following lemma, which is an updated version of the sample complexity bound from Féraud et al. [2019].

Lemma 3.4.4. Sample complexity of Féraud et al. [2019]. *Using any BAI subroutine, with probability at least $(1-\delta)\left(1 - I_{1-p^\star}\left(T - T(\eta), 1 + T(\eta)\right)\right)^M$, the multi-agent BAI algorithm proposed in Féraud et al. [2019] has sample complexity*

$$\mathcal{O}\left(\frac{1}{p^\star}T(\eta) + \frac{1}{4p^{\star 2}}\log\frac{1}{\delta}\right) \ ,$$

where $I_p(a,b)$ is the regularized incomplete beta function evaluated at p with parameters (a,b), $p^\star$ is the probability of the M-th most likely agent, and $T(\eta)$ is the number of local samples needed for the BAI subroutine to find the optimal arm with confidence $1-\eta$.

Proof. (Lemma 3.4.4)

We follow a similar line of reasoning to Féraud et al. [2019] for the derivation of the sample complexity bound.

Let T denote the number of rounds before the algorithm stops (*i.e.*, its sample complexity) and T_n denote the number of rounds agent n has been active. $P_{\mathcal{N}}$ is the probability distribution over agents, so that $P_{\mathcal{N}}(x = n)$ is the probability that the active player x is n at any time. We have $T_n \sim \mathcal{B}(T, P_{\mathcal{N}}(x = n))$, and thus $\mathbb{E}_{P_{\mathcal{N}}}[T_n] = P_{\mathcal{N}}(x = n)T$.

Let $B_{\delta,\eta}$ be the set of agents with the $M := \lceil \frac{\log\delta}{\log\eta} \rceil$ highest T_n. The algorithm does not stop if the following event occurs: $E_1 = \{\exists n \in B_{\delta,\eta}, T_n < T(\eta)\}$.

Applying Hoeffding's inequality, we have:

$$\mathbb{P}\left(T_n - P_{\mathcal{N}}(x = n)T \leq -\epsilon\right) \leq \exp\left(-2\epsilon^2/T\right) \ . \tag{3.2}$$

When $\neg E_1$ occurs, all agents in $B_{\delta,\eta}$ have enough samples to output the optimal arm with confidence $1-\eta$, i.e. $\forall n \in B_{\delta,\eta}$, $T_n \geq T(\eta)$. Thus, with probability at most δ:

$$T(\eta) - P_{\mathcal{N}}(x = n)T \leq T_n - P_{\mathcal{N}}(x = n)T \leq -\sqrt{\frac{T}{2}\log\frac{1}{\delta}} \ . \tag{3.3}$$

This holds for every agent $n \in B_{\delta,\eta}$, and in particular for n such that $P_{\mathcal{N}}(x = n') = \min_{n \in B_{\delta,\eta}} P_{\mathcal{N}}(x = n) := p_{\delta,\eta}$.

Therefore, the total number of samples T is such that:

$$p_{\delta,\eta}T - \sqrt{\frac{1}{2}\log\frac{1}{\delta}}\sqrt{T} - T(\eta) \geq 0 \ , \tag{3.4}$$

which is a second order polynomial equation in $\sqrt{T}$. Solving this equation in T, we have with probability at least $1-\delta$:

$$T \geq \frac{1}{4p_{\delta,\eta}^2}\left(\sqrt{\frac{1}{2}\log\frac{1}{\delta}} + \sqrt{\frac{1}{2}\log\frac{1}{\delta} + 4p_{\delta,\eta}T(\eta)}\right)^2 \ . \tag{3.5}$$

Developing and re-ordering the terms, we have:

$$T \geq \frac{1}{p_{\delta,\eta}}T(\eta) + \frac{1}{4p_{\delta,\eta}^2}\left(\log\frac{1}{\delta} + \sqrt{\log^2\frac{1}{\delta} + 8\log\frac{1}{\delta}p_{\delta,\eta}T(\eta)}\right) \ . \tag{3.6}$$

In particular:

$$T \geq \frac{1}{p_{\delta,\eta}}\left(T(\eta) + \frac{1}{4p_{\delta,\eta}}\log\frac{1}{\delta}\right) \ . \tag{3.7}$$

Therefore, when E_1 does not occur, that is when all M most frequent agents have enough samples to output best arm with confidence η, with probability at least $1-\delta$:

$$T \leq \frac{1}{p_{\delta,\eta}}\left(T(\eta) + \frac{1}{4p_{\delta,\eta}}\log\frac{1}{\delta}\right) \ . \tag{3.8}$$

Let now $\mathcal{N}_M$ be the set of the M most likely agents and define $n^\star = \arg\min_{n\in\mathcal{N}_M} P_{\mathcal{N}}(x = n)$ and $p^\star = \operatorname{argmin}_{n\in\mathcal{N}_M} P_{\mathcal{N}}(x = n)$ as the index and the probability of the M-th most likely agent, respectively.

We now consider the following event: $E_2 = \{n^\star \notin \mathcal{B}_{\delta,\eta}\}$. Under $\neg E_1$, by definition of $\mathcal{B}_{\delta,\eta}$, E_2 is equivalent to the event $\{T_{n^\star} < T(\eta)\}$.

But:

$$\mathbb{P}(E_2) = \mathbf{I}_{1-p^\star}(T - T(\eta), 1 + T(\eta)) \ . \tag{3.9}$$

In order to have exactly $p_{\delta,\eta} = p^\star$, an equivalent formulation of $\neg E_2$ should hold for all agents whose sampling probability $P_{\mathcal{N}}(x = n)$ is greater than $p^\star$. This is why $p_{\delta,\eta} = p^\star$ with probability $\left(1 - \mathbf{I}_{1-p^\star}(T - T(\eta), 1 + T(\eta))\right)^M$.

□

Now, we prove Proposition 3.4.3 using Lemma 3.4.4.

Proof. (Proposition 3.4.3)

From Sajed and Sheffet [2019] (Lemma 4.2), the sample complexity of DP Successive Elimination is:

$$T(\beta) = \log(K/\beta) + \log\log(1/\Delta_2)\left(\frac{1}{\Delta_2^2} + \frac{1}{\epsilon\Delta_2}\right) \ .$$

Using the same global elimination mechanism, we can see DP-MASE as the multi-agent algorithm from Féraud et al. [2019] with DP Successive Elimination as a BAI subroutine. We can therefore apply Lemma 3.4.4 with local sample complexity $T(\beta)$ and conclude that the sample complexity of DP-MASE is:

$$\mathcal{O}\left(\frac{1}{p^\star}T(\beta) + \frac{1}{4p^{\star 2}}\log\frac{1}{\delta}\right)$$

with probability at least $(1-\delta)\left(1 - I_{1-p^\star}\left(T - T(\beta), 1 + T(\beta)\right)\right)^M$.

□

In Proposition 3.4.3, $T(\beta)$ is the local sample complexity, that is the number of steps a single-agent running DP Successive Elimination must execute to output the best arm with confidence $1-\beta$. $T(\beta)$ naturally increases with the confidence and the strength of the privacy guarantees. It also depends negatively on the smallest suboptimality gap Δ_2, which quantifies the hardness of the BAI problem. The global sample complexity of DP-MASE obviously scales with the local sample complexity. However, it also depends on the agent distribution $P_{\mathcal{N}}$, and particularly on the inverse probability of the M-th most likely player: through the global elimination process, the algorithm roughly "waits" for the M-th most frequent player to output its best arm estimate.

DP-MASE is of practical interest when rewards may carry sensitive information and when it is impossible to completely secure the communication channels. However, we could be concerned about withholding other information, such as the optimal option. In this situation, we consider that DP is not the most adapted notion. In the next section, we therefore propose another method based on multi-agent successive elimination that better fits this objective.

3.5 Best Arm Privacy in Multi-Agent BAI

Here, we are no longer concerned with preserving the input data, but rather want to protect the learned optimal option. We believe that the most suitable privacy notion for this problem is (α, η)-privacy, first introduced in Féraud et al. [2019]. In this section, we first discuss existing solutions and later introduce a new, more efficient multi-agent BAI algorithm that

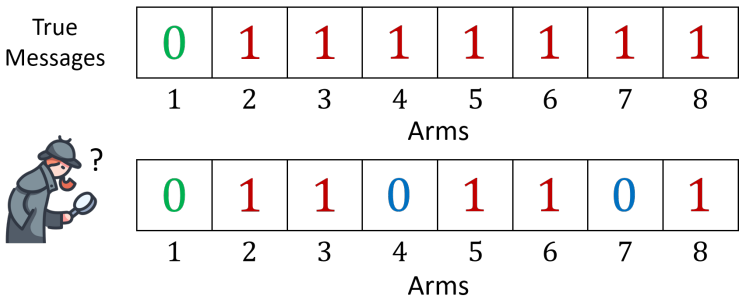


Figure 3.4: Illustration of Corrupted Elimination.

guarantees (α, η)-privacy.

3.5.1 Multi-Agent Successive Elimination with (α, η)-Private Communications

To address this setting, Féraud et al. [2019] propose a (α, η)-private algorithm based on multi-agent successive elimination. To reach (α, η)-privacy, each local arm selection subroutine is run with precision α and confidence $1 - \eta$. Such low local confidence implies that observing a single agent does not bring any useful information, making the algorithm (α, η)-private, while the consensus required to eliminate an arm globally ensures a low failure probability. More precisely, setting the global elimination threshold as $M = \lceil \frac{\log \delta}{\log \eta} \rceil$ ensures that an α-best arm is returned with a probability higher than $1 - \delta$.

Moreover, upper bounds on the sample complexity are derived that directly depend on the local subroutine complexity and the agent probability distribution $P_{\mathcal{N}}$. In particular, for any $(1 - \eta)$-confident BAI subroutine with sample complexity $T(\eta)$, the global sample complexity scales with $\frac{1}{p^\star} T(\eta)$, where $p^\star$ is the probability of the M-th most likely agent, similarly to DP-MASE.

3.5.2 Improved Privacy with Message Corruption

The privacy mechanism in Féraud et al. [2019] inherently links the local confidence level with the privacy level. Indeed, the algorithm is private in the sense of Definition 3.3.6 only because each independent participant is inaccurate in his best arm identification, returning an α-optimal arm with confidence as low as $1 - \eta$. This is likely to harm sample complexity: to guarantee high global accuracy, the voting threshold M must be high to compensate for the low local BAI confidence. In this section, we provide a method that decouples global privacy and local accuracy, providing more flexibility to adjust these two essential aspects of the problem and leading to better sample complexity.

Algorithm 5 CORRUPTED ELIMINATION

Input: Privacy level η, global failure probability δ, corruption probability ξ, voting threshold $M = \lceil \frac{\log \delta}{\log \eta} \rceil$
Output: Estimated best arm in $\mathcal{K}$
Global and local active sets $\mathcal{K} = \mathcal{K}^n = [K]$; local failure probability $\eta_\xi = 1 - \frac{1-\eta'}{(1-\xi)^{K-1}}$, $\mathcal{K}^n := \mathcal{K}$ for all $n \in \mathcal{N}$; global votes $(\lambda_k = 0)_{k\in[K]}$, global time $t = 0$
while $|\mathcal{K}| > 1$ **do**
  Active agent n is sampled: $n \sim P_{\mathcal{N}}$
  Sample all arms in $\mathcal{K}^n$ and update mean estimates
  $\bar{k}, , \bar{k}_G = \text{LocalElimination}\,(n, \xi)$ {Identify arms to eliminate locally}
  $\mathcal{K}^n = \mathcal{K}^n \setminus \bar{k}$ {Remove eliminated arms from local active set}
  $\lambda_k := \lambda_k + 1$ for any $k \in \bar{k}_G(e^n)$ {Vote for global elimination of eliminated arms if corresponding message not corrupted}
  if $|\mathcal{K}^n| = 1$ **then**
    $\mathcal{N} \leftarrow \mathcal{N} \setminus \{n\}$
  end if
  $\mathcal{K}^n := \mathcal{K}^n \bigcap \mathcal{K}$ {Remove the latest globally eliminated arms from local active set}
  for $k \in \bar{k}$ **do**
    $\mathcal{K} := \mathcal{K} \setminus \{k\}$ **if** $\lambda_k > M$
  end for
  $t := t + 1$
end while

Our method to guarantee a high privacy level without sacrificing local BAI confidence comes from the observation that the adversary is only able to derive critical information from sent messages and has no access to the internal state of the observed agent. Thus, instead of guaranteeing privacy by having poor local best arm identification, we limit the information that could be extracted from the sent messages by corrupting them with a random binary noise, as illustrated in Figure 3.4. More formally, instead of sending $\mathcal{M}_n$, agent n sends the corrupted messages $\mathcal{M}_n^\xi = (\lambda_1^n \times \nu_1^n, ..., \lambda_k^n \times \nu_K^n)$, where $\xi > 0$ is the corruption probability and $(\nu_k^n)_{k\in[K]}$ is the corruption mask such that $\nu_k^n \sim \mathcal{B}(1-\xi)$ for any k. We can then apply successive elimination. Algorithm 5 describes the whole procedure, which uses the local elimination subroutine described in Algorithm 6.

We now establish the privacy guarantees and the failure probability of Algorithm 5.

Proposition 3.5.1. Privacy of CORRUPTED ELIMINATION. *Running local BAI subroutines with confidence $1-\eta'$, an adversary knowing $\mathcal{A}$ and observing $\mathcal{M}_n^\xi$ cannot infer the best arm of agent n with probability higher than $(1-\eta') \times (1-\xi)^{K-1}$. Therefore, to maintain an apparent privacy level η, we need to set the confidence level of the BAI subroutine as:*

$$\eta_\xi = \max\left(0, 1 - \frac{1-\eta}{(1-\xi)^{K-1}}\right) \leq \eta \ .$$

Algorithm 6 CORRUPTED ELIMINATION Local Elimination

Input: Active agent n, corruption probability ξ
Output: A set of arms $\bar{k}$ to eliminate locally, and a set of arms $\bar{k}_G$ to vote against for global elimination
$\bar{k} = \{\}$, $\bar{k}_G = \{\}$
for all $k \in \mathcal{K}^n$ **do**
  if $\max_{l \in \mathcal{K}^n} \hat{\mu}_l^n(t) - \hat{\mu}_k^n(t) \geq 2\sqrt{\frac{\log\left(Kt^{n2}/\eta_\xi\right)}{t^n}}$ **then**
    $\bar{k} := \bar{k} \bigcup \{k\}$ {Add k to the set of arms to eliminate if elimination condition is met}
    Draw $\nu_k^n \sim \mathcal{B}(1-\xi)$
    if $\nu_k^n = 1$ **then**
      $\bar{k}_G := \bar{k}_G \cup \{k\}$ {If message is not corrupted, add k to the set of arms to vote against for global elim.}
    end if
  end if
end for

Proof. Let us recall the definition of $\mathcal{M}_n$ and $\mathcal{M}_n^\xi$:

$$\mathcal{M}_n = (\lambda_1^n, ..., \lambda_K^n) \quad , \tag{3.10}$$
$$\mathcal{M}_n^\xi = (\lambda_1^n \times \nu_1^n, ..., \lambda_k^n \times \nu_K^n) \quad , \tag{3.11}$$

where, for any arm k, ν_k^n is the corruption mask, *i.e.*, a random variable with distribution $\mathcal{B}(1-\xi)$.

An adversary can guess the global optimal arm from messages $\mathcal{M}_n$ if these two events hold:

- Event A: the arm selection subroutine of agent n actually finds the optimal arm;
- Conditionally to A, event B: no message associated to sub-optimal arms is corrupted.

By assumption, $\mathbb{P}(A) \geq 1 - \eta_\xi$. Moreover, B holds when all ν_k^n are ones for $k = 2, ..., K$. Since $\nu_k^n \sim \mathcal{B}(1-\xi)$, we have $\mathbb{P}(B) = (1-\xi)^{K-1}$. Therefore, observing messages $\mathcal{M}_n$, an external observer can only infer the best arm with probability $(1-\eta_\xi) \times (1-\xi)^{K-1}$.

To maintain an apparent privacy level η, we then need to set $\eta_\xi > 0$ such as:

$$1 - \eta = (1-\eta_\xi) \times (1-\xi)^{K-1} \quad . \tag{3.12}$$

That is:

$$\eta_\xi = \max\left(0, 1 - \frac{1-\eta}{(1-\xi)^{K-1}}\right) \leq \eta \quad . \tag{3.13}$$

□

Proposition 3.5.2. Failure probability of CORRUPTED ELIMINATION. *The failure proba-*

bility of Corrupted Elimination *is at most:*

$$\min_{\alpha\in\mathbb{N}} \left(1 - I_\xi\left(\alpha+1, M\right)\right) \times \delta_\alpha \ , \ \textit{where } \delta_\alpha = \delta \times {\eta_\xi}^\alpha \ .$$

Proof. Assuming it terminates, Corrupted Elimination fails, *i.e.*, eliminates the best arm, if there are at least $M = \lceil \frac{\log \delta}{\log \eta} \rceil$ votes against it. Let N_e and N_v denote respectively the number of local independent eliminations of the best arm and the number of votes actually sent against the best arm. N_v follows a binomial distribution with N_e trials:

$$N_v \sim \mathcal{B}\left(N_e, 1-\xi\right) \ . \tag{3.14}$$

Given N_e independent eliminations, the probability that N_v exceeds the voting threshold M is:

$$\mathbb{P}\left(N_v \geq M\right) = \sum_{m=M}^{N_e} \binom{N_e}{m} (1-\xi)^m \xi^{N_e - m} \ . \tag{3.15}$$

In order to overcome the randomness of N_e, we want $\alpha \in \mathbb{N}$ such that if $N_e \geq M + \alpha$, $N_v \geq M$ with high probability. Let $\alpha \in \mathbb{N}$ such that $N_e \geq M + \alpha$. Then, the probability of a random variable X following $\mathcal{B}\left(M+\alpha, 1-\xi\right)$ being greater than M is lower than if X followed $\mathcal{B}\left(N_e, 1-\xi\right)$. Thus:

$$\begin{aligned}
\mathbb{P}\left(N_v \geq M\right) &\geq \sum_{m=M}^{M+\alpha} \binom{M+\alpha}{m} (1-\xi)^m \xi^{M+\alpha-m} && (3.16)\\
&= 1 - \sum_{m=0}^{M-1} \binom{M+\alpha}{m} (1-\xi)^m \xi^{M+\alpha-m} && (3.17)\\
&= 1 - I_\xi\left(\alpha+1, M\right) \ , && (3.18)
\end{aligned}$$

where $I_p\left(a, b\right)$ is the regularized incomplete beta function evaluated at p with parameters (a, b).

Now since the local BAI routines are run with confidence η_ξ, the probability that at least $M + \alpha$ independent agents eliminate the best arm is at most $\eta_\xi^{M+\alpha}$. Eventually, for $\alpha \in \mathbb{N}$, Corrupted Elimination eliminates the best arm if the two following events happen:

- $M + \alpha$ independent agents eliminate the best arm locally, which happens with probability at most $\eta_\xi^{M+\alpha}$;
- These $M+\alpha$ local eliminations imply at least M votes, which happens with probability $1 - I_\xi\left(\alpha+1, M\right)$.

By independence of the corruption mechanism, the failure probability of the algorithm is at most:

$$\min_{\alpha\in\mathbb{N}} \left(1 - I_\xi\left(\alpha+1, M\right)\right) \times \eta_\xi^{M+\alpha} \ . \tag{3.19}$$

Noticing that:

$$\eta_\xi^{M+\alpha} = \eta_\xi^{\lceil \frac{\log \delta}{\log \eta} \rceil + \alpha} \leq \delta \times \eta_\xi^\alpha = \eta_\xi^{\frac{\log \delta}{\log \eta} + \alpha} \ , \tag{3.20}$$

and setting $\delta_\alpha = \delta \times {\eta_\xi}^\alpha$ concludes the proof.

□

Proposition 3.5.1 tells us that we can run the local subroutines with better accuracy than Féraud et al. [2019] while keeping the same privacy guarantees. With Proposition 3.5.2, failure probability is less than without communication noise. Indeed, since some messages are corrupted, more independent agents need to vote against the best arm to remove it globally. In Proposition 3.5.3, we propose an approximation of the sample complexity of Corrupted Elimination.

Proposition 3.5.3. Sample complexity of Corrupted Elimination. *If the agent distribution $P_\mathcal{N}$ is uniform, then with probability α, the number of rounds needed by Corrupted Elimination to output an α-optimal arm is approximately:*

$$T = N \times Q^{-1}\left(T(\eta_\xi), 1 - \alpha^{1/M}\right) \ ,$$

where Q^{-1} is the inverse of the regularized gamma function, and $T(\eta_\xi)$ is the number of local samples needed for the BAI subroutine to find the optimal arm with confidence $1 - \eta_\xi$.

Proof. For all $n \in \mathcal{N}$, let T_n denote the number of rounds agent n has been active, and T denote the total number of rounds before the algorithm stops. Then $(T_1, ..., T_N)$ follows a multinomial distribution:

$$(T_1, ..., T_N) \sim \text{Mult}(T, \{p_1, ..., p_N\}) \ , \tag{3.21}$$

where $p_n = P_\mathcal{N}(x = n)$. For simplicity, we consider a uniform distribution, i.e. $p_n = 1/N$ for any $n \in \mathcal{N}$, in the sequel of this proof.

Assuming the local BAI subroutines accurately return the best arm, the algorithm terminates when the M largest $T_1, ..., T_N$ exceed threshold $T(\eta_\xi)$. We thus look for a bound on the probability of this event.

We can make the components of the vector $(T_1, ..., T_N)$ independent by "poissonizing" the multinomial, using the following theorem [DasGupta, 2011]:

Theorem 3.5.4. *[DasGupta, 2011]. Let $N \sim \mathcal{P}(\lambda)$ and suppose $(X_1, ..., X_k) \sim$ Multi$(p_1, ..., p_k)$. Then, marginally, $X_1, ..., X_K$ are independent Poisson random variables, with $X_i \sim \mathcal{P}(p_i\lambda)$.*

Instead of considering T, the number of trials of the multinomial, as fixed, we assume that it follows a Poisson distribution, i.e., $T \sim \mathcal{P}(\overline{T})$ for some $\overline{T} \in \mathbb{N}$. Thus, we can now think of $T_1, ..., T_N$ as mutually independent random variables such that $T_n \sim \mathcal{P}\left(\overline{T}/N\right)$ for all $n \in \mathcal{N}$.

The probability that one count T_n is greater than $T(\eta_\xi)$ can then be expressed in terms of the cumulative distribution function of the Poisson distribution:

$$\mathbb{P}(T_n \geq T(\eta_\xi)) = 1 - Q\left(T(\eta_\xi), \overline{T}/N\right) \ , \tag{3.22}$$

where $Q(s, x)$ is the regularized gamma function evaluated in (s, x).

Consequently, by independence of the counts $T_1, ..., T_N$, the probability that M different agents have more samples than $T(\eta_\xi)$ at time t is, for some $\alpha \in [0, 1]$:

$$(1 - Q\left(T(\eta_\xi), \overline{T}/N\right)^M = \alpha \ .$$

Now, to estimate the sample complexity T, we may just compute $\overline{T}$ given α, M and $T(\eta_\xi)$. To do this, we notice that:

$$Q(T(\eta_\xi), \overline{T}/N) = 1 - \alpha^{1/M} \ .$$

There is no close form for the inverse of Q that would lead to a close form for T. However, we can numerically compute the inverse of Q in its second argument. Thus, the total number of samples would be given by:

$$\overline{T} \approx N \times Q^{-1}\left(T(\eta_\xi), 1 - \alpha^{1/M}\right) \ .$$

Confusing T with its expectation $\overline{T}$, we get an approximation of the sample complexity of Corrupted Elimination.

□

We observe empirically (see Section 3.6) that there is a positive correlation between the global sample complexity and the local failure probability η_ξ (corresponding exactly to the privacy level when $\xi = 0$), even with uniform sampling. This observation was not reflected in the initial bound from Féraud et al. [2019], as increasing BAI confidence $1 - \eta_\xi$ means more local samples $T(\eta_\xi)$ are needed (see for instance the bound for Successive Elimination). However, if the approximation provided in Proposition 3.5.3 is not enough to conclude that the sample complexity increases with η_ξ (because of the first parameter in Q^{-1}), it shows at least an ambiguity. Indeed, when η_ξ increases, $M = \lceil \frac{\log \delta}{\log \eta_\xi} \rceil$ increases, and therefore $\alpha^{1/M}$ increases (as $0 < \alpha < 1$). Empirically, we also observe that Q^{-1} is a decreasing function in its second argument. Therefore, although the final effect is unclear due to the presence of $T(\eta_\xi)$ as a first argument, the sample complexity possibly increases with η_ξ *via* its second argument. We refer to this as the *threshold effect*. Backed by various empirical results, we further develop this argument in the upcoming section.

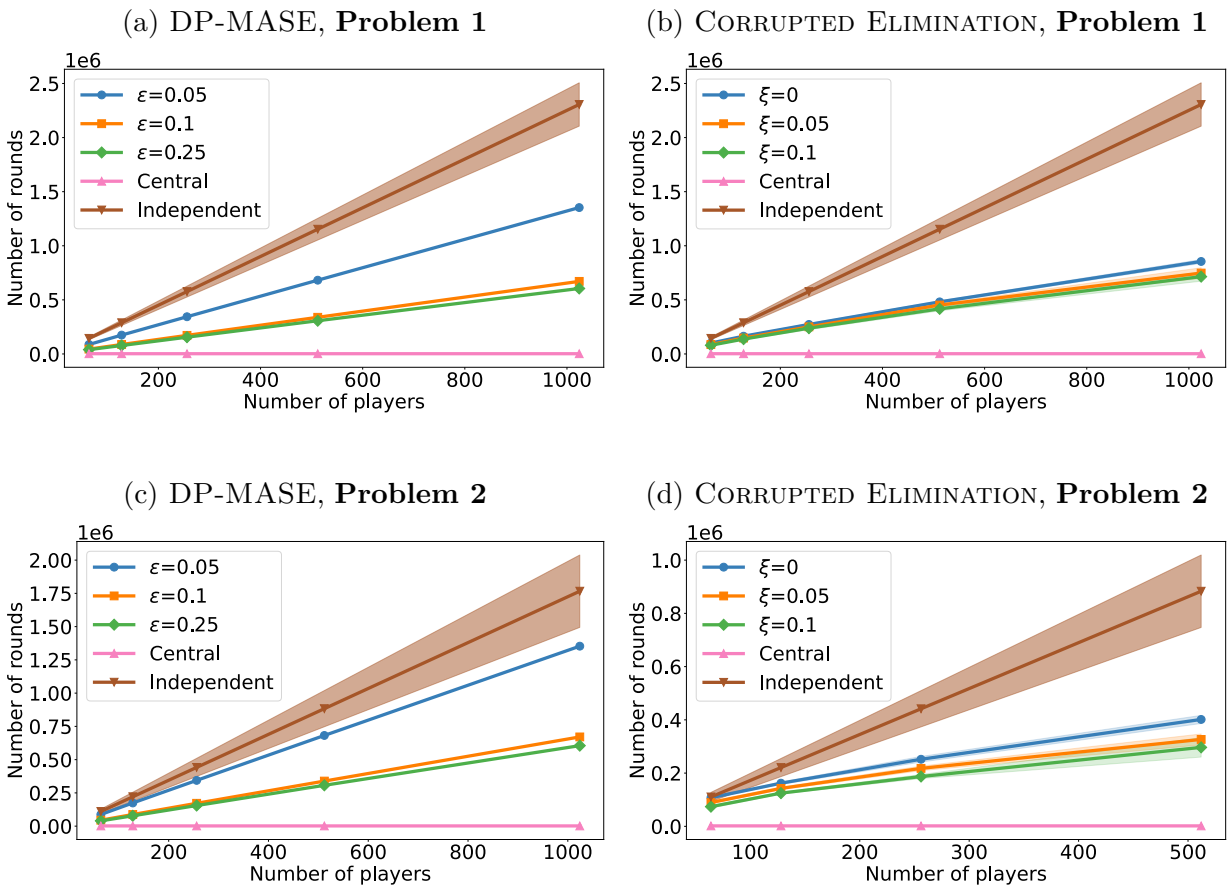


Figure 3.5: Sample complexity of DP-MASE (left) for $\epsilon = 0.05, 0.1, 0.25$ and Corrupted Elimination (right) for $\xi = 0, 0.05, 0.1$ on **Problem 1** (*top*) and **Problem 2** (*bottom*).

3.6 Experiments

3.6.1 Experimental Setup

Environments. We assess the performance of our methods on different stochastic environments suggested in Féraud et al. [2019]. In these problems, there are $K = 10$ arms with means $\mu_1 \geq \ldots \geq \mu_{10}$. In **Problem 1**, rewards are drawn from Bernoulli distributions, with $\mu_1 = 0.7$, $\mu_2 = 0.5$, $\mu_3 = 0.3$, and $\mu_k = 0.1$ for $k = 4, \ldots, 10$. In **Problem 2**, arms have Gaussian distributions and means $\mu_1 = 11$, $\mu_2 = 10.8$, $\mu_k = 10.4$ for $k = 3, \ldots, 10$. Agent distribution $P_{\mathcal{N}}$ is uniform. We run experiments with N ranging from 64 to 1024 agents to evaluate how performance is affected by the scale of the problem. For both methods, we fix the global failure probability δ to 5%. Each data point is an average value over 10 runs, and 95% confidence intervals are shown for every plot.

Evaluation and Baselines. We evaluate the performance of our algorithms for different values of the privacy parameters (ϵ for DP-MASE, η and ξ for Corrupted Elimination). In addition, we compare them with two baselines, corresponding to the extreme cases with full and no communication.

- Central: the agents send all their raw reward samples with the central coordinator at each round, which is equivalent (up to negligible communication delays) to a single-agent Successive Elimination. This approach does not offer any privacy since data is fully shared;
- Independent: every agent runs an independent Successive Elimination algorithm, without sharing any information with the central coordinator. This approach is fully private as there is no communication.

They allow to illustrate both the gain in performance achieved through multi-agent collaboration and the challenges of distributing data across multiple nodes. To measure performance, we report the sample complexity, which is the total number of rounds (corresponding to a single iteration of the while loop in Algorithms 2, 3 and 5) needed to output the best arm with confidence $1 - \delta$.

3.6.2 Empirical Findings

3.6.2.1 Impact of Multi-Agent Collaboration

Figure 3.5 shows the performance of our methods against the Central and Independent baselines. We observe that the BAI problem is solved much faster if agents can freely communicate their raw reward samples, which amounts to the single-agent case. This is expected, as the problem is inevitably harder when the data is distributed to multiple locations. In particular, the sample complexity must scale with the number of agents, as already discussed in Section 3.3.3. We indeed observe a growth in sample complexity with the number of agents for both DP-MASE and Corrupted Elimination. However, they scale better than the naive Independent baseline, which shows that the communication between agents and the collaborative elimination procedure help to be more efficient and attenuate the growth in sample complexity.

3.6.2.2 Sample Complexity and Privacy Level

Figure 3.5 (left) exhibits the influence of the privacy level ϵ on the performance of DP-MASE. Unsurprisingly, for a fixed global confidence $1-\delta$, the sample complexity gets worse with stronger privacy guarantees, i.e., smaller ϵ's. Longer epochs and larger gap thresholds for local elimination are indeed needed to compensate for the noisier mean estimates. We notice a significant increase in sample complexity for small values of ϵ (less than 0.1), while it is relatively stable for larger values.

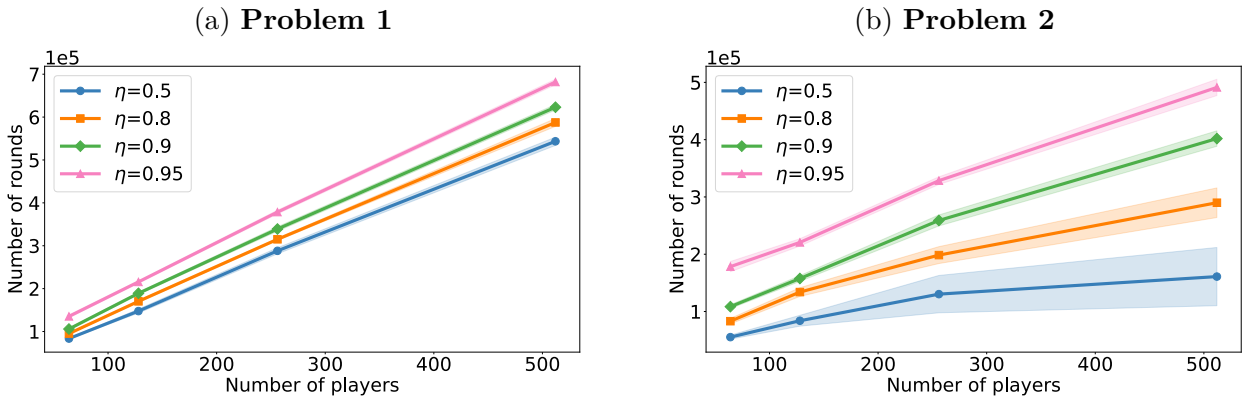


Figure 3.6: Sample complexity of Féraud et al. [2019] ($\xi = 0$) as a function of η

For Corrupted Elimination, we set $\xi = 0$ (corresponding to Féraud et al. [2019]) and compare the sample complexity at different privacy levels η (Figure 3.6). Results show that the sample complexity also increases as the apparent privacy level η grows. This behavior is not reflected in the initial sample complexity bound stated in Féraud et al. [2019]. While Proposition 3.5.3 provides a first theoretical insight, here we explain this behavior qualitatively, by first stressing the implications of higher local accuracy in terms of sample complexity. More confident local agents means that 1) more local samples are needed to estimate the best arm, and 2) less independent eliminations are needed to reach a given global accuracy, hence a smaller voting threshold M. The former effect will tend to increase the sample complexity, while the latter (the *threshold effect*) will tend to decrease the sample complexity. Based on our experiments, we therefore argue that the *threshold effect* is predominant in the case of Corrupted Elimination.

3.6.2.3 Comparison between Corrupted Elimination and Féraud et al. [2019]

We now compare the sample complexity of Corrupted Elimination against the method proposed by Féraud et al. [2019] (corresponding to $\xi = 0$). The apparent privacy level η is set to 0.9. In accordance with Proposition 3.5.1, for each value of ξ, we compute $\eta_\xi = 1 - \frac{1-\eta}{(1-\xi)^{K-1}}$, and then run local BAI subroutines with confidence $1 - \eta_\xi$, which guarantees the same apparent privacy level and hence a fair comparison.

We observe that corrupting messages actually decreases the sample complexity of Corrupted Elimination, and that it tends to further decrease with higher corruption probability ξ (Figure 3.5, right). In particular, we improve over the method proposed by Féraud et al. [2019]. We also attribute this behavior to the *threshold effect* discussed above. Indeed, ξ is positively correlated with the local accuracy $1 - \eta_\xi$ for a given apparent privacy level η. In other words, larger corruption probability means we run more confident local BAI without compromising the global privacy of the best arm. The observed effect of ξ on sample complexity thus again suggests that the reduced threshold M prevails over the resulting increase in the number of local samples.

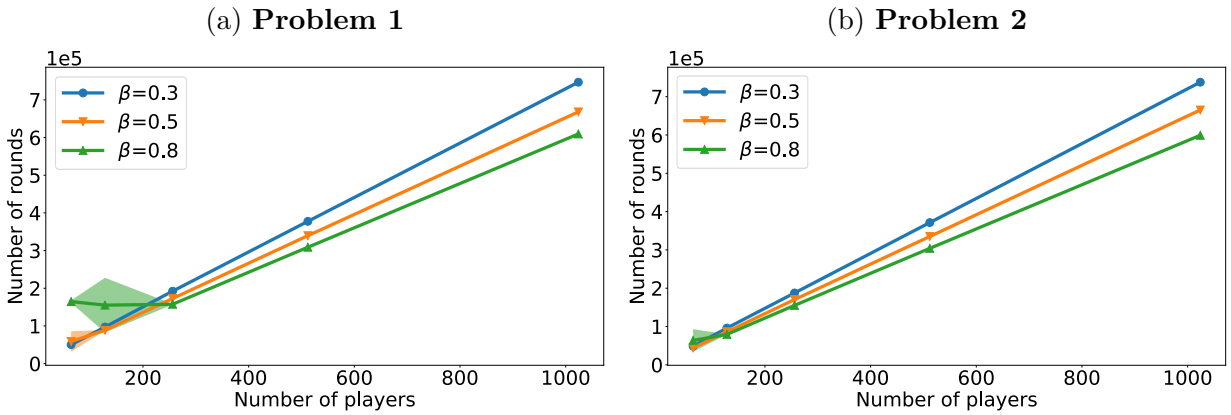


Figure 3.7: Sample complexity of DP-MASE for $\beta = 0.3, 0.5, 0.8$.

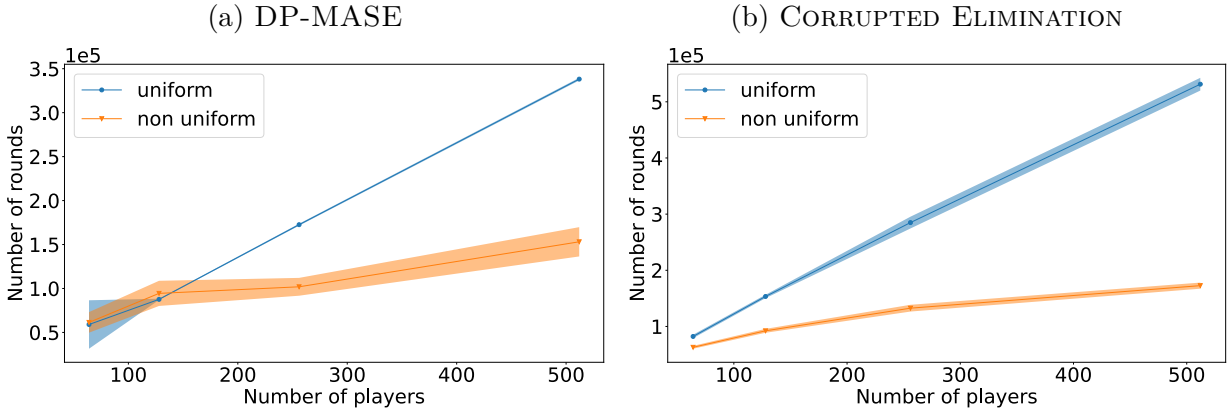


Figure 3.8: Sample complexity of DP-MASE (left) and Corrupted Elimination (right) on **Problem 1** for uniform and non-uniform agent distributions.

3.6.2.4 Sample Complexity and Local Accuracy for DP-MASE

Given a fixed privacy level $\epsilon = 0.5$, Figure 3.7 depicts the sample complexity of DP-MASE for different local accuracy levels β. We observe that the sample complexity increases with local accuracy. This is in contradiction with what we observe for Corrupted Elimination, and suggests that the increase in local samples associated with higher local accuracy prevails over the threshold effect for DP-MASE. Our intuition is that, as DP-MASE works in epochs, it is likely that all agents will reach the end of a given epoch and proceed to an elimination at about the same time — at least with uniform agent sampling. In Corrupted Elimination, each agent may vote for the global elimination of an arm at any time — not only at the end of an epoch, so that it is more likely to gather M votes against an arm early in the process. In terms of speeding up the elimination of suboptimal arms, the global voting process is thus more important in Corrupted Elimination than in DP-MASE.

3.6.2.5 Sample Complexity and Agent Distribution

In previous experiments, we assumed $P_{\mathcal{N}}$ was uniform. However, Proposition 3.4.3 suggests that $P_{\mathcal{N}}$ affects the sample complexity in the Multi-Agent Successive Elimination scheme. Indeed, through the global elimination process, the algorithm roughly waits for the M-th most frequent player to output its best arm estimate. Therefore, we expect DP-MASE and Corrupted Elimination to be faster for unbalanced distributions, without losing accuracy, and indeed observe this behavior empirically. To simulate an unbalanced agent distribution, we choose the following:

$$P_{\mathcal{N}}^{\gamma}(n) \propto (n+\alpha)^{-\gamma} \ ,$$

for some $\alpha \geq 0$ and $\gamma \in (0,1)$. Thus, the probability of sampling agent n decreases with n.

We illustrate the impact of $P_{\mathcal{N}}$ on the sample complexity for both DP-MASE and Corrupted Elimination, comparing the uniform distribution to $P_{\mathcal{N}}^{\gamma}$ with $\gamma = 0.8$. We choose $\epsilon = 0.1$ and $\beta = 0.5$ for DP-MASE, and set $\xi = 0.05$ for Corrupted Elimination. As expected, both methods are much faster when the agent distribution is unbalanced.

3.7 Chapter Summary

In this chapter, we have proposed two methods to address multi-agent best arm identification, where multiple participants work together to identify with fixed confidence the best option in a common multi-armed bandit. Based on the same successive elimination scheme, each method guarantees the privacy of a type of data against adversaries observing the communications. Direct comparison between our two methods is impossible, as they rely on distinct privacy definitions. However, they are suitable for addressing complementary privacy concerns in multi-agent best arm identification. DP-MASE ensures that the communications are differentially private with respect to the rewards, while Corrupted Elimination prevents the adversaries from inferring the best arm. We provide theoretical analysis on privacy, sample complexity, and failure probability, and experiments on toy problems give insights into the behavior of our methods. In particular, they show that Corrupted Elimination is faster than a similar method without message corruption, for comparable privacy levels. Since our primary goal was to investigate and address different privacy needs within the relatively new setting of collaborative BAI, our methods could be refined to achieve better sample complexity.

Chapter Four

Privacy-Preserving Deep Reinforcement Learning

While private multi-armed bandits, studied in the previous chapter, is a valuable framework for privacy-preserving sequential decision-making, it cannot handle problems where decisions impact the system dynamics over time, such as the control of advanced systems (*e.g.*, autonomous vehicles) or the post-training of LLMs. It is therefore essential to develop privacy-preserving methods for the more general reinforcement learning framework, which is our second contribution presented in this chapter.

Contrary to differentially private bandits, which have received a lot of attention in the literature, differentially private RL has remained relatively unexplored, and existing work is mainly theoretical. In particular, current solutions either focus on restrictive privacy notions or cannot apply to standard deep reinforcement learning benchmarks, let alone scale to complex, high-dimensional real-world problems. This is at odds with the growing use of reinforcement learning in real-world applications involving sensitive data, as well as the increasing evidence of privacy vulnerabilities in RL algorithms [Pan et al., 2019, Gomrokchi et al., 2023, Prakash et al., 2022], all highlighting the urgent need to develop general, scalable privacy-preserving methods for this setting.

In response, this chapter introduces RL methods with formal differential privacy guarantees applicable to deep reinforcement learning, laying the foundation for the deployment of private RL agents in the real world. It is structured into two main parts. The first part addresses offline reinforcement learning, a practically relevant setting for companies willing to train RL agents on existing sensitive data without costly interaction with real systems. The second part focuses on the design of differentially private policy gradient methods, whose potential applications include the fine-tuning of LLMs on private data.

The research presented in this chapter has led to a paper presented at the *European Workshop on Reinforcement Learning 2024* [Rio et al., 2024] and under review at *IJCAI 2025*, as well as a preprint [Rio et al., 2025], currently under review at *ICML 2025*.

4.1 Context

In the past years, Reinforcement Learning (RL) agents have been increasingly deployed in the real world, spanning various domains such as healthcare [Liu et al., 2022], recommendation engines [Afsar et al., 2023], and finance [Hambly et al., 2021], or to control advanced systems, for instance, autonomous vehicles [Wang et al., 2023] and robotics [Tang et al., 2024]. RL has also greatly contributed to the huge success of LLMs, being used to align these models with human preferences (*i.e.*, RL from Human Feedback (RLHF), see Kaufmann et al. [2023]) and to enhance their reasoning capabilities [DeepSeek-AI, 2025]. To succeed, these RL agents are typically trained on a large amount of personal data.

As a consequence, there has been growing concern over privacy leakage from RL agents. Indeed, empirical evidence suggests that RL is no more immune to privacy attacks than other areas of ML. In particular, Gomrokchi et al. [2023] exploit temporal data correlation in RL to design powerful membership inference attacks (MIAs) against RL policies in the black-box setting, using convolutional neural classifiers. Given access to the output policy, an adversary can learn to infer the presence of a specific trajectory — which is the result of a sequence of interactions between a user and the system — in the training dataset with great accuracy. Pan et al. [2019] and Prakash et al. [2022] have also designed privacy attacks against the transition dynamics and the reward function of the underlying Markov decision process (MDP). Moreover, several studies have shown evidence that privacy leakage can happen during the fine-tuning training phase of LLMs [Carlini et al., 2021, Fu et al., 2023], suggesting that RLHF on private data may be unsafe. The existence of such privacy threats is a huge issue as the targeted data can be highly sensitive in some scenarios. For instance, in RL-based personalized treatment recommendation, a trajectory can reveal information about a patient's health history, while systems for online advertising or news recommendation are trained on users' browsing history that can reveal the users' preferences on various topics (*e.g.*, politics). Moreover, companies typically fine-tune LLMs on valuable proprietary data to accomplish industry-specific tasks.

Designing privacy-preserving RL methods, notably based on differential privacy — the current gold standard in terms of privacy protection, particularly against MIAs — is therefore of great importance. Yet, while DP has been successfully applied in various domains, such as neural network training and multi-armed bandits, and despite RL facing similar privacy challenges, private RL has received comparatively little attention. In particular, no existing work has proposed a private RL method that matches the versatility, scalability, and empirical effectiveness of DP-SGD [Abadi et al., 2016] for supervised learning. A primary obstacle is the set of technical challenges specific to RL, such as the many ways of collecting data and the correlated nature of training samples resulting from online interactions [Fan et al., 2025], which makes it difficult to even formulate a universal and meaningful DP definition. As a result, existing research largely remains theoretical and demonstrates limited practical applicability. In the online setting, numerous private algorithms have been developed (*e.g.*, Vietri et al. [2020], Garcelon et al. [2021], Qiao and Wang [2023b]) but their

scope remains restricted to tabular and linear Markov decision processes (MDPs) with finite horizon. Qiao and Wang [2023a] have proposed the first private algorithms for offline RL, building on value iteration methods, but they present similar limitations. In harder settings, approaches such as Wang and Hegde [2019] and Cundy et al. [2024] focus on the protection of specific elements of the problem (for instance, the reward function). Therefore, existing algorithms either do not protect all sensitive information or cannot intrinsically scale to the problems typically encountered in deep RL and in the real world, leaving the aforementioned privacy threats unaddressed.

Consequently, we believe that there is a pressing need to steer the field of differentially private RL towards more practical, scalable approaches. This chapter thus focuses on the development of differentially private RL algorithms that are applicable to general MDPs, provide comprehensive privacy protection at the trajectory level, and are suitable for deep neural policies and value functions.

4.2 Contributions

After introducing useful background and definitions regarding DP reinforcement learning, and stating a few preliminary results (Section 4.4), this chapter introduces two distinct, practical approaches for the training of privacy-preserving RL agents. The first part (Section 4.5) addresses the offline RL setting, where we develop a deep model-based approach, while the second part (Section 4.6) deals with the online RL setting, for which we propose a DP policy gradient. We highlight the specific contributions from each chapter below.

In both settings, we use the notion of *trajectory-level* differential privacy (TDP) for reinforcement learning, first introduced in Qiao and Wang [2023a], which we deem more practical than *joint differential privacy*, the most used central DP notion from the RL literature (see Section 4.4). In particular, after identifying privacy vulnerabilities in states, actions, and rewards alike, we aim to protect all components of a trajectory. Guaranteeing privacy at the trajectory level precludes the use of vanilla DP-SGD, designed to provide privacy at the sample level (*i.e.*, at the transition level in RL), introducing significant technical challenges.

To the best of our knowledge, this chapter provides the first deep RL methods for general MDPs with differential privacy guarantees at the trajectory level. From an empirical perspective, we tackle previously unaddressed benchmarks in the DP RL literature, such as standard control tasks from Gym Control and MuJoCo, as well as the fine-tuning of LLMs using RLHF. Our approaches are free from the inherent limitations of existing methods and have the potential to scale to even more complex, higher-dimensional problems. We thus aim to start filling the huge gap between the current private RL literature and real-world applications, hence paving the way for the usage of private RL in complex, risk-sensitive scenarios.

4.2.1 Differentially Private Offline Reinforcement Learning

Offline RL [Levine et al., 2020], which refers to the set of methods enabling the training of control agents from static datasets, avoids online interaction with the system that may be impractical or hazardous in real-world scenarios, showing great promise for real-world applications. Also, it arguably offers a more natural framework for privacy compared to the classic online setting given its similarities with supervised learning. To tackle this setting, we introduce a model-based approach named PriMORL. Protecting entire trajectories, rather than individual examples, precludes the use of vanilla optimizers such as DP-SGD. Additionally, the standard approach of using bootstrap ensembles to handle model uncertainty presents an extra challenge in the private setting, as the ensemble size directly impacts the privacy budget. A key contribution of this work is thus the introduction of a training method for model ensembles that ensures differential privacy at the trajectory level and effectively controls the privacy budget. We also provide a theoretical analysis of how private training influences model reliability and then prove the formal privacy guarantees of the resulting policy. Empirically, we are the first to tackle standard control tasks with deep function approximation in the trajectory-level DP setting. Our experiments show that PriMORL can train deep DP RL agents with limited performance cost compared to the non-private baseline, providing a new standard for DP RL.

4.2.2 Differentially Private Policy Gradient

We observe that policy gradient methods [Sutton et al., 1999], although widely popular and achieving state-of-the-art results on various deep RL tasks, have not been studied in the private setting. Yet, directly optimizing the policy through gradient ascent, policy gradient methods exhibit clear similarities with classic deep learning optimization techniques like SGD. This similarity prompts exploring the use of clipped noisy gradient updates, which are at the core of DP-SGD, to derive differentially private versions of policy gradient algorithms. Moreover, policy gradient methods are at the core of RLHF for LLM alignment: given the current popularity of the topic and the associated privacy concerns, researching DP policy gradient methods seems highly warranted.

Further motivated by the apparent connection between gradient clipping in DP-SGD and trust-region approaches in RL, we investigate the subclass of policy gradient methods referred to as *trust-region methods*, which includes TRPO [Schulman et al., 2015] and PPO [Schulman et al., 2017]. These algorithms, extensively used for their stability and versatility, aim at preventing too large policy updates which can result in dramatic performance drop during training. In the presence of DP noise, we characterize the inclusion of the gradient updates within a given trust region and derive theoretical insights into how to set the DP clipping norm for the noisy updates to remain within the trust region with high probability, which we validate empirically on a discrete environment. Relaxing the theory, we then provide a practical PPO-like algorithm with formal, trajectory-level differential pri-

vacy guarantees. For realistic privacy budgets, our approach shows limited impact on the performance of policy gradient methods on previously unaddressed deep RL benchmarks, effectively paving the way for deploying DP RL agents in complex environments.

4.3 Related Work

Offline Model-Based RL. Offline RL [Levine et al., 2020] focuses on training agents without further interactions with the system, making it essential in scenarios where data collection is impractical (*e.g.*, Singh et al. [2022], Liu et al. [2020]). Model-based RL [Moerland et al., 2023] can further reduce costs or safety risks by using a learned environment model to simulate beyond the collected data and improve sample efficiency [Chua et al., 2018]. Argenson and Dulac-Arnold [2021] demonstrate that model-based offline planning, where the model is trained on a static dataset, performs well in robotic tasks. However, offline RL faces challenges like *distribution shift* [Fujimoto et al., 2019], where the limited coverage of the dataset can lead to inaccuracies in unexplored state-action regions, affecting performance. Methods like MOPO [Yu et al., 2020], MOReL [Kidambi et al., 2020], and Count-MORL [Kim and Oh, 2023] address this by penalizing rewards based on model uncertainty, achieving strong results on offline benchmarks. Still, key design choices in offline MBRL require further exploration, as highlighted by Lu et al. [2022].

Policy Gradient Methods in RL. In RL, policy gradient (PG) methods [Sutton et al., 1999] optimize the policy directly within its parameter space using the policy gradient theorem. This theorem enables gradient ascent by formulating the gradient of the expected discounted return with respect to the policy parameters. Many policy gradient algorithms have been proposed over the years to improve upon the vanilla, high-variance Reinforce [Williams, 1992], among which *actor-critic methods* [Mnih et al., 2016, Haarnoja et al., 2018], and *deterministic PG methods* [Silver et al., 2014, Lillicrap et al., 2016]. Among them, *trust-region methods* constrain policy updates while optimizing a surrogate loss. TRPO [Schulman et al., 2015], of which Natural Policy Gradient [Kakade, 2001] can be considered a special case, imposes a hard constraint on the KL divergence between consecutive policies. Its successor, PPO [Schulman et al., 2017], uses a relaxed constraint and has been widely popular for its ease of implementation and state-of-the-art performance on various tasks. Recently, PG methods like PPO have been successfully applied to aligning LLMs with human preferences through RLHF [Kaufmann et al., 2023].

Private RL. Differential Privacy (DP), established by Dwork [2006], has become the gold standard for privacy protection and has been the focus of extensive research over the past two decades. Notably, DP-SGD [Abadi et al., 2016] has facilitated the training of deep neural networks with DP guarantees, encouraging the deployment of private models at scale. However, many practical challenges remain [Ponomareva et al., 2023]. At the same time,

sophisticated attack strategies have underscored the necessity for robust DP algorithms [Rigaki and García, 2024]. Recent studies have shown that reinforcement learning (RL) is also vulnerable to privacy threats [Pan et al., 2019, Prakash et al., 2022, Gomrokchi et al., 2023]. As RL is increasingly deployed in the real world [den Hengst et al., 2020, Liu et al., 2022, Afsar et al., 2023, Wang et al., 2023], the need for privacy-preserving training techniques is critical. Although DP has been successfully extended to multi-armed bandits [Basu et al., 2019], the DP RL literature remains mainly theoretical. Indeed, existing RL algorithms [Vietri et al., 2020, Garcelon et al., 2021, Liao et al., 2021, Luyo et al., 2021, Chowdhury and Zhou, 2022, Zhou, 2022, Ngo et al., 2022, Qiao and Wang, 2023b], with formal DP guarantees mainly apply to episodic tabular or linear MDPs and lack empirical validation beyond basic simulations. Moreover, private offline RL remains underexplored. Only Qiao and Wang [2023a] have proposed DP offline algorithms, which, while theoretically strong, are also restricted to finite-horizon tabular and linear MDPs. A couple of works [Wang and Hegde, 2019, Cundy et al., 2024] have addressed more general settings and harder problems, albeit with more restrictive privacy notions. Indeed, Wang and Hegde [2019] study DP with respect to neighboring reward functions, and Cundy et al. [2024] seek to protect sensitive state variables from an information-theoretic perspective. Recently, Zhang et al. [2025] have addressed private reinforcement learning from human feedback, but their scope is limited to the protection of human preferences, and does not cover potentially sensitive information in the state (the prompt) and action (the prompt completion). Our work differs from these approaches in that it seeks to protect all elements of a trajectory, offering more comprehensive privacy protection. In this context, current DP RL methods are fundamentally unable to scale to the problems encountered in the real world; in particular, no existing work has introduced DP methods that can simultaneously protect entire trajectories and handle deep RL environments in the infinite-horizon discounted setting, a critical step toward deploying private RL algorithms in real-world applications.

4.4 Differentially Private Reinforcement Learning

Before diving into the main contributions of this chapter, this section introduces key concepts and definitions regarding differentially private reinforcement learning.

4.4.1 Reinforcement Learning

To model a Reinforcement Learning (RL) problem, we consider an infinite-horizon discounted MDP, that is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$ where $\mathcal{S}$ and $\mathcal{A}$ are respectively the state and action spaces, $P : \mathcal{S} \times \mathcal{A} \longrightarrow \Delta(\mathcal{S})$ the transition dynamics (where $\Delta(\mathcal{X})$ denotes the space of probability distributions over $\mathcal{X}$), $r : \mathcal{S} \times \mathcal{A} \longrightarrow [0, 1]$ the reward function, $\rho_0 \in \Delta(\mathcal{S})$ the initial state distribution and $\gamma \in [0, 1)$ a discount factor. The dynamics satisfy the Markov property, *i.e.*, the next state s' only depends on current state and action.

The goal of RL is to learn a stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ which maximizes the expected discounted return:

$$J(\pi) = \mathbb{E}_{\tau\sim\pi,\mathcal{M}}\left[\sum_{t\geq 0}\gamma^t r_t\right] \quad , \tag{4.1}$$

where $r_t := r(s_t, a_t)$. The expectation is taken with respect to the trajectories $\tau = ((s_t, a_t, r_t))_{t\geq 0}$ generated by π in the MDP $\mathcal{M}$, *i.e.*, $s_0 \sim \rho_0$, $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim P(\cdot|s_t, a_t)$.

To evaluate a given policy π, we often use the concept of *value function*. Value functions map a state s or a state-action pair (s, a) to the expected discounted return of an agent starting from s or (s, a) and then following the policy π. Formally, the state and state-action value functions of a policy π are defined as:

$$V^\pi(s) = \mathbb{E}_{\tau\sim\pi,\mathcal{M}}\left[\sum_{t\geq 0}\gamma^t r_t \middle| s_0 = s\right] \quad ,$$
$$Q^\pi(s, a) = \mathbb{E}_{\tau\sim\pi,\mathcal{M}}\left[\sum_{t\geq 0}\gamma^t r_t \middle| s_0 = s, a_0 = a\right] \quad .$$

The latter is often called *Q-function*. From these definitions, we also define the advantage function as $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$. The above quantities can be rewritten in a recursive way to obtain the *Bellman equations*, at the basis of many RL algorithms and their analysis:

$$V^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[r(s, a) + \gamma\mathbb{E}_{s'\sim P(s,a)}\left[V^\pi(s')\right]\right] \quad ,$$
$$Q^\pi(s, a) = r(s, a) + \gamma\mathbb{E}_{s'\sim P(s,a)}\left[V^\pi(s')\right] \quad .$$

We also define the (normalized) state-action occupancy measure under policy π and transition dynamics P:

$$\begin{aligned}\rho^\pi_P(s, a) &= (1-\gamma)\sum_{t\geq 0}\gamma^t\mathbb{P}(s_t = s, a_t = a|\pi, P, \rho_0) \\ &= (1-\gamma)\sum_{t\geq 0}\gamma^t\mathbb{P}(s_t = s|\pi, P, \rho_0)\cdot\pi(a_t|s_t) \quad .\end{aligned}$$

It can be interpreted as the expected discounted frequency with which the state-action pair (s, a) is visited over time, when starting from state $s \sim \rho_0$, following policy π, and evolving according to the environment dynamics P. Hereafter, we omit the expectation subscripts unless needed for clarity.

When handling large or continuous action spaces, policies and value functions are generally represented as deep neural networks, *i.e.*, $\pi := \pi_\theta$, $V := V_\phi$ and $Q := Q_\omega$ where θ, ϕ, ω are sets of parameters. **Throughout the rest of this chapter, we operate in this setting, referred to as *deep reinforcement learning*.**

We now provide a brief overview of the three main classes of (deep) reinforcement learning methods: *value-based* methods, *policy-based* methods and *model-based* methods.

4.4.1.1 Value-based methods

Value-based methods focus on learning a value function and do not explicitly represent the policy. They derive the policy indirectly by maximizing the value function:

$$\pi^\star \in \arg\max_\pi V^\pi(s) \quad \text{or} \quad \pi^\star \in \arg\max_\pi Q^\pi(s,a) \ .$$

One of the main examples of value-based algorithms is Q-Learning [Watkins, 1989], which directly learns the optimal Q-function $Q^\star : (s,a) \longrightarrow \max_\pi Q^\pi(s,a)$ from transition samples (s,a,r,s') based on the following update rule:

$$Q(s,a) \leftarrow Q(s,a) + \eta \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right) \ ,$$

where η is a learning rate. This update rule is based on the optimal Bellman equation $Q^\star(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} \left[\max'_a Q^\star(s',a') \right]$. Under conditions on the learning rate and the visitation frequency of each state-action pair, Q-Learning is guaranteed to converge to the optimal Q-function [Watkins and Dayan, 1992]. We can then derive an optimal policy $\pi^\star$ by selecting, at each state s, the action of maximum value:

$$a^\star \in \arg\max_a Q^\star(s,a) \ .$$

DQN [Mnih et al., 2013] extends Q-learning to the deep neural function approximation setting, enabling learning in high-dimensional state spaces. DQN stores transitions into a replay buffer $\mathcal{B}$ and minimizes a least square objective at each iteration i:

$$\omega_{i+1} \in \arg\max_\omega \mathbb{E}_{(s,a,r,s') \sim \mathcal{B}} \left[\left(r + \gamma \max_{a'} Q_{\omega_i}(s',a') - Q_\omega(s,a) \right)^2 \right] \ .$$

It also uses an ϵ-greedy policy to balance exploration and exploitation.

While value-based methods are often used for their simplicity and ease of implementation, they are difficult to adapt to large or continuous actions spaces and require to specify an exploration strategy manually (as does DQN).

4.4.1.2 Policy-based methods

Contrary to value-based approaches, policy-based methods directly optimize the policy. In deep reinforcement learning, the policy network π_θ is updated via gradient ascent in the parameter space to maximize the expected return $J(\theta) := J(\pi_\theta)$. The *policy gradient theorem*

[Sutton et al., 1999] characterizes the gradients of $J(\theta)$ w.r.t. θ:

$$\nabla_\theta J(\theta) = \mathbb{E}\left[Q^\pi(s_t, a_t)\nabla_\theta \log \pi_\theta(a_t|s_t)\right] \quad .$$

In practice, this gradient is estimated from samples, allowing iterative updates of the policy parameters. We typically use estimates of the form

$$\hat{g} = \hat{\mathbb{E}}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t\right] \quad ,$$

where $\hat{A}_t$ is an estimator of the advantage function at step t. This class of algorithms is referred to as *policy gradient* methods.

Advantage estimators can be computed using Monte Carlo methods (*e.g.*, REINFORCE [Williams, 1992]) or from a trained value function V_ϕ, also represented by a neural network (*e.g.*, A2C [Mnih et al., 2016]). The latter are often called *actor-critic* methods, as the value function is used as a "critic" evaluating the policy we are learning. In contrast to value-based methods, the value function is only used as a tool to provide a direction for the policy update. Another popular class of algorithms are *trust-region* methods such as TRPO [Schulman et al., 2015] and PPO [Schulman et al., 2017]. Taking into account the geometry of the policy space during policy optimization, they are typically more stable and reliable.

Policy-based methods are better suited for continuous action spaces and can directly learn stochastic policies that naturally balance exploration and exploitation, without the need for manually designed exploration strategies.

4.4.1.3 Model-based methods

Model-based methods learn a model of the environment and exploit it to generate additional data, which is then used to derive a policy (generally in conjunction with real environment data). Precisely, these methods learn estimates of the transition dynamics $\hat{P}$ and the reward function $\hat{r}$, resulting in an estimate of the MDP $\hat{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \hat{P}, \hat{r}, \gamma, \rho_0)$. The predictive model $\hat{\mathcal{M}}$ is then used to generate rollouts which are used directly or indirectly during policy optimization (which is often referred to as *planning*).

In deep reinforcement learning with continuous action spaces, the model is typically represented as a normal distribution parameterized by neural networks μ_ϕ and Σ_ψ:

$$(s_{t+1}, r_t) \sim \mathcal{N}\left(\mu_\phi(s_t, a_t), \Sigma_\psi(s_t, a_t)\right) \quad .$$

The standard iterative procedure for Model-Based Policy Optimization (MBPO) [Janner et al., 2019] in deep RL is as follows. At each epoch, the model is trained on real environment data D_{env} via maximum likelihood. Then, we use the model to generate a series of k-step rollouts, starting from real states sampled from the data buffer (*i.e.*, *branching*). The policy

parameters are then updated based on the model data using the policy gradient.

Model-based RL is generally more efficient than model-free RL because of its ability to sample data from the model, which is particularly useful when data collection is costly. However, learning a good model is difficult, and model bias often leads to worse asymptotic performance compared to model-free methods [Janner et al., 2019].

4.4.2 Differential Privacy in Reinforcement Learning

We consider the training of deep RL policies (or *agents*) on sensitive user data. When a user u interacts with the environment, they generate a trajectory $\tau_u = ((s_t^u, a_t^u, r_t^u))_{t\geq 0}$. In the following, we identify a user u with the trajectory τ_u they generate, and use these terms interchangeably. The states $(s_t^u)_{t\geq 0}$, the actions $(a_t^u)_{t\geq 0}$ and the rewards $(r_t^u)_{t\geq 0}$ may all contain sensitive information about user u. Table 4.1 provides different examples from real-world scenarios. It emphasizes the need to protect all elements of a trajectory (states, actions, and rewards alike), warranting more comprehensive privacy protection than provided by Wang and Hegde [2019], Cundy et al. [2024] or Zhang et al. [2025].

Table 4.1: Examples of sensitive information that could be revealed through the states, actions and rewards for different application scenarios of private RL.

	Treatment Recommendation	**Targeted Advertisement**
States	Patient's medical history	User's browsing history
Actions	Patient's treatment	Proposed ads
Rewards	Patient's response to treatment	User's preferences (clicked ads)
	LLM Fine-Tuning with RLHF	**Autonomous Driving**
States	Prompt content	User's environment
Actions	Memorized content, Human pref.	User's driving behavior
Rewards	Human preferences	User's driving behavior

Formalizing privacy in reinforcement learning is challenging. In other fields, differential privacy is typically applied at the *example-level*, under the assumption that the examples in the dataset are independent. If this assumption is already questionable in supervised learning, it certainly does not hold in RL where the transitions (s_t, a_t, r_t) are temporally correlated. Several works, for instance Liu et al. [2016], have demonstrated that data correlation degrades privacy guarantees in the traditional *per-example* setting. Moreover, Fan et al. [2025] highlight that in sequential settings, sensitive information exists not only in individual events, such as a transition, but also in temporal patterns. It thus appears that protecting individual transitions is insufficient in RL, calling instead for data protection at the *trajectory level*. Preventing adversaries from inferring information about a given user, associated with a trajectory, is also very intuitive. In sensitive domains such as healthcare, the mere presence of a trajectory τ_u in the training data — which can be assessed with an

MIA — may reveal sensitive attributes: for example, if the training set consists of patients with a specific disease, membership of τ_u implies that user u has that disease.

When defining differential privacy in RL, we therefore consider two adjacent datasets as two datasets D, D' where D' is obtained by adding or removing one full trajectory from D (*add-or-remove DP*). We refer to this as *trajectory-adjacency*. Considering privacy at the trajectory level has significant technical implications. In particular, this makes the use of methods designed for example-level privacy — such as DP-SGD — unsuitable. In this chapter, a key technical contribution is to develop training methods that efficiently guarantee privacy at the level of trajectories.

This chapter considers both the offline and online RL settings. Given their distinct privacy implications, we separately develop the threat models and the corresponding differential privacy definitions for each setting.

4.4.2.1 Offline setting

In offline RL, we assume access to a fixed dataset of K trajectories $D = (\tau_u)_{u=1}^K$, where each trajectory $\tau_u = (s_t^u, a_t^u, r_t^u)_{t\geq 0}$ has been collected with an unknown behavioral policy π^B. π^B therefore corresponds to the user behavior during data collection. The objective is then to learn a policy $\hat{\pi}$ from D (without any further interaction with the environment) which performs as best as possible in $\mathcal{M}$.

Due to its similarities with supervised learning, it is rather natural to define differential privacy in the offline RL setting. An offline RL method can indeed be seen as a black-box randomized algorithm h taking in as input a fixed dataset D, partitioned in trajectories, and outputting a policy π. An adversary with access to π may learn to infer the membership of a specific trajectory in D, which can, as emphasized before, reveal sensitive user information. Hence, similarly to Qiao and Wang [2023a], we use the following informal DP definition for offline RL, which we refer to as *trajectory-level differential privacy* (TDP): adding or removing a single trajectory from the input dataset of an offline RL algorithm must not impact significantly the distribution of the output policy.

We introduce the following formal definition for trajectory-level differential privacy (TDP) in offline RL which protects whole trajectories. It states that the learned policy is roughly the same for two trajectory-adjacent datasets D, D'. It can be seen as a reformulation of the definition from Qiao and Wang [2023a], which is the first work to tackle DP in this setting.

Definition 4.4.1. *Offline Trajectory-level Differential Privacy (TDP).* Let h be an offline RL algorithm, that takes as input an offline dataset and outputs a policy. Given $\epsilon > 0$ and $\delta \in (0, 1)$, $h : D \longrightarrow \Pi$ is (ϵ, δ)-TDP if for any trajectory-adjacent datasets D, $D' = D \setminus \{\tau_u\}$, and any subset of policies $E \subseteq \Pi$:

$$\mathbb{P}\left(h(D) \in E\right) \leq e^{\epsilon} \cdot \mathbb{P}\left(h(D') \in E\right) + \delta \ .$$

4.4.2.2 Online setting

Formalizing differentially private RL in the online setting, where a potentially infinite sequence of users $u_1, u_2, \ldots$ interact with the system, appears more difficult: there are many ways of collecting and using data (for instance, *on-policy vs. off-policy*), and contrary to the offline setting, the input and output of the training algorithm are not clearly separated. Providing a meaningful and practical definition of differential privacy in this setting therefore requires careful consideration, and it is unlikely that a single universal definition can accommodate all use cases.

In the online RL literature, the most prevalent notion of DP — applied in the context of episodic RL with finite action spaces, where each episode is of fixed length H and $|\mathcal{A}| = A < \infty$ — is *joint differential privacy* (JDP, Definition 4.4.2) [Vietri et al., 2020]. Let $U = (u_1, ..., u_M)$ be a sequence of M users. Each user $u \in U$ can be represented as a tree of depth H which encodes all the state and reward responses they would give to all the A^H possible action sequences recommended by the RL policy, and we only observe the state and reward information along a single root-to-leaf path in the tree, *i.e.*, a single trajectory per user. JDP states that the presence of a user u_m, $m \in [\![1, M]\!]$, does not significantly influence the actions recommended to the other users $u \neq u_m$. Therefore, even if all users, past and future, collude against u_m by sharing their recommendations (similarly, if an adversary had access to this data), they could not extract more information about u_m's private data than allowed by the DP bound.

Definition 4.4.2. *Joint Differential Privacy (JDP).* Assume an episodic MDP with horizon H and finite action space $\mathcal{A}$. Given $\epsilon > 0$ and $\delta \in (0, 1)$, a mechanism $h : U \to \mathcal{A}^{MH}$ is (ϵ, δ)-joint DP if for any $M \in \mathbb{N}^\star$, any $u \in [\![1, M]\!]$, any pair of user sequences U, U' differing on a single user, and any subset of action sequences $E \subseteq \mathcal{A}^{MH}$:

$$\mathbb{P}[h_{-m}(U) \in E] \leq e^\epsilon \mathbb{P}[h_{-m}(U') \in E] + \delta \ ,$$

where $h_{-m}(U)$ is the sequence of actions recommended to all users $u_1, ..., u_M$ but u_m.

JDP assumes an adversary observing the sequences of recommended actions, or, equivalently, collusion between users. While this threat model suits certain scenarios, alternative models may be more appropriate in others. We consider a setting where we alternate between training phases, where the agent learns from on-policy private user data, and exploitation phases, where the updated policy is deployed for use by other users. This type of *on-policy* procedure, described in Algorithm 7, arises in many contexts. For example, an LLM may be fine-tuned with RL at various stages, and between training phases, it can be made available to users, for instance *via* a public API. Similarly, a recommender system can be deployed on user devices and regularly improved with RL as data is collected. In such settings, we are not really concerned with an adversary that would observe every recommended action or users exchanging information to collude against another user. Instead, our concern lies with what the released policy, which can be easily accessed by a malicious adversary, may

Algorithm 7 Private Online RL (*Training Phase*)

Initialize π_1, iteration $i \leftarrow 1$, on-policy data $D_i \leftarrow \emptyset$
for user $u = 1, 2, ...$ **do**
  $\tau_u \leftarrow \emptyset$
  for $t = 0, 1, 2, ...$ **do**
    u sends **sensitive state** s_t^u to the system S
    S recommends **sensitive action** $a_t^u \sim \pi_i(\cdot|s_t^u)$
    S and u observe **sensitive reward** r_t^u
    $\tau_u \leftarrow \tau_u \cup \{(s_t^u, a_t^u, r_t^u)\}$
  end for
  $D_i \leftarrow D_i \cup \{\tau_u\}$
  if time to update **then**
    Update policy: $\pi_{i+1} \leftarrow h(\pi_i, D_i)$
    $i \leftarrow i + 1$, $D_i \leftarrow \emptyset$
  end if
end for

reveal about the private data it has been trained on. A natural privacy requirement is thus to ensure that the released policy (e.g., a fine-tuned LLM in RLHF) does not leak too much information about any specific user data (or trajectory) used during training.

We thus consider the following DP model: at each iteration i, we view the RL training algorithm as a randomized mechanism h taking interaction data $D_i = (\tau_u)_{u=M_{i-1}+1}^{M_i}$ — where the i-th policy update occurs after the M_i-th user has interacted with the system — and the current policy π_i as input, and releasing a new policy π_{i+1} as output. The released policy π_{i+1} can be used by future users $u = M_i + 1, M_i + 2, \ldots$ and also be observed by a malicious adversary. Therefore, we do not want π_{i+1} to reveal sensitive information about any past user $u = 1, ..., M_i$.

Unlike the offline setting, we cannot properly define trajectory-level DP over the whole user sequence, since removing or adding a trajectory at iteration i may change subsequent data D_j for $j > i$. To circumvent this, JDP assumes a tree representation of each user: while we only observe one trajectory , the tree encodes all possible trajectories, so that we can define user-adjacent datasets over the whole sequence. However, such representation is impractical in more general γ-discounted MDPs with continuous action spaces. Therefore, we focus on trajectory-level DP updates at each iteration, such as stated in Definition 4.4.3.

Definition 4.4.3. *On-Policy TDP Updates.* Given $\epsilon > 0$ and $\delta \in (0, 1)$, the update mechanism at iteration i, $\pi_{i+1} = h(D_i, \pi_i)$, is ($\epsilon$, δ)-TDP if for any pair of trajectory-adjacent on-policy dataset D_i, D_i', where D_i' is obtained from D_i by adding or removing the trajectory τ_u for $u \in [\![M_{i-1} + 1, ..., M_i]\!]$, and any subset of policies $E \subseteq \Pi$:

$$\mathbb{P}[h(D_i, \pi_i) \in E] \leq e^{\epsilon}\mathbb{P}[h(D_i', \pi_i) \in E] + \delta \ .$$

Although its privacy guarantees are defined at the iteration level, Algorithm 7 with

TDP updates can still be considered to offer strong privacy protection over the whole user sequence. Indeed, thanks to the *post-processing* property of DP, the update at iteration i cannot reverse the privacy protection provided by the last iterate π_i; moreover, the on-policy data D_i only contains data from the most recent users $u = M_{i-1} + 1, \ldots, M_i$. Therefore, the i-th update does not leak further information regarding the trajectories of past users $u = 1, \ldots, M_{i-1}$. Consequently, if the update mechanism at each iteration is (ϵ, δ)-TDP, the released policy π_{i+1} effectively protects the trajectories from all users $u = 1, \ldots, M_i$. Furthermore, this private mechanism is compatible with settings beyond episodic MDPs with finite action spaces, and also explicitly shields the released policy, not merely the actions, therefore protecting against privacy attacks that would exploit information about the policy itself (for instance, Fu et al. [2023] attack fine-tuned LLMs by exploiting the model logits).

Note that, in Algorithm 7, users need not arrive sequentially but can interact asynchronously with the environment. Moreover, being on-policy, Algorithm 7 only uses the data of a given user for a single policy update before discarding it, which avoids accumulating privacy leakage as a consequence of the sequential composition property of DP. By contrast, the privacy guarantees of an off-policy method that stores and reuses data would demand more careful consideration.

4.5 Differentially Private Offline Model-Based Reinforcement Learning

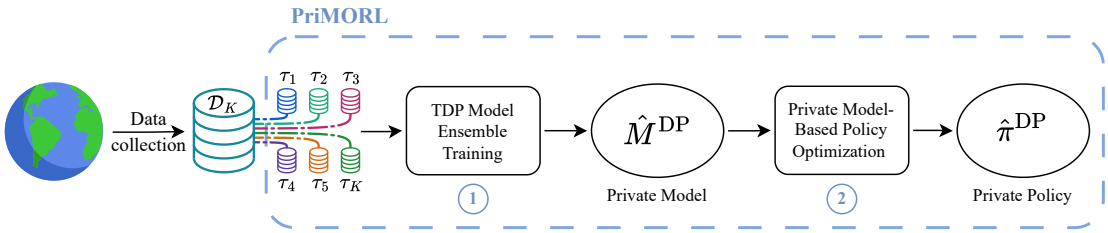


Figure 4.1: PriMORL with its two main components: 1) private model training; 2) MBPO.

The first contribution of this chapter presents our model-based approach for learning differentially private RL agents from offline data, which we call PriMORL (for Private Model-Based Offline RL). Figure 4.1 provides a high-level description of the two-phase approach of PriMORL. We first address the learning of a private model from offline data (Section 4.5.1). Then, we demonstrate how we optimize a policy under the private model (Section 4.5.2), and show that ensuring model privacy alone is enough to achieve a private policy. We also provide a theoretical analysis of the impact of model privacy on the policy. Finally, we assess PriMORL empirically on standard control tasks (Section 4.5.3).

4.5.1 Model Learning with Differential Privacy

We consider a model-based approach as described in Section 4.4.1. We use the model $\hat{\mathcal{M}}$ as a simulator of the environment to learn a policy $\hat{\pi}$, without further access to the dataset or interactions with the real environment modeled by $\mathcal{M}$. Note that if the policy $\hat{\pi}$ is trained to maximize the expected discounted return under the model $\hat{\mathcal{M}}$, we eventually want to evaluate the policy in the true environment $\mathcal{M}$, that is using $J(\hat{\pi})$. Alongside the sample efficiency gains of a model-based approach, learning a private model of the environment can be valuable in its own right, as system engineers can leverage simulators to design and test algorithms [Kégl et al., 2021].

Following previous work (*e.g.*, Yu et al. [2020]), we jointly model the transition dynamics $\hat{P}$ and reward $\hat{r}$ with a Gaussian distribution $\hat{M}$ conditioned on the current state and action:

$$\hat{M}_\theta \left(\Delta_t^{t+1}(s), r_t | s_t, a_t\right) = \mathcal{N}\left(\mu_\phi(s_t, a_t), \Sigma_\psi(s_t, a_t)\right) \quad ,$$

where $\theta = (\phi, \psi)$ are the parameters of a neural network. As standard in the literature, the model learns the state differences $\Delta_t^{t+1}(s) := s_{t+1} - s_t$ instead of the next state in order to ensure local continuity.

To carry out uncertainty estimation (see Section 4.5.2), we train an ensemble of N models $\hat{M}_{\theta_i}$, $i \in [\![1, N]\!]$, all sharing the same architecture. The core aspect of PriMORL, as illustrated in Figure 4.1, is therefore to learn a trajectory-level DP dynamics model ensemble.

This poses two major challenges. First, the traditional approach to privatize neural networks, DP-SGD, is designed for example-level privacy and is unsuitable for guaranteeing TDP. Moreover, since the training of all the models in the ensemble consumes the same dataset D, we must deal with the dependence of the privacy budget on the ensemble size N. A key contribution of our work, developed in this section, is thus to introduce a training method that 1) guarantees privacy at the trajectory level and 2) efficiently manages the privacy budget across an ensemble of models.

4.5.1.1 Trajectory-level DP Training for Model Ensembles

As DP-SGD ensures per-example privacy by clipping each per-example gradient, limiting the contribution of each data point to the final model, the key to achieving trajectory-level privacy is to compute and clip per-trajectory updates. Therefore, our training method partitions the dataset by trajectories, *i.e.*, $D = \bigcup_{k=1}^{K} \{\tau_k\}$, computes independent updates from each trajectory's data, and bounds the L_2-norm of each update before aggregation. To leverage privacy amplification by subsampling (*i.e.*, at each round, a given trajectory is sampled with very small probability, making it more indistinguishable within the larger dataset), as well as multiple local updates from the same trajectory's data, we further build on ideas from the DP federated learning literature [McMahan et al., 2018], abstracting the

different users as clients. Finally adapting it to ensembles of dynamics models, the combined approach allows us to address the unique privacy challenges of our task. We present the resulting training procedure in Algorithm 8.

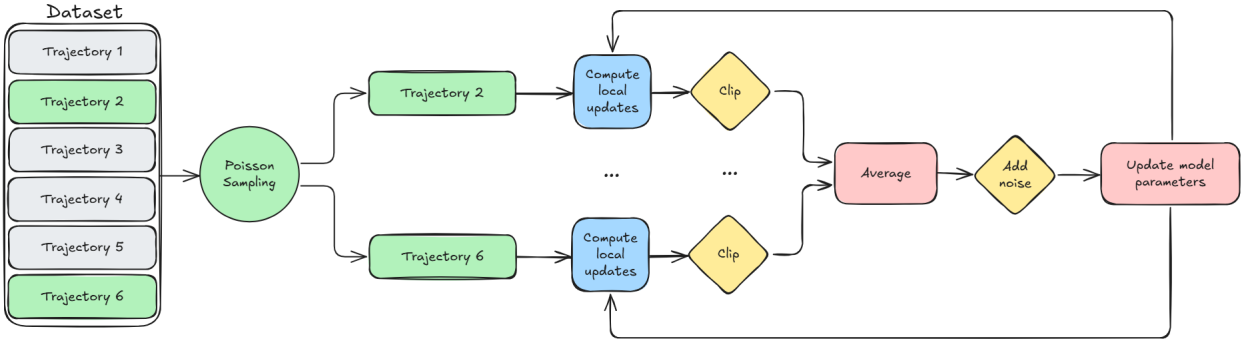


Figure 4.2: Illustration of trajectory-level DP model training. In this example, the dataset has six trajectories and we sample two trajectories ($q = 1/3$) at each training round.

The core idea behind TDP Model Ensemble Training, illustrated in Figure 4.2, is to draw at each iteration t a random subset $\mathcal{U}_t$ of the K trajectories (line 2) using Poisson sampling. Each trajectory is sampled (with replacement) with probability q, resulting in an expected qK trajectories being selected per step. The sampling ratio q plays a critical role in determining the strength of privacy guarantees. Specifically, a smaller q reduces the likelihood of any given trajectory being included in an update, thereby limiting its influence on the final model — this forms the basis of privacy amplification by sub-sampling. However, in the offline RL setting, where trajectory data is highly correlated, q must remain large enough to ensure that the model update incorporates a sufficiently diverse set of trajectories. Interestingly, while the theoretical analysis of common private deep learning methods like DP-SGD relies on Poisson sampling, most implementations actually use fixed-size batches with shuffling in practice, in order to overcome the computational challenges due to batches of varying size. This can lead to significant underestimation of the actual privacy leakage, as pointed out in Chua et al. [2018]. Our implementation, however, does indeed use Poisson sampling, allowing us to compute correct theoretical privacy guarantees.

For each $\tau_k \in \mathcal{U}_t$, the clipped gradients $\{\Delta_{i,k}^{\text{clipped}}(t)\}$ are then computed from τ_k's data only (line 3 to 7). During this step, we may perform multiple local updates on the same trajectory's data without incurring more privacy leakage, as only properly privatized trajectory-level updates are incorporated into the released global model. This is made possible because the global model is updated with clipped gradients only. We later introduce ensemble-adapted clipping strategies to control the privacy budget over model ensembles, ensuring that the sensitivity of the ensemble gradient $\Delta_k^{\text{clipped}}(t) = (\Delta_{i,k}^{\text{clipped}}(t))_{i=1}^N$ is bounded by S. We then compute an unbiased estimator of the subset gradient average whose sensitivity is bounded by S/qK (line 8). We then apply the Gaussian mechanism with magnitude $\sigma = zS/qK$, where z controls the strength of the privacy guarantee ϵ, and update the ensemble model $\theta(t) = (\theta_i(t))_{i=1}^N$ with noisy gradient (line 9).

Algorithm 8 TDP Model Ensemble Training

Input: offline dataset D, sampling ratio $q \in (0,1)$, noise multiplier $z \geq 0$, clipping norm $S > 0$, local epochs E, batch size B, learning rate η
Output: private model $\hat{M}_\theta$
Initialize model parameters θ_0
for each iteration $t \in [\![0, T-1]\!]$ **do**
  $\mathcal{U}_t \leftarrow$ (sample with replacement trajectories from D with prob. q)
  for each trajectory $\tau_k \in \mathcal{U}_t$ **do**
    Clone current models $\{\theta_i^{\text{start}}\}_{i=1}^N \leftarrow \{\theta_i(t)\}_{i=1}^N$
    $\theta \leftarrow \theta^{\text{start}} := (\theta^{\text{start}})_{i=1}^N$
    for each local epoch $i \in [\![1, E]\!]$ **do**
      $\mathcal{B} \leftarrow$ (τ_k's data split into size B batches)
      for each batch $b \in \mathcal{B}$ **do**
        $\theta \leftarrow \theta - \eta \nabla \mathcal{L}(\theta; b)$
        $\theta \leftarrow \theta^{\text{start}} + \text{EnsembleClip}(\theta - \theta^{\text{start}}; S)$
      end for
    end for
    $\Delta_{t,k}^{\text{clipped}} \leftarrow \theta - \theta^{\text{start}}$
  end for
  $\Delta_i^{\text{avg}}(t) = \frac{\sum_{k \in \mathcal{U}_t} \Delta_{i,k}^{\text{clipped}}(t)}{qK}$
  $\theta(t+1) \leftarrow \theta(t) + \Delta^{\text{avg}}(t) + \mathcal{N}\left(0_{N_d}, \left(\frac{zS}{qK}\right)^2 I_{N_d}\right)$
end for

} EnsClipGD $\left(\tau_k, \{\theta_i^{\text{start}}\}_{i=1}^N; C, E, B\right)$

4.5.1.2 Privacy Guarantees for the Model

We can now derive formal privacy guarantees for a model trained using Algorithm 8. A key challenge in our setting arises from training an ensemble of N models for uncertainty estimation, all using the same dataset D. Treating each model independently, with separate clipping and noise addition, would be inefficient and significantly increase the privacy budget by composition. This could be mitigated by limiting the ensemble size, but at the cost of performance, as shown in Lu et al. [2022].

To address this challenge, we process all the gradients of the model ensemble simultaneously and distribute the global clipping norm S across all models, on the same principle as the per-layer clipping used in McMahan et al. [2018]. Denoting $\Delta_{i,\ell}$ the gradient of layer ℓ for model i, we propose and experiment with two ensemble clipping strategies:

- **Flat Ensemble Clipping**, which clips the whole model gradient $\Delta_i = (\Delta_{i,\ell})_{\ell=1}^L$ with $S_i = S/\sqrt{N}$;
- **Per Layer Ensemble Clipping**, which clips per-layer gradients $\Delta_{i,\ell}$ with $S_{i,\ell} = S/\sqrt{N \times L}$.

For both strategies, $S = \sqrt{\sum_{i=1}^N S_i^2} = \sqrt{\sum_{i=1}^N \sum_{\ell=1}^L S_{i,\ell}^2}$, such that S is adequately distributed

Algorithm 9 Ensemble Clipping (EnsembleClip)

Input: ensemble size N, number of model layers L, unclipped gradient $\Delta = \{\Delta_{i,\ell}\}_{i,\ell=1}^{N,L}$, clipping norm S
Output: clipped gradient Δ^{clipped}
$\Delta_i \leftarrow (\Delta_{i,\ell})_{\ell=1}^{L}$, $C_i = \frac{C}{\sqrt{N}}$

$$\Delta_i^{\text{clipped}} \leftarrow \frac{\Delta_i}{\max\left(1, \frac{\|\Delta_i\|_2}{C_i}\right)}, \quad j = 1, ..., m.$$

across all models and we verify that $\Delta_k^{\text{clipped}} = (\Delta_{i,k}^{\text{clipped}})_{i=1}^{K}$ has sensitivity bounded by S (see the proof of Theorem 4.5.1), and that the contribution of a given trajectory to the *model ensemble* is appropriately limited. Ensemble clipping eliminates the linear dependence of the privacy budget on the number of models. However, it does not entirely remove the negative impact of increasing N. Indeed, for a given noise level, a larger N requires a smaller clipping threshold S_i or $S_{i,\ell}$, which can degrade model convergence by losing too much information from the original gradient. Nevertheless, the clipping threshold scales with the square root of N, mitigating the impact to some extent.

We now formally derive the privacy guarantees for an ensemble of models trained with Algorithm 8. Mapping users in federating learning to trajectories in offline RL, we can directly adapt Theorem 1 from McMahan et al. [2018] to state that, with the sensitivity of clipped gradients $\Delta_{i,k}^{\text{clipped}}$ effectively bounded by S, the moments accounting method from Abadi et al. [2016] computes correctly the privacy loss of Algorithm 8 at trajectory-level for the noise multiplier $z = \sigma/\mathbf{S}$ with $\mathbf{S} = S/qK$. We can therefore use the moments accountant to compute, given $\delta \in (0, 1)$, $z > 0$, $q \in (0, 1)$ and $T \in \mathbb{N}$, the total privacy budget ϵ spent by Algorithm 8, and obtain (ϵ, δ)-TDP guarantees for our dynamics model, as stated in Theorem 4.5.1.

Theorem 4.5.1. (ϵ, δ)-TDP guarantees for dynamics model. *Given $\delta \in (0, 1)$, noise multiplier z, sampling ratio q and number of training iterations T, let $\epsilon := \epsilon^{MA}(z, q, T, \delta)$ be the privacy budget computed by the moments accounting method from Abadi et al. [2016]. The dynamics model output by Algorithm 8 is (ϵ, δ)-TDP.*

Proof. Theorem 1 from McMahan et al. [2018] shows that the moments accounting method from Abadi et al. [2016] computes correctly the privacy loss of DP-FedAvg at user-level for the noise multiplier $z = \sigma/\mathbf{S}$ with $\mathbf{S} = S/qK$ if, for each user u_k, the clipped gradient $\Delta_k^{\text{clipped}}$ computed from u_k's data has sensitivity bounded by S (referred to as **Condition 1**). With TDP Model Ensemble Training, we train the model ensemble as a single big model: at each training iteration, the same input batch is processed forward by all models in a single pass, a single loss is computed for the ensemble, and the parameters are then updated in a single backward pass. The ensemble of models can therefore be seen as a concatenation of all individual models, equivalent to a larger model $\theta = (\theta_i)_{i=1}^{N}$. We can therefore extend this theorem to our setting by mapping users in federating learning to trajectories in offline RL,

as long as **Condition 1** holds for every trajectory τ_k.

Since we use **ensemble clipping**, we verify that, for each trajectory τ_k, the ensemble gradient $\Delta_k^{\text{CLIPPED}} = \left(\Delta_{i,k}^{\text{CLIPPED}}\right)$ has sensitivity bounded by S. With **flat ensemble clipping**, the gradient of each model $i \in [\![1, N]\!]$ is clipped by a factor $S_i = \frac{S}{\sqrt{N}}$ (see Algorithm 9). By construction, $\Delta_{i,k}^{\text{CLIPPED}}$ has sensitivity bounded by S_i, *i.e.*, $\max_{d(D,D')=1} \|\Delta_{i,k}^{\text{CLIPPED}}(D) - \Delta_{i,k}^{\text{CLIPPED}}(D')\|_2 \leq S_i$. Therefore, for two trajectory-neighboring datasets D and D':

$$\begin{aligned}
\|\Delta_k^{\text{CLIPPED}}(D) - \Delta_k^{\text{CLIPPED}}(D')\|_2 &= \| \left(\Delta_k^{\text{CLIPPED}}(D) - \Delta_k^{\text{CLIPPED}}(D')\right)_{i=1}^N \|_2 \\
&= \sqrt{\sum_{i=1}^N \|\Delta_{i,k}^{\text{CLIPPED}}(D) - \Delta_{i,k}^{\text{CLIPPED}}(D')\|_2^2} \\
&\leq \sqrt{\sum_{i=1}^N S_i^2} \\
&= \sqrt{\sum_{i=1}^N \frac{S^2}{N}} \\
&= S \ .
\end{aligned}$$

This implies $\max_{d(D,D')=1} \|\Delta_k^{\text{CLIPPED}}(D) - \Delta_k^{\text{CLIPPED}}(D')\|_2 \leq S$: $\Delta_k^{\text{CLIPPED}}$ has sensitivity bounded by S. We can derive the same proof for **per-layer ensemble clipping**. Therefore, Theorem 1 from McMahan et al. [2018] holds for TDP Model Ensemble Training.

We can therefore use the moments accountant ϵ^{MA} to compute, given $z > 0$, $\delta \in (0, 1)$, $q \in (0, 1)$ and $T \in \mathbb{N}$, the total privacy budget ϵ spent by Algorithm 8, *i.e.*, $\epsilon = \epsilon^{\text{MA}}(z, q, T, \delta)$.

The dynamics model output by Algorithm 8 is therefore (ϵ, δ)-TDP. □

4.5.2 Policy Optimization under a Private Model

Now that we learned a private model $\hat{M}$ from offline data, we use it as a simulator of the environment to learn a private policy $\hat{\pi}$ with a model-based policy optimization approach. In this section, we discuss the challenges of model-based policy optimization with privacy constraints and derive formal privacy guarantees for a policy learned under a private model.

4.5.2.1 Impact of Privacy on Policy Value

It is first essential to examine the complexities of policy optimization in model-based offline RL and assess whether they are amplified in the private setting. A major challenge in model-based offline RL is to handle the discrepancy between the true and the learned dynamics when optimizing the policy. Indeed, model inaccuracies cause errors in policy evaluation that may be exploited, resulting in poor performance in the real environment. According to the Simula-

tion Lemma [Kearns and Singh, 2002, Xu et al., 2020], the value evaluation error of a policy π in model-based RL can be decomposed into a *model error* term and a *policy distribution shift* term. Formally, if the model error is bounded as $\mathbb{E}_{(s,a)\sim\rho_P^{\pi^B}}\left[D_{KL}\left(P(\cdot|s,a)\|\hat{P}(\cdot|s,a)\right)\right] \leq E_m$ and the distribution shift is bounded as $\max_s D_{KL}\left(\pi(\cdot|s)\|\pi^B(\cdot|s)\right) \leq E_\pi$, then the value evaluation error of π is bounded as:

$$|\hat{V}^\pi - V^\pi| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{E_m} + \frac{2\sqrt{2}}{(1-\gamma)^2}\sqrt{E_\pi} \ , \tag{4.2}$$

where $\hat{V}^\pi$ and V^π denote the value of π under the learned and true dynamics, respectively. Controlling this quantity for an arbitrary π is crucial in our setting, as it ensures that the learned MDP is a reasonable simulator of the true MDP. Moreover, (4.2) directly implies a bound on the sub-optimality gap, since $|V^\star - V^{\hat{\pi}}| \leq 2\sup_\pi |\hat{V}^\pi - V^\pi|$.

Under some assumptions regarding the model loss function, Proposition 4.5.2 states the model error term in terms of the size N_D of the dataset. In the sequel, we focus on the model error term and denote E_{DS} the distribution shift term, which does not depend on the learned dynamics and is therefore not directly affected by private training.

Proposition 4.5.2. Value evaluation error in non-private offline MBRL. *Let π be an arbitrary policy. Let the model loss function be L-Lipschitz and Δ-strongly convex, and assumptions from the simulation lemma hold. There is a stochastic convex optimization algorithm for learning the model and a constant M such that, with probability at least $1-\alpha$, and for sufficiently large N_D, the value evaluation error of π is bounded as:*

$$|\hat{V}^\pi - V^\pi| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2} \cdot M \cdot \frac{L\log^{1/2}(N_D/\alpha)}{\sqrt{\Delta N_D}} + E_{DS}$$

When we learn the model with differential privacy, we disrupt model convergence because of gradient clipping and noise. This likely results in a less accurate dynamics model (although it may help prevent overfitting in some cases) and increased value evaluation error. Intuitively, DP training impacts model error in (4.2) as a direct result of gradient perturbations: Bassily et al. [2014], in particular, show that noisy gradient descent (GD) has increased excess risk compared to non-private GD. In the simpler case where the model is trained with a vanilla DP noisy GD algorithm, Proposition 4.5.3 states the value evaluation error under the private model.

Proposition 4.5.3. Value evaluation error in private offline MBRL. *Let assumptions from Proposition 4.5.2 hold. If the model is learned with (ϵ,δ)-DP gradient descent, then, with probability at least $1-\alpha$, there is a constant M' such that for large enough N_D, the value evaluation error of π is bounded as:*

$$|\hat{V}^\pi_{DP} - V^\pi| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2} \cdot M' \cdot \frac{Ld^{1/4}C(N_D,\delta,\alpha)}{\sqrt{\Delta N_D \epsilon\alpha}} + E_{DS} \ ,$$

where $C(N_D, \delta, \alpha) = \log(N_D/\delta) \cdot \operatorname{poly}\log(\alpha^{-1})$ and $\hat{V}^{\pi}_{\mathrm{DP}}$ is the value of π under the privately learned dynamics. Comparing the value evaluation errors in Propositions 4.5.2 and 4.5.3, we observe how DP training may degrade performance in MBRL. The private bound has an explicit dependence on the problem dimension d which is not present in the non-private bound, and the $\sqrt{\epsilon}$ factor in the denominator shows that the error will degrade with strong privacy guarantees.

We now prove these two propositions.

Proof. (Propositions 4.5.2 and 4.5.3)

Let $\mathcal{F}$ denote the function class of the model. The model is estimated by maximizing the likelihood of the data $D = (s_i, a_i, s'_i)_{i=1}^N$, which is collected by an unknown behavioral policy π^B. This is equivalent to minimizing the negative log-likelihood. The population risk of the estimated model $\hat{P}$ obtained with DP-SGD is therefore:

$$\mathcal{L}(\hat{P}) = \mathbb{E}_{(s,a)\sim\rho^{\pi^B}_P, s'\sim P(\cdot|s,a)} \left[-\log \hat{P}(s'|s,a) \right] \quad .$$

Let us further assume that the true model P belongs to the function class $\mathcal{F}$, and that $P \in \operatorname{argmin}_{P'\in\mathcal{F}} \mathcal{L}(P')$. We can therefore write the excess population risk of the model estimator $\hat{P}$ as:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) = \mathbb{E}_{(s,a)\sim\rho^{\pi^B}_P, s'\sim P(\cdot|s,a)} \left[\frac{\log P(s'|s,a)}{\log \hat{P}(s'|s,a)} \right] \quad .$$

But, denoting $D_{\mathrm{KL}}(A, B)$ the Kullback-Leibler divergence between distributions A, B:

$$D_{\mathrm{KL}} \left(P(s,a), \hat{P}(s,a) \right) = \mathbb{E}_{s'\sim P(\cdot|s,a)} \left[\frac{\log P(s'|s,a)}{\log \hat{P}(s'|s,a)} \right] \quad .$$

We can therefore rewrite the above excess population risk as:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) = \mathbb{E}_{(s,a)\sim\rho^{\pi^B}_P} \left[D_{\mathrm{KL}} \left(P(s,a), \hat{P}(s,a) \right) \right] \quad . \tag{4.3}$$

Private case. When the objective function $\mathcal{L}$ is L-Lipschitz and Δ-strongly convex, Bassily et al. [2014] show (Theorem F.2) that, given N_D data points, a noisy gradient descent algorithm with (ϵ, δ)-DP guarantees satisfies, with probability at least $1 - \alpha$:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) = \mathcal{O}\left(\frac{L^2 \sqrt{d} \log^2(N_D/\delta) \cdot \operatorname{poly}\log(1/\alpha)}{\Delta N_D \epsilon \alpha} \right) \quad . \tag{4.4}$$

Non-private case. In the non-private case, Shalev-Shwartz et al. [2009] provide the following bound under the same assumptions:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) = \mathcal{O}\left(\frac{L^2 \log(N_D/\alpha)}{\Delta N_D}\right) \ . \tag{4.5}$$

Let π be an arbitrary policy. From the Simulation Lemma [Kearns and Singh, 2002, Xu et al., 2020], for a MDP $\mathcal{M}$ with reward upper bounded by $r_{\max} = 1$ and dynamics P, a behavioral policy π^B and a learned transition model $\hat{P}$, if we have:

$$\mathbb{E}_{(s,a)\sim\rho_P^{\pi^B}}\left[D_{\mathrm{KL}}\left(P(s,a), \hat{P}(s,a)\right)\right] \leq \varepsilon_M \ , \tag{4.6}$$

which by 4.3 is equivalent to:

$$\mathcal{L}(\hat{P}) - \mathcal{L}(P) \leq \varepsilon_M \ , \tag{4.7}$$

and if, moreover, the divergence between π and the behavioral policy is bounded:

$$\max_s D_{\mathrm{KL}}\left(\pi(\cdot|s), \pi^B(\cdot|s)\right) \leq \varepsilon_\pi \ , \tag{4.8}$$

then the value evaluation error of π is bounded as:

$$|\hat{V}^\pi - V^\pi| \leq \frac{\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_M} + \frac{2\sqrt{2}\gamma}{(1-\gamma)^2}\sqrt{\varepsilon_\pi} \ . \tag{4.9}$$

Since $f(x) = \mathcal{O}(g(x))$ implies $\sqrt{f(x)} = \mathcal{O}(\sqrt{g(x)})$ [1], we note that we can replace $\sqrt{\varepsilon_M}$ in the model term of the right-hand side of 4.9 by the (square root of) the bounds from 4.4 and 4.5 in the private case and in the non-private case, respectively.

□

4.5.2.2 Private Policy Optimization

In the last section, we showed that private model training impacts the reliability of our model for evaluating policies due to an increased dynamics error, which can lead to misjudging the quality of a policy in the true environment.

As in MOPO [Yu et al., 2020] and MOReL [Kidambi et al., 2020], we handle this by penalizing the reward with a measure of the uncertainty of the model, denoted $u : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$. Therefore, if the model is believed to be unreliable at a given state-action pair (s, a) (*i.e.*, large $u(s, a)$), the possibly over-estimated reward will be corrected as:

$$\tilde{r}(s,a) = \hat{r}(s,a) - \lambda \cdot u(s,a) \ , \tag{4.10}$$

[1] Indeed, for $f(x)$ positive, for any $x \geq x_0$, $|f(x)| = f(x) \leq M' \times g(x)$, then, for any $x \geq x_0$, $\sqrt{f(x)} = |\sqrt{f(x)}| \leq \sqrt{M'} \times \sqrt{g(x)} = M \times \sqrt{g(x)}$

Algorithm 10 Private Model-Based Optimization with SAC

Input: private model $\hat{M} = (\hat{P}, \hat{r})$, empty replay buffer $\mathcal{B}$, uncertainty estimator $u \in \{u_{\text{MA}}, u_{\text{MPD}}\}$
Output: private policy $\hat{\pi}^{\text{DP}}$
Initialize policy parameters ξ, Q-function parameters ω_1, ω_2 and target parameters $\omega_{\text{targ},1}, \omega_{\text{targ},2}$
for epoch $e \in [\![1, E]\!]$ **do**
  while episode is not terminated **do**
    Observe state s and select action $a \sim \pi_\xi(\cdot|s)$
    Execute a in the pessimistic MDP $\tilde{\mathcal{M}}$ and observe next state $s' \sim \hat{P}(\cdot|s,a)$, reward $r \sim \hat{r}(s,a) - \lambda u(s,a)$ and done signal d
    Store (s, a, r, s', d) in replay buffer $\mathcal{B}$
    if time to update **then**
      Sample a batch of transitions $B = \{(s, a, r, s', d)\}$ from buffer $\mathcal{B}$
      Compute targets for Q-functions:

$$y(r, s', d) = r + \gamma(1-d)\left(\min_{i=1,2} Q_{\omega_{\text{targ},i}}(s', \tilde{a}') - \alpha \log \pi_\xi(\tilde{a}'|s')\right), \quad \tilde{a}' \sim \pi_\xi(\cdot|s') \ .$$

      Update Q-functions by one step of gradient descent using:

$$\nabla_{\omega_i} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_{\omega_i}(s,a) - y(r,s',d)\right)^2, \quad \text{for } i = 1, 2.$$

      Update policy by one step of gradient ascent using:

$$\nabla_\xi \frac{1}{|B|} \sum_{s \in B} \left(\min_{i=1,2} Q_{\omega_i}(s, \tilde{a}_\zeta(s)) - \alpha \log \pi_\xi\left(\tilde{a}_\zeta(s)|s\right)\right), \quad \tilde{a}_\zeta(s) \sim \pi_\xi(\cdot|s).$$

      Update target networks with: $\omega_{\text{targ},i} \leftarrow \rho \omega_{\text{targ},i} + (1-\rho)\omega_i, \quad \text{for } i = 1, 2 \ .$
    end if
  end while
  Evaluate π_ξ is the true environment $\mathcal{M}$.
end for

where λ is an hyperparameter. u is typically derived from a deviation measure over the ensemble of models: Yu et al. [2020] consider the maximum aleatoric uncertainty

$$u_{\text{MA}}(s,a) = \max_{i \in [\![1,N]\!]} \|\Sigma_{\psi_i}(s,a)\|_F \ ,$$

while Kidambi et al. [2020] consider the maximum pairwise difference

$$u_{\text{MPD}}(s,a) = \max_{i,j \in [\![1,N]\!]} \|\mu_{\phi_i}(s,a) - \mu_{\phi_j}(s,a)\|_2 \ .$$

The policy is then optimized under the resulting pessimistic MDP $\widetilde{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \hat{P}, \tilde{r}, \gamma, \rho_0)$.

We showed earlier that private training affects only model error. Interestingly Lu et al. [2022] find that the uncertainty measures proposed in the literature are more strongly correlated with model error than with distribution shift. Based on this, we believe that existing uncertainty measures are well-suited to mitigate the diminished reliability of the model under private training, as they will effectively capture the increased error. We then use and compare both u_{MA} and u_{MPD} and find that neither is consistently superior (see Section 4.5.3.6). However, we observe that the choice of estimator can affect performance on a specific task.

Given a choice of uncertainty estimator $u \in \{u_{\text{MA}}, u_{\text{MPD}}\}$, the policy is optimized with Soft Actor-Critic (SAC, Haarnoja et al. [2018]) under the pessimistic private MDP $\widetilde{\mathcal{M}} = (\mathcal{S}, \mathcal{A}, \hat{P}, \tilde{r}_u, \gamma, \rho_0)$, with $\tilde{r}_u = \hat{r}(s,a) - \lambda \cdot u(s,a)$. Offline model-based methods typically mix real offline data from D with model data during policy learning (in MOPO, for instance, each batch contains 5% of real data). Here, however, we learn the policy exclusively from model data to avoid incurring privacy loss beyond what is needed to train the model, and thus control the privacy guarantees. Algorithm 10 provides a pseudo-code for SAC policy optimization in the pessimistic private MDP.

Using the post-processing property of DP, we can now state (Theorem 4.5.4) that, given the (ϵ, δ)-TDP model $\hat{M}$ learned as described in Section 4.5.1 and the resulting private pessimistic MDP $\widetilde{\mathcal{M}}$, the policy learned with model-based policy optimization under $\widetilde{\mathcal{M}}$ is also (ϵ, δ)-TDP.

Theorem 4.5.4. (ϵ, δ)-TDP guarantees for PriMORL. *Given an (ϵ, δ)-TDP model $(\hat{P}, \hat{r})$ learned with Algorithm 8, the policy obtained with model-based policy optimization (Algorithm 10) under the resulting private pessimistic MDP $\widetilde{\mathcal{M}}$ is (ϵ, δ)-TDP.*

Proof. First, we establish that the pessimistic MDP $\widetilde{M}$ is private for $u \in \{u_{\text{MA}}, u_{\text{MPD}}\}$. By Theorem 4.5.1, both the mean estimators $\{\mu_{\phi_i}\}_{i=1}^N$ and covariance estimators $\{\Sigma_{\psi_i}\}_{i=1}^N$ are private. Therefore, both uncertainty estimators $u_{\text{MA}}(s,a) = \|\Sigma_{\psi_i}(s,a)\|_F$ and $u_{\text{MPD}}(s,a) = \max_{i,j} \|f_{\phi_i}(s,a) - f_{\phi_j}(s,a)\|_2$, as data-independent transformations of the above quantities, are also private thanks to the post-processing property of DP. Therefore, the pessimistic model $\widetilde{M}$ remains (ϵ, δ)-TDP.

Now, we can think of SAC model-based policy optimization (Algorithm 10) as an abstract, randomized function h_Π, that takes as input $\hat{M}$ and outputs a policy $\hat{\pi}$. Furthermore, let h_M denote the mechanism that takes as input the private offline dataset D and outputs the private pessimistic model $\hat{M}$, and which is (ϵ, δ)-TDP following 4.5.1. We observe that $h = h_\Pi \circ h_M$, where h is the global offline RL algorithm which is the object of Definition 4.4.1. Since SAC only uses data from the model, h_Π is independent of the private offline data D. In other words, h_Π is a data-independent transformation of the private mechanism h_M. Thanks again to the post-processing property of differential privacy, h is also (ϵ, δ)-TDP. □

Table 4.2: Description of the tasks and datasets

	CartPole	Pendulum	HalfCheetah
Origin	Custom	Custom	D4RL
Observation space $\mathcal{S}$	$\mathbb{R}^5$	$\mathbb{R}^3$	$\mathbb{R}^{17}$
Action space $\mathcal{A}$	$[-1, 1]$	$[-2, 2]$	$[0, 1]^6$
Nb. of episodes K	30,000	30,000	2,003

4.5.3 Experiments

In this section, we empirically assess PriMORL in several continuous control tasks: CartPole-Balance and CartPole-Swingup from the DeepMind Control Suite [Tassa et al., 2018], Pendulum from OpenAI's Gym [Brockman et al., 2016], and MuJoCo's HalfCheetah [Wawrzynski, 2009].

4.5.3.1 Tasks and Datasets

Table 4.2 summarizes important aspects of the different tasks and corresponding offline datasets, with further details provided below.

Presentation of the tasks. CartPole-Swingup requires to swing up then balance an unactuated pole by applying forces on a cart at its base, while CartPole-Balance only requires keeping balance. Pendulum involves controlling an inverted pendulum by applying a torque to keep it upright and balanced over 200 steps. For this task, we normalize the episodic return to obtain a normalized score between 0 and 1000, using the following formula: $s_{\text{normalized}} = \frac{s-(-1500)}{0-(-1500))}$. HalfCheetah is another, higher-dimensional, continuous control task from OpenAI's Gym [Brockman et al., 2016] based on the physics engine MuJoCo [Todorov et al., 2012] where we move forward a 2D cat-like robot by applying torques on its joints. In all those tasks, the duration of an episode is 1,000 steps.

Offline datasets. As the existing SwingUp offline benchmark from Gülçehre et al. [2020] is very small ($K = 40$), we build our own dataset with 30,000 trajectories (*i.e.*, 30M steps). We follow the same approach for Balance and Pendulum for which we are not aware of existing offline benchmarks. Section 4.5.3.5 below discusses why the size of the dataset is a key issue in differentially private offline RL, justifying such large datasets. To collect our offline dataset for CartPole and Pendulum, we used DDPG [Lillicrap et al., 2016], a model-free RL algorithm for continuous action spaces. We ran 600 independent runs of $50,000$ steps each for Cartpole-Balance, 150 independent runs of $200,000$ steps each for Cartpole-SwingUp, and 6 independent runs of $1M$ steps each for Pendulum. We collect all training episodes to ensure a correct mix between random, medium and expert

Table 4.3: Privacy-specific parameters for the computation of the privacy budgets $\epsilon := \epsilon^{MA}(z, T, q, \delta)$. S does not factor into the privacy budget calculations, but is included here as a privacy-related parameter.

		S	z	T	q	δ	ϵ
Pendulum	Low	10^{-2}	0.35	7.10^3	10^{-3}	10^{-5}	22.3
	High	10^{-2}	0.52	7.10^3	10^{-3}	10^{-5}	**5.1**
Balance	Low	10^{-2}	0.25	7.10^3	10^{-3}	10^{-5}	85.0
	High	10^{-2}	0.45	7.10^3	10^{-3}	10^{-5}	**8.2**
Swingup	Low	10^{-2}	0.25	10.10^3	10^{-3}	10^{-5}	94.2
	High	10^{-2}	0.38	10.10^3	10^{-3}	10^{-5}	**17.0**

Table 4.4: PriMORL configurations.

Variant	Trajectory-Level Ens. Training	Clip	Noise	DP
No Clip	✓	✗	✗	$\epsilon = \infty$
No Privacy	✓	✓	✗	$\epsilon = \infty$
Low, High	✓	✓	✓	$\epsilon < \infty$

episodes, hence following the philosophy of D4RL [Fu et al., 2020]. For HalfCheetah, we directly use the Medium-Expert dataset ($K = 2,003$) from the D4RL benchmark [Fu et al., 2020].

4.5.3.2 Evaluation and Baselines

Following common practice, we evaluate the offline policies by running them in the real environment. We aim to assess the policy's performance degradation when varying the privacy level, as DP training may negatively affect it. Therefore, we compare PriMORL to MOPO, a close non-private baseline, which is a standard approach in the DP literature. Moreover, we consider different configurations for PriMORL, outlined in Table 4.4. The No Privacy variant, without noise ($z = 0$), isolates the impact of trajectory-level model training on performance. The two private variants ($\epsilon < \infty$) PriMORL Low and PriMORL High correspond to different noise multipliers, allowing us to explore different privacy-performance trade-offs.

4.5.3.3 Implementation

This section provides additional details regarding the implementation of our algorithm, including architecture choices and training hyperparameters.

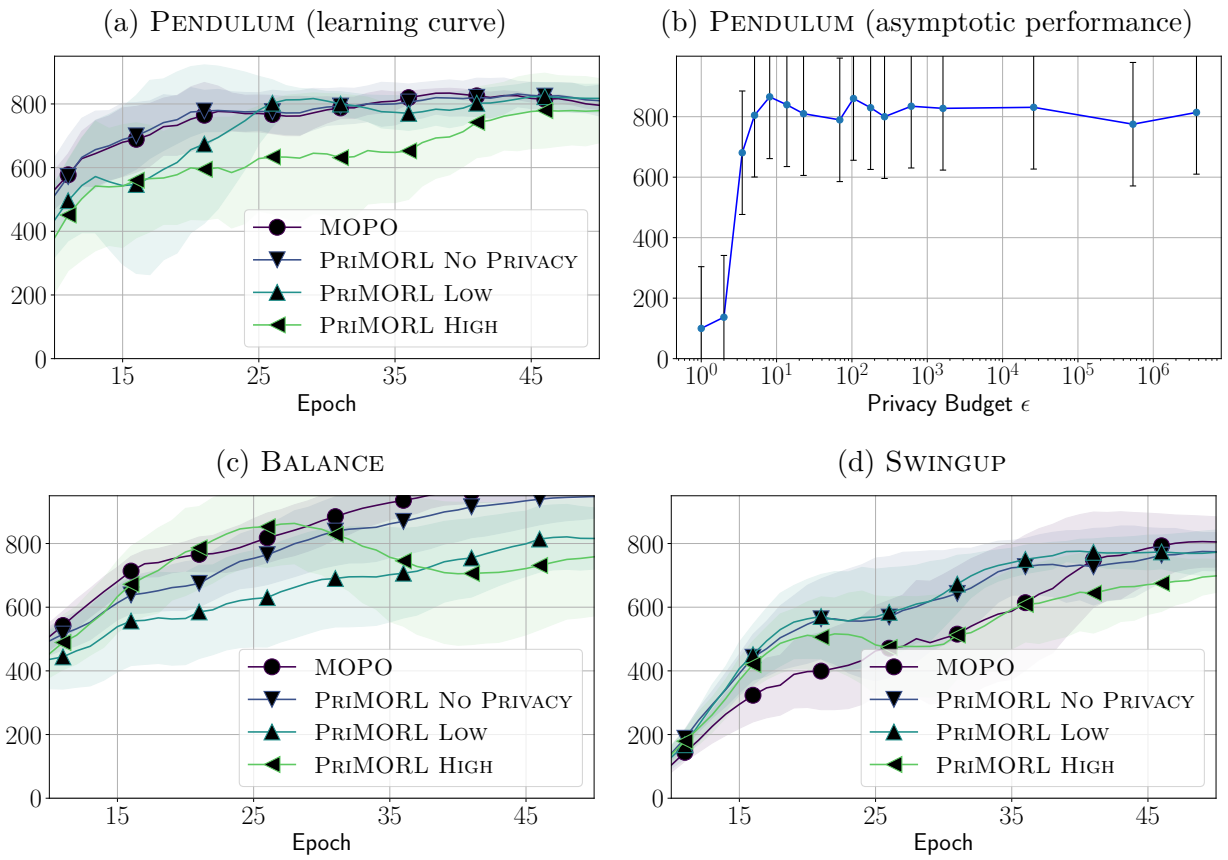


Figure 4.3: Pendulum (*top*): learning curves (*left*) and asymptotic perf. as a function of ϵ. CartPole (*bottom*): learning curves for Balance (*left*) and Swingup (*right*). y-axis is the average episodic return over 10 episodes per SAC training epoch. Mean performance and 95% confidence intervals are computed by re-training the model and the policy from scratch on at least 5 random seeds to assess the stability of the full training process.

Architecture and training details. Table 4.5 describes the model architecture for each task, while Table 4.6 summarizes the training hyperparameters used in the experiments for both model learning and privacy optimization. Before model training, we split the offline dataset into two parts: a train set used to train the model, and a test set used to track model performance. We consider the test set public so that this operation does not involve additional privacy leakage. The split is made by episode (instead of by transitions), so that the test set contains 1% of the episodes for CartPole and Pendulum and 20% for HalfCheetah. To tune the clipping norm, we set $z = 0$ and progressively decreased S until it started to adversely affect performance. Moreover, we set the sampling ratio q so that a few dozen episodes are randomly selected at each step, which proved to work best in our experiments, which correspond to $q = 10^{-3}$ for CartPole and Pendulum. The model is trained until convergence using *early stopping*. Test set prediction error is used to track model improvement. For SAC training, the real-to-model ratio r_{real} is zero, meaning that SAC is trained using only simulated data from the model, and does not access any data from the offline dataset. We do not report the privacy loss resulting from hyperparameter tuning, although we recognize its importance in real-world applications.

Table 4.5: Model architecture details. Each model in the ensemble is approximated with a deep neural network with Swish activation functions and decaying weights. Models take as input a concatenation of the current state s and the taken action a and predict the difference between the next state s' and the current state s along with the reward r.

	CartPole	Pendulum	HalfCheetah
Model input dimension	6	4	23
Model output dimension	6	4	18
Model hidden layers	2	2	4
Neurons per layer	128	64	200
Weight decay	✓	✓	✓
Activation functions	SWISH	SWISH	SWISH
Ensemble size N	5	3	7

Privacy parameters. Table 4.3 summarizes the privacy-specific parameters, and in particular the hyperparameters used to compute the privacy budgets ϵ with the moments accountant method from Abadi et al. [2016]. Privacy budgets are all computed for $\delta = 10^{-5}$, *i.e.* less than K^{-1} as recommended in the literature. ϵ also depends on the noise multiplier z, the number of training rounds T and the sampling ratio q. Since we use early stopping and the different training runs have different durations, we use the average number of training rounds in the privacy budget computations.

For PriMORL High, the value of the noise multiplier z is chosen by incrementally increasing z until policy performance drops below acceptable levels. The corresponding ϵ is therefore roughly the best privacy budget we can obtain while keeping acceptable policy performance. The value for PriMORL Low is chosen to provide a weaker level of privacy that typically yields higher policy performance, illustrating the trade-off between the strength of the privacy guarantee and the performance.

4.5.3.4 Main Results

CartPole and Pendulum. We present empirical results on Balance, SwingUp and Pendulum for PriMORL and baselines in Table 4.7 and Figure 4.3. These results exhibit a well-expected trade-off: performance tends to degrade with stronger privacy guarantees (*i.e.*, smaller ϵ values), as the model training gets perturbed with larger levels of noise. Moreover, private model training makes the policy performance less stable over several runs, which is also expected since differential privacy adds another source of randomness during training. We notice that noise is not the sole factor that negatively impacts performance, as suggested by the gap between MOPO and PriMORL No Privacy: gradient clipping and trajectory-level training also contribute to performance degradation. In some cases, a small amount of DP noise might actually be beneficial, acting as a kind of regularization, as in Swingup and Pendulum. Despite this trade-off, private agents trained with PriMORL

Table 4.6: Training and Hyperparameters details

	CartPole	Pendulum	HalfCheetah
Test set size	$1\% \times K$	$1\% \times K$	$10\% \times K$
Early stopping	3 patience = 10	3 patience = 10	3 patience = 5
Sampling ratio q	10^{-3}	10^{-3}	10^{-2}
Model local epochs E	1	1	1
Model batch size B	16	16	16
Model LR η	10^{-3}	10^{-3}	10^{-3}
Clipping Strategy	Flat	Per-Layer	Per-Layer
SAC LR	3.10^{-4}	3.10^{-4}	3.10^{-4}
Rollout length H	20	30	5
Reward penalty λ	2.0	2.0	1.0
Auto-α	✓	✓	✓
Target entropy H	-3	-3	-3
Uncertainty estimate	u_{MPD} (Bal.), u_{MA} (Swi.)	u_{MPD}	u_{MA}

Table 4.7: Asymptotic episodic return (averaged over the last 10 epochs) and the corresponding theoretical upper bound on ϵ, as computed from the hyperparameters z, q, T and δ using the moments accountant (see Table 4.3). Average performance and 95% confidence intervals are computed by re-training the model and the policy from scratch on at least 5 random seeds. In **bold**, ratio to the non-private baseline MOPO.

	Pendulum		CartPole-Balance		CartPole-Swingup	
Method	ϵ	Return	ϵ	Return	ϵ	Return
MOPO	∞	795.9 ± 6.5	∞	976.3 ± 26.8	∞	804.9 ± 89.6
PriMORL No Priv.	∞	810.4 ± 27.5 **(101.8%)**	∞	947.5 ± 68.3 **(97.1%)**	∞	774.1 ± 81.7 **(96.17%)**
PriMORL Low	22.3	817.4 ± 21.7 **(102.7%)**	85.0	815.8 ± 97.2 **(83.6%)**	94.2	772.4 ± 73.9 **(95.96%)**
PriMORL High	5.1	778.9 ± 53.5 **(97.9%)**	8.2	758.2 ± 187.2 **(77.7%)**	17.0	698.3 ± 57.5 **(86.75%)**

remain competitive with MOPO for ϵ in the 10^1 to 10^2 range. For Pendulum, we also plot policy performance against ϵ, and observe even no performance degradation until ϵ reaches the 1 to 10 range.

The realized privacy budgets ϵ, although weak theoretically, align with the realistic and widely used goal of achieving $\epsilon \leq 10$ in deep learning applications [Ponomareva et al., 2023]. As discussed in Chapter 2, several factors may account for this gap. Ponomareva et al. [2023] point out the overly restrictive assumptions on the adversary side, which may yield unnecessarily pessimistic privacy bounds. In offline RL especially, the definition of DP assumes the adversary only has to discriminate between two precise neighboring datasets D and $D' = D \cup \{\tau\}$ as well as the release of all gradients, whereas in practice the adversary faces the much harder task of reconstructing a high-dimensional trajectory based on the output policy and limited side information only. This further suggests that such ϵ's can

Table 4.8: Results for HalfCheetah Medium-Expert. Return is the return of the SAC policy evaluated over 10 episodes at the end of each training epoch, averaged over the last 20 epochs.

Method	z	Return
MOPO	0.0	10931 ± 1326
PriMORL No Clip	0.0	7062 ± 2230
PriMORL No Noise	0.0	8792 ± 2053
PriMORL	$z = 1.10^{-4}$	9729 ± 2018
	$z = 1.10^{-3}$	3697 ± 1465

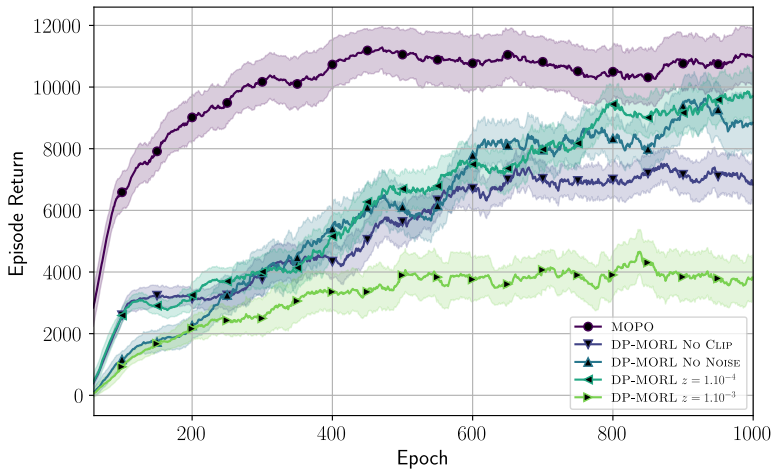


Figure 4.4: Learning curves for the SAC policy on HalfCheetah (*right*). Policy performance (episodic return) is evaluated in the true MDP at the end of each training epoch, over 10 evaluation episodes with different random seeds.

provide adequate privacy protection in practice, which is also backed by recent work on empirical privacy auditing [Carlini et al., 2019, Ponomareva et al., 2022].

Moreover, such ϵ values are comparable to the values reported in existing work. In particular, Qiao and Wang [2023a], in a similar offline setting with TDP guarantees, achieve similar privacy-performance trade-offs. They also consider the "low privacy regime" with ϵ's approaching 50 for their best-performing variant, highlighting the trade-offs between privacy and performance. However, their results are obtained on much simpler, finite-horizon tasks. Therefore, we argue that our results are significantly stronger.

HalfCheetah. We further lead exploratory experiments on HalfCheetah, whose results are reported in Figure 4.4 and Table 4.8, to assess the behavior of PriMORL on higher-dimensional tasks. For this task, we use $S = 15.0$ and $q = 10^{-2}$. If PriMORL can train competitive policies with small enough noise levels — a tiny amount of noise like $z = 10^{-4}$

proving even beneficial, possibly acting as a kind of regularization —, we were not able to obtain reasonable ϵ's. Indeed, a noise multiplier as small as $z = 10^{-3}$ is enough to cause a significant decline in performance.

Although Ponomareva et al. [2023] argue that any finite ϵ is an improvement over a model with no formal privacy guarantees, opening the way to further quantifiable privacy improvements and possibly indicating a substantial decrease in the model's ability to memorize training data, such privacy guarantees are certainly not enough to provide adequate privacy protection. These results primarily suggest that scaling to higher-dimensional problems is particularly challenging in private RL. The latter could be expected based on the theoretical analysis led in Section 4.5.2.1, as DP training adds a dependence on the dimension d of the task in the valuation gap.

Nonetheless, we also point out that the size of the dataset for HalfCheetah is very limited, and argue that larger datasets with substantially more episodes would translate into more competitive privacy-performance trade-offs, as we develop in the next section.

4.5.3.5 Discussion: The Price of Privacy in Offline RL

Despite promising results on benchmarks previously unexplored in DP RL, we observe that achieving privacy is particularly costly in this setting. In addition to the notorious instability of RL methods, exacerbated by the introduction of DP noise, we argue that the limited size of typical offline RL datasets is a major hurdle. From a theoretical perspective, larger datasets would enable stronger privacy-performance trade-offs for two reasons:

- The privacy budget $\epsilon = \mathcal{O}(\frac{q\sqrt{T}}{z})$ scales with the sampling ratio q (*privacy amplification by subsampling* [Balle et al., 2018]);
- The noise magnitude σ is inversely proportional to $\mathbb{E}\left[|\mathcal{U}_t|\right] = qK$.

The privacy-performance trade-off would thus benefit from both small q — reducing the privacy budget ϵ — and large qK — decreasing noise levels and thus improving performance —, which are conflicting objectives for a fixed K. However, if we can access larger datasets of size $K' \gg K$, we can then find a K' large enough so that we can get both $q' \ll q$ and $q'K' \gg qK$ — achieving both much stronger privacy and better performance. **Therefore, with large datasets, we can simultaneously reduce noise levels and decrease the privacy budget ϵ by sub-sampling amplification, resulting in improved privacy trade-offs.** In fact, for fixed hyperparameters z, q, T and S, and the resulting privacy budget $\epsilon = \mathcal{O}(\frac{q\sqrt{T}}{z})$, the noise magnitude $\sigma = \frac{zS}{qK}$ can be made arbitrarily small by increasing K.

Another practical advantage of being able to leverage privacy amplification by subsampling, — *i.e.*, to decrease q — is that it makes the cost of additional training iterations negligible in the overall privacy budget, as observed in McMahan et al. [2017]. Since the tasks tackled in the previous section required a lot of iterations to reach convergence (empirically, thousands of iterations for CartPole and tens of thousands of iterations for

Table 4.9: Ablation study investigating the impact of using different clipping methods (*flat clipping* (FC) and *per-layer clipping* (LC)) and different uncertainty estimates (u_{MA} and u_{MPD}). For instance, MA + LC means the model has been train with *per-layer clipping* and the policy optimized under uncertainty estimate u_{MA}. We report policy performance as the mean episodic return over 10 evaluation episodes, averaged over the last 10 epochs of policy optimization. Average performance and 95% confidence intervals are computed on at least 3 seeds.

	z	MA + LC	MPD + LC	MA + FC	MPD + FC
Pendulum	0.35	734.5 ± 28.9	$\mathbf{817.4 \pm 21.7}$	638.9 ± 91.7	757.9 ± 76.5
	0.52	750.1 ± 75.97	$\mathbf{778.9 \pm 53.5}$	612.4 ± 76.8	723.0 ± 47.0
Balance	0.25	785.4 ± 77.9	792 ± 85.8	$\mathbf{819.9 \pm 66.1}$	815.8 ± 97.2
	0.45	749.0 ± 115.6	738.5 ± 85.7	722.7 ± 105.7	$\mathbf{758.2 \pm 187.2}$
Swingup	0.25	711.4 ± 90.6	704.2 ± 46.2	772.4 ± 73.9	$\mathbf{777.1 \pm 35.0}$
	0.38	536.1 ± 112.8	575.4 ± 89.7	$\mathbf{698.3 \pm 57.5}$	590.8 ± 34.8

HalfCheetah), and as DP noise notoriously tends to slow down training, the benefits of increasing the size of the dataset to leverage smaller sampling ratios are significant.

These arguments underscore the need for large datasets in offline RL in order to achieve strong privacy-performance trade-offs. As an example, McMahan et al. [2018] consider datasets with 10^6 to 10^9 users to train DP recurrent language models, and this is arguably the main reason why they achieve strong formal privacy guarantees. For comparison, the classic RL Unplugged and D4RL benchmarks provide datasets with $K \approx 10^1$ to $K \approx 10^3$ datasets. This motivated us to self-collect large-scale datasets with tens of thousands of trajectories for our experiments on CartPole and Pendulum, but this still appear insufficient. Conducting experiments in deep offline RL with datasets of the same size as McMahan et al. [2018] would demand substantial computational resources. However, there may be real-world scenarios involving access to vast numbers of trajectories, making this case worthy of thorough investigation.

Overall, we argue that PriMORL, already capable of producing good policies with significant noise levels, has the potential to achieve stronger privacy guarantees provided access to large enough datasets.

4.5.3.6 Ablation Study: Clipping and Uncertainty Estimator

In Table 4.9, we compare performance for different combinations of clipping methods and uncertainty estimators. We can see that no combination is significantly superior overall, but these choices may impact privacy performance for a specific task. Preliminary results from this ablation led us to choose, for each task, the clipping strategy and the uncertainty estimate stated in Table 4.6.

4.6 Differentially Private Policy Gradient

The previous part addressed the offline RL setting with a model-based approach. We now turn to the online scenario, where we continuously interact with the environment; precisely, we focus on the setting where we alternate between training phases, where the agent learns from the most recent on-policy data, and exploitation phases, where the policy is served to users (*e.g.*, an LLM released *via* a public API) — as described in Algorithm 7. In this context, we develop policy gradient methods with trajectory-level DP updates, with an emphasis on trust-region methods. We first investigate, in Section 4.6.2, the theoretical properties of DP policy gradients. We characterize the inclusion of the gradient updates within a given trust region in the presence of DP noise, and derive theoretical insights into how to set the DP clipping norm for the noisy updates to remain within the trust region with high probability, providing solid theoretical foundations for our approach. Relaxing the theory, we then provide a practical PPO-like algorithm with formal, trajectory-level differential privacy guarantees (Section 4.6.3). Finally, in Section 4.6.4, we show empirically that our approach allows the introduction of differential privacy in policy gradient algorithms with limited impact on performance on previously unaddressed deep RL benchmarks.

4.6.1 Policy Gradient and Trust-Region Methods

We focus on the class of RL algorithms called *policy gradient* methods described in Section 4.4.1. These methods directly update π_θ, with the goal of maximizing $J(\theta) = J(\pi_\theta)$, based on the policy gradient theorem. They use sample estimates of the gradient $\Delta_\theta J(\theta)$:

$$\hat{g} = \hat{\mathbb{E}}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t\right] \ ,$$

where $\hat{A}_t$ is an estimator of the advantage function at step t.

A weakness of policy gradient methods is that small updates in the parameter space may lead to large policy shifts which can harm performance. Natural Policy Gradient (NPG) [Kakade, 2001] addresses this by rescaling the gradient with the inverse Fisher information matrix (FIM), $F(\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s)^T\right] \in \mathbb{R}^{d\times d}$, *i.e.*, using $\tilde{g} = F(\theta)^{-1}\hat{g}$, taking into account the geometry of the policy space. On the other hand, Trust Region Policy Optimization (TRPO) [Schulman et al., 2015] uses a hard constraint on the Kullback-Leibler (KL) divergence between the current and the new policy to prevent destructively large policy updates (*i.e.*, a *trust-region* constraint) while optimizing a surrogate loss:

$$\begin{aligned} \max_\theta L_{\theta_{\text{old}}}(\theta) &:= J(\theta) + \mathbb{E}_{s\sim\rho_{\theta_{\text{old}}}, a\sim\pi_\theta}\left[A^{\theta_{\text{old}}}(s,a)\right] \\ \text{s.t.} \quad & \mathbb{E}_{\pi_{\theta_{\text{old}}}}\left[\text{KL}(\pi_{\theta_{\text{old}}}, \pi_\theta)\right] \leq \alpha \ , \end{aligned}$$

Proximal Policy Optimization (PPO) [Schulman et al., 2017] follows the same logic as

TRPO while being significantly easier to implement. It optimizes a surrogate objective which appropriately clips $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$, the probability ratio between the current and the new policy:

$$L^{\text{PPO}}(\theta) = \mathbb{E}\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon))\right] \ .$$

Such algorithms are referred to as *trust-region* methods. In the following, we study differentially private policy gradient algorithms from the perspective of trust-region methods, laying the theoretical foundation for deep privacy-preserving RL in the online setting

4.6.2 A Theoretical Private Policy Gradient

4.6.2.1 Private Policy Gradient Updates with Trust Region

Our goal is to propose a differentially private policy update $\pi \leftarrow h(\pi, D)$ for the on-policy procedure depicted in Algorithm 7, in the the function approximation setting where the policy is parameterized by $\theta \in \Theta$, *i.e.*, $\pi := \pi_\theta$. As described in Chapter 2, DP-SGD privatizes the training of machine learning models by injecting noise in the gradients. Therefore, it is sensible to consider a *differentially private policy gradient* (DPPG) algorithm based on the following noisy policy gradient updates:

$$\theta = \theta_{\text{old}} + \eta \cdot \left(\hat{g}\left(D^{\theta_{\text{old}}}; \theta\right) + \xi\right) \ ,$$

where θ_{old} are the parameters of the policy at the current iteration (the one used to collect the data), $\hat{g}(D^{\theta_{\text{old}}}; \theta)$ is the sample gradient of $L(\theta) := L(\pi_\theta)$ estimated on on-policy data $D^{\theta_{\text{old}}}$, which we denote hereafter $\hat{g}$ for simplicity, and $\xi \sim \mathcal{N}(0, \sigma^2 \cdot I_d)$ is a zero-mean Gaussian noise. To guarantee differential privacy, the noise scale σ must be large enough to hide changes in $\hat{g}$'s magnitude between two neighbor datasets. Unfortunately, without further assumptions on $L(\theta)$, $\hat{g}$ is unbounded. This requires clipping the gradient $\hat{g}$ with some *clipping norm* S. Denoting $\bar{S} = 1/\max(\|\hat{g}\|/S, 1)$ the effective clipping factor (since we only scale down $\hat{g}$ if its norm is more than S), the resulting update $\Delta\theta = \theta - \theta_{\text{old}}$ is thus:

$$\Delta\theta = \eta \cdot \left(\bar{S}\hat{g} + \xi\right) \ . \tag{4.11}$$

Given a noise multiplier $z > 0$, if $\xi \sim \mathcal{N}(0, z^2 S^2 I_d)$, the update (4.11) is (ϵ, δ)-DP w.r.t. data D^{π_θ} for some $\epsilon := \epsilon(z, \delta)$.

Equation (4.11) shows that we can use the clipping norm S to control the size of the policy updates. Coincidentally, this is also the underlying idea behind trust-region methods, as described in Section 4.6.1: constrain two consecutive policies $\pi_{\theta_{\text{old}}}$ and π_θ to remain close in some sense. TRPO, for instance, uses the KL divergence as a measure of proximity. Building on this parallel between trust-region methods and (noisy) policy gradient updates with clipping, we argue that, for policy gradient methods, the introduction of differential privacy can be reduced to the computation of appropriate trust regions (through the clipping

norm S), thus avoiding the sacrifice of theoretical properties of the DP-less methods, which we develop in the next section.

In the following, we explore the theoretical properties of DP Policy Gradient algorithms based on the update rule (4.11). We first focus on the standard policy gradient seen as a trust-region method, before delving into TRPO. In particular, we derive insights about how we should set S, which plays a crucial role in accommodating the presence of noise in the policy updates.

4.6.2.2 Private Standard Policy Gradient

Although not explicitly a trust-region method, the standard policy gradient can be formulated as a trust-region update maximizing a first-order approximation of TRPO's loss $L_{\theta_{\text{old}}}(\theta)$ under an L_2 constraint [Schulman et al., 2015]:

$$\begin{aligned} \max_{\theta} L^{\text{PG}}(\theta) &:= \nabla_\theta L_{\theta_{\text{old}}}(\theta)^\top(\theta - \theta_{\text{old}}) \\ \text{s.t.} \quad & \frac{1}{2}\|\theta - \theta_{\text{old}}\|_2^2 \leq \alpha \ . \end{aligned} \tag{4.12}$$

In the following, we consider that we have access to the true gradient $g := \nabla_\theta L_{\theta_{\text{old}}}(\theta)$. We denote $\bar{g} = \bar{S}g$ the clipped gradient and perform noisy policy gradient updates as described in Equation (4.11) with:

$$\xi \sim \mathcal{N}(0, z^2 S^2 I_d) \ .$$

Trust Region constraint with noisy updates. First, we show that, choosing S appropriately, we are able to ensure that constraint (4.12) holds with high probability. We have:

$$\frac{1}{2}\|\theta - \theta_{\text{old}}\|_2^2 = \frac{\Delta\theta^\top \Delta\theta}{2} = \frac{\eta^2}{2}(\bar{g} + \xi)^\top(\bar{g} + \xi) \ .$$

This quadratic form in ξ follows a (scaled) non-central χ^2 distribution. Proposition 4.6.1 characterizes this distribution and provides an upper bound on the clipping norm S such that the noisy update remains within the trust region.

Proposition 4.6.1. *The L_2 trust-region size follows a scaled non-central χ^2 distribution:*

$$\frac{\Delta\theta^\top \Delta\theta}{2} = \frac{\eta^2 z^2 S^2}{2}\chi^2\left(d, \nu^2\right) \ , \tag{4.13}$$

with $\nu^2 = \|\bar{g}\|^2 / z^2 S^2 \leq z^{-2}$. Therefore, given $\alpha \geq 0$ and $\beta \in (0,1)$ and denoting $q_{1-\beta}^{\chi^2(d,\nu^2)}$ the

quantile of order $1-\beta$ *of* $\chi^2(d,\nu^2)$*, if we set:*

$$S \leq \frac{1}{\eta z}\sqrt{\frac{2\alpha}{q_{1-\beta}^{\chi^2(d,z^{-2})}}} \ ,$$

then $\mathbb{P}[\frac{\Delta\theta^\top\Delta\theta}{2} \leq \alpha] \geq 1-\beta$*, i.e., the DP policy update (4.11) remains within the* L_2 *trust region of size* α *with probability at least* $1-\beta$*.*

Proof. The (random) trust region size can be written as:

$$\frac{\Delta\theta^T\Delta\theta}{2} = X^TX = \sum_{i=1}^{d} X_i^2 \ ,$$

with $X = \frac{\eta}{\sqrt{2}}(\bar{g}+\xi)$. We have $X \sim \mathcal{N}_d(\frac{\eta}{\sqrt{2}}\bar{g}, \frac{\eta^2}{2}z^2S^2I_d)$. In particular, its marginals X_i's are i.i.d. with $X_i = \mathcal{N}(\mu_i,\sigma^2)$. with $\mu_i = \frac{\eta}{\sqrt{2}}\bar{g}_i$ and $\sigma^2 = \frac{\eta^2}{2}z^2S^2$. Because $X_i = \frac{\eta}{\sqrt{2}}zS\cdot\mathcal{N}(\frac{\bar{g}_i}{zS},1)$, we can reformulate this in terms of unit-variance normal variables $Z_i \sim \mathcal{N}(\frac{\bar{g}_i}{zS},1)$:

$$\frac{\Delta\theta^T\Delta\theta}{2} = \frac{\eta^2}{2}z^2S^2\sum_{i=1}^{d} Z_i^2 \ .$$

We can then apply the following result: if $(Z_1,...,Z_d)$ are d independent, normally distributed with mean μ_i and unit variance, then $\sum_{i=1}^d Z_i^2$ follow a non-central χ^2 distribution with d degrees of freedom and non-centrality parameter $\lambda = \sum_{i=1}^d \mu_i^2$, denoted $\chi^2(d,\lambda)$.

In our case, the non-centrality parameter is:

$$\lambda = \sum_{i=1}^{d}\frac{\bar{g}_i^2}{z^2S^2} = \frac{\|\bar{g}\|^2}{z^2S^2} \ ,$$

and therefore:

$$Z := \sum_{i=1}^{d} Z_i^2 \sim \chi^2\left(d, \frac{\|\bar{g}\|^2}{z^2S^2}\right) \ .$$

We can numerically compute the PDF, CDF and PPF of Z using, *e.g.*, `scipy.stats.ncx2` in Python. In particular, for $\beta \in (0,1)$, we can compute the quantile $q_{1-\beta}$ such that $\mathbb{P}\left(Z \leq q_{1-\beta}^{\chi^2(d,\|\bar{g}\|^2/z^{-2}S^{-2})}\right) = 1-\beta$.

Therefore, let us assume that we want the update to remain within a L_2 trust region of size α with probability $1-\beta$, *i.e.*, $\mathbb{P}(\frac{\Delta\theta^T\Delta\theta}{2} \leq \alpha) = 1-\beta$. We have:

$$\begin{aligned}\mathbb{P}(\frac{\Delta\theta^T\Delta\theta}{2} \leq \alpha) &= \mathbb{P}(\frac{\eta^2}{2}z^2S^2\cdot Z \leq \alpha)\\ &= \mathbb{P}(Z \leq \frac{2\alpha}{\eta^2z^2S^2}).\end{aligned}$$

From the definition of the quantile function, $\mathbb{P}(Z \leq q_{1-\beta}^{\chi^2(d,\|\bar{g}\|^2/z^{-2}S^{-2})}) = 1-\beta$. To derive an inequality involving S and the other parameters of the problem, we bound $\frac{\|\bar{g}\|^2}{z^2S^2} \leq \frac{1}{z^2}$ and use the fact that the function of $\lambda \mapsto q_x^{\chi^2(d,\lambda)}$ is increasing in λ: $q_{1-\beta}^{\chi^2(d,\|\bar{g}\|^2/z^{-2}S^{-2})} \leq q_{1-\beta}^{\chi^2(d,z^{-2})}$. Therefore $\mathbb{P}(Z \leq q_{1-\beta}^{\chi^2(d,\|\bar{g}\|^2/z^{-2}S^{-2})}) \leq \mathbb{P}(Z \leq q_{1-\beta}^{\chi^2(d,z^{-2})})$, and it follows that $\mathbb{P}(Z \leq q_{1-\beta}^{\chi^2(d,z^{-2})}) \geq 1-\beta$.

Equalizing $\frac{2\alpha}{\eta^2 z^2 S^2} = q_{1-\beta}^{\chi^2(d,z^{-2})}$ and deriving S, we have:

$$S \leq \frac{1}{\eta z}\sqrt{\frac{2\alpha}{q_{1-\beta}^{\chi^2(d,z^{-2})}}} \ .$$

Therefore, setting S like above, we have $\mathbb{P}(\frac{\Delta\theta^T\Delta\theta}{2} \leq \alpha) \geq 1-\beta$. □

Note that the quantiles $q_{1-\beta}^{\chi^2(d,\nu^2)}$ can be easily computed numerically. Moreover, computing the expectation $\mathbb{E}\left[\frac{\Delta\theta^\top\Delta\theta}{2}\right] = \frac{\eta^2}{2}\left(\|\bar{g}\|^2 + z^2S^2d\right)$ and using the Markov inequality, Proposition 4.6.2 provides a slightly more interpretable result:

Proposition 4.6.2. *Given $\alpha \geq 0$ and $\beta \in (0,1)$, if we set:*

$$S \leq \frac{1}{\eta}\sqrt{\frac{2\alpha\beta}{1+z^2d}} \ , \tag{4.14}$$

then $\mathbb{P}[\frac{\Delta\theta^\top\Delta\theta}{2} \leq \alpha] \geq 1-\beta$.

Proof. This result can be seen as a special case of the upper bound on S provided in Proposition 4.17 below, if we replace F with I_d. Therefore, we refer the reader to the proof of Proposition 4.17. Since $\max_i \sigma_i(I_d) = 1$ and $\mathrm{tr}(I_d) = d$, replacing F by I_d in Proposition 4.17 yields the desired result. □

As expected, $S \propto \sqrt{\alpha}$ relates to the size of the trust region, and the larger the probability to remain within the trust region (corresponding to a smaller β or a larger $q_{1-\beta}^{\chi^2(d,\nu^2)}$), the smaller we need to set S. Stronger DP guarantees (larger z) and higher-dimensional problems (larger d) also require more conservative updates, which is in line with the intuition. Of course, the value of S is also tied to the learning rate η, since the latter also determines the size of the update.

Objective gap. We showed that a small enough S keeps the update within the trust region with high probability. While this prevents a destructively large update, it tells us nothing about the direction of the update and whether we actually improve the objective. We therefore consider one step of update and denote $\theta^\star = \theta_{\text{old}} + \eta g$ the parameter obtained with a non-private policy gradient update, and $\tilde{\theta} = \theta_{\text{old}} + \eta(\bar{g} + \xi)$ the parameter obtained following the DP update. We are interested in the objective gap:

$$L^{\mathrm{PG}}(\tilde{\theta}) - L^{\mathrm{PG}}(\theta^\star) = g^\top(\tilde{\theta} - \theta^\star) \ .$$

Proposition 4.6.3 shows that, if we set S small enough, we can control the objective gap with high probability. Similarly to previous results, the upper bound is inversely correlated to the learning rate and noise multiplier.

Proposition 4.6.3. *Given $\lambda > 0$ and $\beta_2 \in (0,1)$, if we set:*

$$S \leq \frac{\lambda}{\eta z \|g\|}\sqrt{\frac{\beta_2}{1-\beta_2}} \ . \tag{4.15}$$

then, denoting $K(\lambda) = (\eta(1-\bar{S})\|g\| + \lambda)$:

$$\mathbb{P}\left[L^{PG}(\tilde{\theta}) \geq L^{PG}(\theta^\star) - K(\lambda)\right] \geq 1 - \beta_2 \ .$$

Proof. We are interested in the following quantity:

$$L^{\mathrm{PG}}(\tilde{\theta}) - L^{\mathrm{PG}}(\theta^\star) = g^T(\tilde{\theta} - \theta^\star) \ .$$

Since $\tilde{\theta} - \theta^\star = \eta(\bar{g} + \xi) - \eta g = \eta(\bar{S}g + \xi) - \eta g = \eta[(\bar{S} - 1)g + \xi]$, we have:

$$\begin{aligned} L^{\mathrm{PG}}(\tilde{\theta}) - L^{\mathrm{PG}}(\theta^\star) &= \eta \cdot g^T[(\bar{S} - 1)g + \xi] \\ &= \eta g^T \xi + \eta(\bar{S} - 1)\|g\|^2 \ , \end{aligned}$$

with $\eta g^T \xi \sim \mathcal{N}(0, \sigma^2)$.

Therefore:

$$L^{\mathrm{PG}}(\theta^\star) - L^{\mathrm{PG}}(\tilde{\theta}) = -\eta g^T \xi + \eta(1 - \bar{S})\|g\|^2 \ .$$

By symmetry, $-\eta g^T \xi$ has the same distribution as $\eta g^T \xi$. Therefore, denoting $\sigma^2 = \eta^2 z^2 S^2 \|g\|^2$ and applying Cantelli's inequality for $\lambda > 0$:

$$\begin{aligned} \mathbb{P}\left(L^{\mathrm{PG}}(\theta^\star) - L^{\mathrm{PG}}(\tilde{\theta}) - \eta(1-\bar{S})\|g\|^2 \geq \lambda\right) &= \mathbb{P}\left(L^{\mathrm{PG}}(\tilde{\theta}) \leq L^{\mathrm{PG}}(\theta^\star) - (\eta(1-\bar{S})\|g + \lambda)\right) \\ &= \mathbb{P}\left(L^{\mathrm{PG}}(\tilde{\theta}) \leq L^{\mathrm{PG}}(\theta^\star) - K(\lambda)\right) \\ &\leq \frac{\sigma^2}{\sigma^2 + \lambda^2} \ . \end{aligned}$$

We want S such that $\sigma^2/(\sigma^2 + \lambda^2) \leq \beta_2 \iff \sigma^2 \leq \lambda^2 \cdot \beta_2/(1-\beta_2)$, which gives us:

$$S \leq \frac{\lambda}{\eta z \|g\|}\sqrt{\frac{\beta_2}{1-\beta_2}} \ .$$

□

Therefore, by computing appropriate trust regions — that is, by setting S small enough that (4.15) and either (4.13) or (4.14) hold — our DP policy gradient preserves the theoretical properties of the non-private policy gradient.

4.6.2.3 Private TRPO

The issue with the previous approach is that the trust region does not take into account the geometry of the policy space: even policy updates that are small within the parameter space can result in large policy shifts, potentially hurting performance. TRPO addresses this by formulating a trust region constraint with respect to the (expected) KL divergence between $\pi_{\theta_{\text{old}}}$ and π_θ:

$$\mathbb{E}_{\pi_{\theta_{\text{old}}}}\left[\text{KL}(\pi_{\theta_{\text{old}}}, \pi_\theta)\right] \leq \alpha \ .$$

where the expectation is taken on the states encountered when rolling policy $\pi_{\theta_{\text{old}}}$. Moreover, in the neighborhood of θ_{old}, the KL-divergence can be reasonably approximated using the Fisher information matrix of $\pi_{\theta_{\text{old}}}$, $F := F(\theta_{\text{old}})$:

$$\mathbb{E}\left[KL(\pi_{\theta_{\text{old}}}, \pi_\theta)\right] \approx \frac{1}{2}(\theta - \theta_{\text{old}})^\top F(\theta - \theta_{\text{old}}) := \frac{1}{2}\Delta\theta^\top F \Delta\theta \ .$$

Based on this approximation, we can formulate the (approximate) TRPO trust-region constraint in the DP setting, using $\Delta\theta = \eta(\bar{g} + \xi)$,

$$\frac{\eta^2}{2}(\bar{g} + \xi)^\top F(\bar{g} + \xi) \leq \alpha \ . \tag{4.16}$$

As with standard PG, Proposition 4.6.4 characterizes the distribution of (4.16) and derives an upper bound on S [2]:

Proposition 4.6.4. *The size of the (approximated) KL trust region follows a generalized non-central χ^2 distribution:*

$$\frac{1}{2}\Delta\theta^\top F \Delta\theta = \frac{\eta^2}{2} z^2 S^2 \sum_{i=1}^{d} \sigma_i(F) \chi^2(1, \nu_i^2) \ ,$$

where $F = P^\top \text{diag}(\sigma_1(F), ..., \sigma_d(F)) P$ is the spectral decomposition of F and $\nu = (\nu_1, ..., \nu_d) = \frac{1}{zS} P^\top \bar{g}$.

Moreover, given $\alpha \geq 0$ and $\beta \in (0, 1)$, if we set:

$$S \leq \frac{1}{\eta} \sqrt{\frac{2\alpha\beta}{\max_i \sigma_i(F) + z^2 \text{tr}(F)}} \ , \tag{4.17}$$

then $\mathbb{P}[\frac{\Delta\theta^\top F \Delta\theta}{2} \leq \alpha] \geq 1 - \beta$.

[2] If we want to use this bound in practice and need to estimate F or its eigenvalues on private data, then this step should also be privatized. We consider this beyond the scope of this work.

Proof. Distribution. The (random) trust region size can be written as:

$$\widehat{D_{\mathrm{KL}}} = \frac{1}{2}\Delta\theta^T F \Delta\theta = X^T A X \ ,$$

with $X = \frac{\eta}{\sqrt{2}}(\bar{g} + \xi) \sim \mathcal{N}_d(\mu, \Sigma)$ ($\mu = \frac{\eta}{\sqrt{2}}\bar{g}$ and $\Sigma = \frac{\eta^2}{2} z^2 S^2 I_d$) and $A = F$. In particular, its marginals X_i's are i.i.d. with $X_i = \mathcal{N}(\mu_i, \sigma^2)$. with $\mu_i = \eta\bar{g}_i$ and $\sigma^2 = \eta^2 z^2 S^2$.

With these notations, we can apply the following results from Paolella [2019] (Appendix A.1): *for $Y = X^T A X$ with $\mathcal{N}_d(\mu, \Sigma)$, using the spectral decomposition $\Sigma^{1/2} A \Sigma^{1/2} = P \Lambda P^T$ (with P an orthogonal matrix and $\Lambda = diag(\lambda_1, ..., \lambda_d)$). Then, denoting $\nu = P^T \Sigma^{-1/2} \mu = (\nu_1, ..., \nu_d)^T$, $Y \sim \sum_{i=1}^d \lambda_i W_i^2$ with $W_i \sim \chi^2(1, \nu_i^2)$, where $\chi^2(1, \nu_i^2)$ is a non-central χ^2 distribution with one degree of freedom and non-centrality parameter λ_i.*

We have $\Sigma^{1/2} A \Sigma^{1/2} = \frac{\eta^2}{2} z^2 S^2 F = P(\frac{\eta^2}{2} z^2 S^2 \mathrm{diag}(\sigma_1(F), ..., \sigma_d(F)))P^T$, where the $\sigma_i(F)$'s are the eigenvalues/singular values of F. Then $\nu = P^T \cdot (\frac{\eta}{\sqrt{2}} zS)^{-1} I_d \cdot \frac{\eta}{\sqrt{2}} \bar{g} = \frac{1}{zS} P^T \bar{g}$. Therefore:

$$\frac{1}{2}\Delta\theta^T F \Delta\theta \sim \frac{\eta^2}{2} z^2 S^2 \sum_{i=1}^{d} \sigma_i(F) \chi^2(1, \nu_i^2), \quad . \tag{4.18}$$

Clipping norm bound. We have:

$$\frac{1}{2}\Delta\theta^T F \Delta\theta = \frac{\eta^2}{2}(\bar{g} + zS \cdot \zeta)^T F (g + zS \cdot \zeta) \tag{4.19}$$

$$= \frac{\eta^2}{2}(\bar{g}^T F \bar{g} + 2 \cdot zS \cdot \zeta^T F \bar{g} + (zS)^2 \cdot \zeta^T F \zeta) \ . \tag{4.20}$$

$$= a_0 + \sum_i b_i \zeta_i + \sum_{i,j} c_{ij} \zeta_i \zeta_j \ . \tag{4.21}$$

We now compute the expectation of the above random variable. Because ζ has zero mean, and thus $\mathbb{E}[\zeta^T F \bar{g}] = 0$:

$$\mathbb{E}\left[\widehat{D_{\mathrm{KL}}}\right] = \frac{\eta^2}{2}\left(\bar{g}^T F \bar{g} + (zS)^2 \cdot \mathbb{E}[\zeta^T F \zeta]\right)$$

For the second term, denoting $F = (f_{ij})_{1 \leq i,j \leq d}$, we have:

$$\begin{aligned}
\mathbb{E}[\zeta^T F \zeta] &= \mathbb{E}\left[\sum_{i,j} f_{ij}\zeta_i\zeta_j\right] \\
&= \sum_{i,j} f_{ij}\mathbb{E}[\zeta_i\zeta_j] \\
&= \sum_{i \neq j} f_{ij}\mathbb{E}[\zeta_i]\mathbb{E}[\zeta_j] + \sum_i f_{ii}\mathbb{E}[\zeta_i^2] \\
&= 0 + \sum_i f_{ii}\mathbb{V}[\zeta_i] \\
&= \mathrm{tr}(F).
\end{aligned}$$

Therefore, we have:

$$\mathbb{E}\left[\frac{\eta^2}{2}\tilde{g}^T F \tilde{g}\right] = \frac{\eta^2}{2} \cdot \left(\bar{g}^T F \bar{g} + (zS)^2 \cdot \mathrm{tr}(F)\right) \ . \tag{4.22}$$

We also have the following bound on $\bar{g}^T F \bar{g}$:

$$\begin{aligned}
\bar{g}^T F \bar{g} &= \|F^{1/2}\bar{g}\|_2^2 \\
&\leq \|F^{1/2}\|_{\mathrm{op}}^2 \cdot \|\bar{g}\|_2^2 \\
&= \max_i \sigma_i(F^{1/2})^2 \cdot \|\bar{g}\|_2^2 \\
&\leq \max_i \sigma_i(F) \cdot S^2 \ ,
\end{aligned}$$

which implies:

$$\mathbb{E}\left[\frac{\eta^2}{2}\tilde{g}^T F \tilde{g}\right] \leq \frac{\eta^2}{2} S^2 \cdot \left(\bar{\sigma}(F) + z^2 \cdot \mathrm{tr}(F)\right) = \bar{\mathbb{E}} \ . \tag{4.23}$$

Using Equation (4.23), we are now interested in deriving the value of S for which the update stays within the trust region with probability $1 - \beta$. For this, we can use the Markov inequality:

$$\mathbb{P}[\widehat{D_{\mathrm{KL}}} \geq \alpha] \leq \frac{\mathbb{E}[\widehat{D_{\mathrm{KL}}}]}{\alpha} \leq \frac{\bar{\mathbb{E}}}{\alpha}$$

We want the event $[\widehat{D_{\mathrm{KL}}} \geq \alpha]$ to have low probability (less than β). To get $\bar{\mathbb{E}}/\alpha \leq \beta$, we therefore have to set:

$$S \leq \frac{1}{\eta}\sqrt{\frac{2\alpha\beta}{\max_i \sigma_i(F) + z^2 \mathrm{tr}(F)}} \ .$$

Then with probability at least $1 - \beta$, the noisy update stays within the trust region of radius α, *i.e.*, $\widehat{D_{\mathrm{KL}}} \leq \alpha$.

□

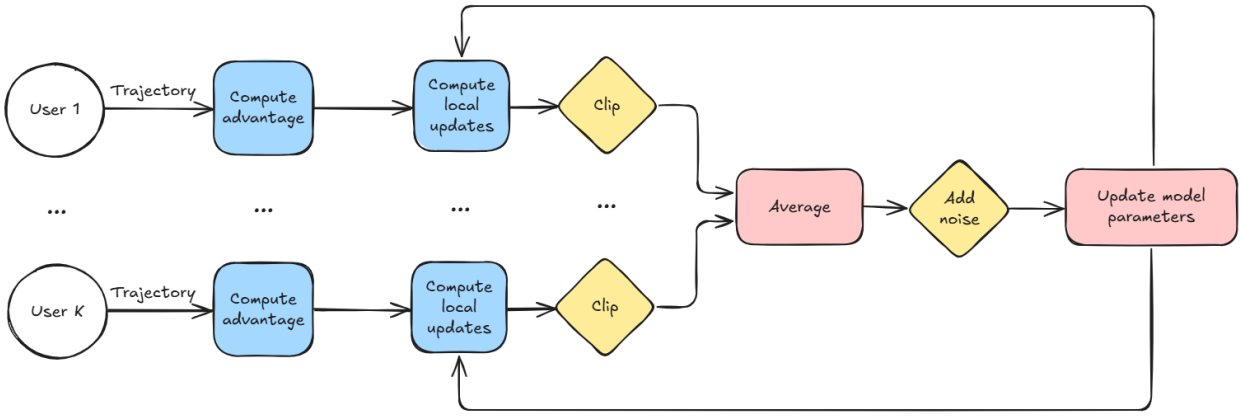


Figure 4.5: Illustration of the trajectory-level differentially private updates used in Algorithm 11.

4.6.3 Practical Algorithm

After studying the theoretical properties of differentially private policy gradient methods in Section 4.6.2, we now introduce a practical approach, described in Algorithm 11.

The core idea behind this algorithm, that is trajectory-level differentially private updates, is illustrated in Figure 4.5. Every K users, the policy is updated with a differentially private gradient step that protects each of the K trajectories. We refer to all the steps between two policy updates as an iteration. All users from iteration i, *i.e.*, users $\mathcal{U}_i = [\![(i-1)\cdot K+1, i\cdot K]\!]$), thus interact with the environment following the same policy π_{θ_i}. At a given iteration i, this global update proceeds in five stages:

1. *Local updates:* K copies of the policies $\left(\theta^{(u)}\right)_{u\in\mathcal{U}_i}$ are updated independently based on each trajectory data τ_u, using the ComputeLocalUpdate subroutine;
2. *Trajectory-level clipping:* Each local update Δ_u is clipped independently such that the clipped update Δ_u has norm $\|\Delta_u\| \leq S$, adequately limiting the influence of each trajectory in the subsequent global update;
3. *Averaging:* The clipped local updates are aggregated in a single global update through averaging, *i.e.*, $\bar{\Delta}_i = \sum_{u\in\mathcal{U}_i} \bar{\Delta}_u/K$;
4. *Noise addition:* The resulting global update is perturbed with Gaussian noise with scale $\sigma = z\mathbf{S}$, *i.e.*, $\tilde{\Delta}_i \leftarrow \bar{\Delta}_i + \mathcal{N}\left(0_d, (\frac{zS}{K})^2 I_d\right)$, ensuring it is differentially private;
5. *Global update:* The policy π_θ is updated with the global DP update, *i.e.*, $\theta_{i+1} = \theta_i + \eta\tilde{\Delta}_i$ and all the data used in this round is discarded.

$\bar{\Delta}_u$ is computed through the auxiliary function ComputeLocalUpdate, which depends on the underlying policy gradient method. Our algorithm is flexible enough that it just requires ComputeLocalUpdate to output an estimate of the ascent direction with norm

Algorithm 11 DP Policy Gradient

Initialize θ, iteration $i = 1$, active users $\mathcal{U}_i \leftarrow \emptyset$
for user $u = 1, 2, ...$ **do**
  $\mathcal{U}_i \leftarrow \mathcal{U}_i \cup \{u\}$
  Run policy π_θ in environment for T timesteps and collect trajectory $\tau_u = ((s_t^u, a_t^u, r_t^u))_{t=1}^T$
  if u % K = 0 **then**
    for $u \in \mathcal{U}_i$ **do**
      Compute advantage estimates $\hat{A}_u = (\hat{A}_t^u)_{t=1}^T$
      Copy policy locally: $\theta^{(u)} \leftarrow \theta$
      Compute clipped update $\bar{\Delta}_u$ from u's data: $\bar{\Delta}_u = \textsc{ComputeLocalUpdate}(\theta^{(u)}, \tau_u, \hat{A}_u, S)$
    end for
    $\bar{\Delta}_i \leftarrow \sum_{u \in \mathcal{U}_i} \bar{\Delta}_u / K$
    $\widetilde{\Delta}_i \leftarrow \bar{\Delta}_i + \mathcal{N}\left(0_d, (\frac{zS}{K})^2 I_d\right)$
    $\theta \leftarrow \theta + \eta \widetilde{\Delta}_i$
    $i \leftarrow i + 1$
    $\mathcal{U}_i \leftarrow \emptyset$
  end if
end for

less than S. Algorithm 12 describes the PPO-style local update procedure that we use in our experiments. It is worth noting that this local update procedure directly yields a clipped update.

Moreover, we treat S as a tunable hyperparameter, hence avoiding the computation of the upper bounds derived in Section 4.6.2, as they either do not take into account the geometry of the policy space or require the estimation of the FIM, which can be difficult in high-dimensional problems and also poses privacy concerns.

The following remarks elaborate on the functioning of the algorithm.

Remark 4.6.5. **Delayed updates.** Using delayed updates allows us to decrease the amount of noise added by a factor K. Algorithm 11 assumes a constant policy update frequency K for simplicity, but this needs not be the case. Moreover, when simulators are available, PG methods often use several parallel environments to speed up training and use larger batches. Using delayed updates in our user-oriented setting is consistent with this feature of PG methods.

Remark 4.6.6. **DP clipping and PPO clipping.** PPO clips the ratio $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$. We found that when clipping the gradients for DP purposes, further clipping $r_t(\theta)$ was not necessary. This supports our theoretical findings suggesting that clipping the gradient was akin to the computation of trust regions, and avoid using too many hyperparameters.

Remark 4.6.7. **Actor-Critic style.** Algorithm 11 is also compatible with the actor-critic framework where we also train a value function V_ψ to compute the advantage estimates. According to our threat model, the adversary observes the policy π_θ, and therefore we only

Algorithm 12 Compute Local Update - PPO

Input: Current policy parameters θ, data τ_u, advantages estimates $\hat{A}_u$, clipping norm S, local epochs E, minibatch size B
Copy current parameters $\theta_0 \leftarrow \theta$.
for $e = 1, ..., E$ **do**
  $\mathcal{B}_u \leftarrow$ (u's data split into size B batches)
  for batch $b \in \mathcal{B}_u$ **do**
    $L(\theta; b) \leftarrow B^{-1} \sum_{(s_t^u, a_t^u, r_t^u) \in b} \frac{\pi_\theta(a_t^u | s_t^u)}{\pi_{\theta_0}(a_t^u | s_t^u)} \hat{A}_t^u$
    $\theta \leftarrow \theta - \eta \nabla L(\theta; b)$
    $\theta \leftarrow \theta_0 + \text{Clip}(\theta - \theta_0; C)$
  end for
end for
Return update $\bar{\Delta}_u = \theta - \theta_0$

need to privatize the policy updates. If however some parameters are shared between π_θ and V_ψ (often the case with visual inputs), it becomes necessary to make the value function updates private as well, and we proceed as for the policy updates.

Remark 4.6.8. **Private adaptive optimization.** To help convergence, we use the Adam optimizer during the local update step (*i.e.*, within the ComputeLocalUpdate function). Adam keeps track of the two first moments of the gradient to adapt the learning rate at each epoch. In the private setting, we must be particularly careful not to pass noise-free gradient information from one user to the next through the optimizer. Indeed, if we would use the non-private gradient of user k to update Adam's estimates of the moments, this information would be used in the local update of another user ℓ at the next iteration. Therefore, we would actually use several times the gradient information of user k, resulting in more privacy leakage due to the sequential composition property of DP. To avoid this, at the start of each iteration i, we substitute the Adam internal statistics with the last private global update $\widetilde{\Delta}_{i-1}$ and its square $\widetilde{\Delta}_{i-1}^2$. Adam is then used as usual during the local update phase, since the data of only one user is concerned.

4.6.3.1 Privacy guarantees.

Appropriately limiting the influence of each trajectory τ_u in the global policy update, the global update procedure described in the previous section is trajectory-level differentially private, as stated in Proposition 4.6.9.

Proposition 4.6.9. *Given $z > 0$ and $\delta \in (0, 1)$, at each iteration $i \geq 0$, the global policy update from Algorithm 11 is (ϵ, δ)-TDP, where $\epsilon := \epsilon(z, \delta)$.*

Proof. The objective is to show that the mechanism $\tilde{f} = f + \mathcal{N}\left(0_d, (\frac{zS}{K})^2 I_d\right)$ with:

$$f : \{\tau_u\}_{u \in \mathcal{U}_i} \longrightarrow \bar{\Delta}^{(i)} = \frac{1}{K} \sum_{u \in \mathcal{U}_i} \bar{\Delta}_u \ ,$$

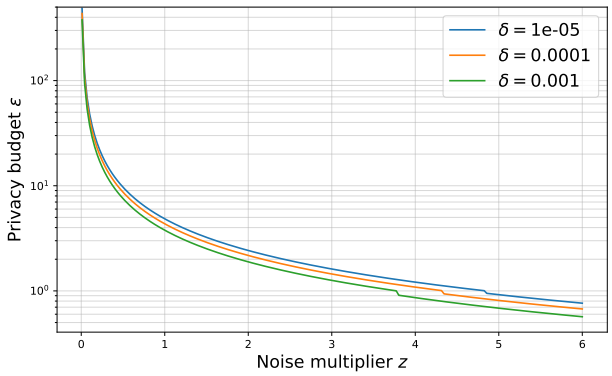


Figure 4.6: Relationship between the noise multiplier z (*x-axis*) and the privacy budget ϵ (*y-axis*) of Algorithm 11.

is differentially private.

First, we show that f has sensitivity $\mathbf{S} = S/K$ with respect to trajectories. Each clipped update $\bar{\Delta}_u$ is clipped with norm S, that is $\|\bar{\Delta}_u\| \leq S$. Moreover, τ_u's data only contributes to Δ_u. Therefore, given $v \in \mathcal{U}_i$:

$$\begin{aligned} f\left(\{\tau_u\}_{u\in\mathcal{U}_i}\right) - f\left(\{\tau_u\}_{u\in\mathcal{U}_i\setminus\{v\}}\right) &= \bar{\Delta}_v/K \\ &\leq S/K \ . \end{aligned}$$

In other words, when we remove or add one trajectory τ_u, $u \in \mathcal{U}_i$, the value of f changes by at most $\mathbf{S} = S/K$. Applying the Gaussian mechanism with noise magnitude $\sigma = z \cdot \mathbf{S}$ to f guarantees that the resulting mechanism $\widetilde{f}$ is (ϵ, δ)-TDP for some $\epsilon := \epsilon(z, \delta)$. □

Since user u's data for $u \in \mathcal{U}_i$ is discarded at the end of round i, such that each policy update operates on disjoint data, the policy released at each iteration adequately protects past users' data, as discussed in Section 4.4.

To compute the privacy budget ϵ from the noise multiplier z — a tunable hyperparameter of the algorithm — and the failure probability δ — a parameter of the problem, quantifying the probability of uncontrolled privacy leakage —, we refer back to the definition of the Gaussian mechanism. While many works in the literature overlook this fact, the standard Gaussian mechanism only works for $\epsilon \in (0, 1)$; meaning that, for small values of z, using it leads to incorrect evaluation of the privacy guarantees. Therefore, we distinguish between two cases depending on the value of z:

- Case $z > C_1(\delta)$: The standard Gaussian mechanism with noise scale $\sigma = zS$ from Chapter 2 achieves (ϵ, δ)-DP with $\epsilon = C_1(\delta)/z < 1$;
- Case $z \leq C_1(\delta)$: The standard Gaussian mechanism does not hold since $C_1(\delta)/z \geq 1$, meaning ϵ would be greater than 1. Fortunately, Zhao et al. [2019] provide Gaussian

mechanisms for the case $\epsilon \geq 1$: according to Theorem 5, given $\delta \in (0, 5)$ and $C_2(\delta) = \sqrt{\log \frac{2}{\sqrt{16\delta+1}-1}}$, the Gaussian mechanism $\mathcal{M}_2$ with scale $\sigma = \frac{(C_2(\delta)+\sqrt{C_2(\delta)^2+\epsilon})\cdot\Delta}{\epsilon\sqrt{2}}$ is (ϵ, δ)-DP. Inverting the relationship, we can derive ϵ given δ and z as $\epsilon = \frac{1+2\sqrt{2}C_2(\delta)z}{2z^2}$.

Numerically, we observe that $\mathcal{M}_2$ yields slightly worse ϵ than $\mathcal{M}_1$ for the same z. For instance, given $\delta = 10^{-5}$, $\mathcal{M}_1$ yields $\epsilon \approx 4.8$ while $\mathcal{M}_2$ yields $\epsilon \approx 5.0$. The latter is the correct value of the privacy budget ϵ. Figure 4.6 plots the relationship between z and the total privacy budget ϵ spent by Algorithm 11 for different δ's. We can see a break in the curves at $\epsilon = 1$, corresponding to the switch between mechanisms $\mathcal{M}_1$ (right portion of the curves) and $\mathcal{M}_2$ (left portion of the curves). We can see that $\mathcal{M}_2$ provides less optimistic privacy budgets (larger ϵ) than $\mathcal{M}_1$ would if we (incorrectly) applied the latter for smaller values of z, although the difference is rather small.

The figure also shows that different δ's have a slight impact on ϵ: a smaller δ protects better against a complete failure of the privacy bound, at the cost of a slightly larger ϵ for a given z. Aligning with common practice, in our experiments we set δ to be less than 10^{-M}, where M is the number of episodes or users.

Remark 4.6.10. Since local updates are computed on local copies of the policy parameters and are never integrated to the global policy, we can leverage several rounds of local updates based on a single user's data within the ComputeLocalUpdate function without incurring more privacy cost — provided that the global update is properly privatized —, on the same model as McMahan et al. [2018]. In practice, we should nevertheless limit the number of local epochs as aggregating policies that are to dissimilar can harm performance.

4.6.4 Experiments

We now evaluate our DP Policy Gradient algorithm from Section 4.6.3, referred to as DPPG, on several benchmarks: Riverswim [Osband et al., 2013], Gym Control's CartPole and Acrobot [Brockman et al., 2016], MuJoCo's HalfCheetah and Hopper [Todorov et al., 2012], as well as a personalized medication dosing simulator for diabetic patients, referred to as Dosing [Salas et al., 2019]. Table 4.11 and Figure 4.10 present asymptotic results and learning curves for these tasks (except for the tabular environment Riverswim, see Secton 4.6.4.1), while Table 4.10 summarizes the hyperparameters. We also apply our approach to an RLHF sentiment tuning task to demonstrate the potential of our approach for the important problem of private LLM-tuning.

4.6.4.1 Riverswim

We first tackle the finite-horizon tabular MDP Riverswim [Osband et al., 2013], a standard environment in DP RL where finding an optimal solution requires a good exploration. The MDP, illustrated in Figure 4.7, has six states and two actions (*swim left* and *swim right*).

Table 4.10: Training and Hyperparameters details. For RIVERSWIM, the learning rate is updated every E_η policy updates with $\eta \leftarrow \eta/k_\eta$, and the clipping norm S is computed from the trust region size α and confidence level β using Prop. 4.6.1 & 4.6.4; for the other tasks, the learning rate is fixed and S is treated as a hyperparameter.

	RIVERSWIM	CONTROL	MUJOCO	DOSING
DISCOUNT FACTOR γ	0.99	0.99	0.99	0.99
LEARNING RATE η	12	7.26×10^{-4}	2.04×10^{-4}	9.25×10^{-4}
LR UPDATE FREQ. E_η	50 EP.	-	-	-
LR UPDATE FACTOR k_η	5.0	-	-	-
MIN. LR	0.06	-	-	-
NUM. LOCAL MINIBATCH'S	1	2	64	2
LOCAL EPOCHS	1	8	8	4
NUM. STEPS T	20	64	2048	64
CLIPPING NORM S	PR. 4.6.1 & 4.6.4	0.05	1.80	0.08
TRUST REGION SIZE α	3.5	-	-	-
CONFIDENCE $1-\beta$	0.6	-	-	-
NUM. FISHER EPISODES	25	-	-	-
FISHER REGULARIZER	1.10^{-3}	-	-	-
ENTROPY COEF.	-	0.36	0.02	0.01
GAE LAMBDA	-	0.85	0.91	0.97
NUM. HIDDEN NEURONS	-	64	64	64
USERS PER UPDATE K	1	8	8	8

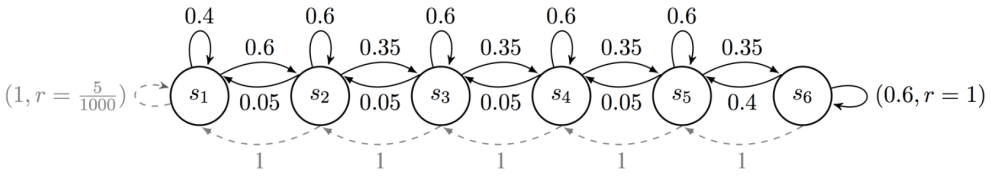


Figure 4.7: The Riverswim environment (from Chowdhury and Zhou [2022]).

Each episode lasts $H = 20$ steps: the agent starts from the left bank of the river, and can either get a small reward $r = 5/1000$ by swimming left toward the bank, or a much larger reward $r = 1$ by swimming up the river, reaching the right bank, and swim right towards the right bank. The environment is stochastic: due to the river current, the swimmer may choose the action *swim right* and end up staying in their current state, or even move left. The transition probabilities influence the difficulty of the task, and we try different probabilities of swimming right and obtaining the rewards at the rightmost state.

On this task, we use a log-linear policy:

$$\pi_\theta(a|s) \propto \exp(\theta^T \phi_{s,a}) \ ,$$

and a linear baseline $V_\psi(s) = \Phi^T \phi_{s,a}$, where $\phi_{s,a}$ is the concatenation of the one-hot encoded

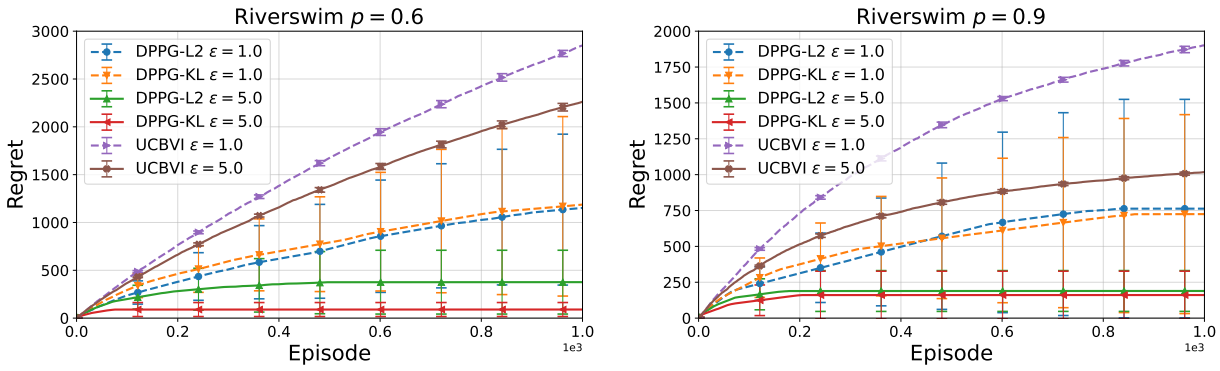


Figure 4.8: Cumulative regret on RiverSwim for $\epsilon = 1.0$ (*dashed line*) and $\epsilon = 5.0$ (*solid line*). DPPG-L2 and DPPG-KL are two versions of our algorithms using L_2 and KL trust regions, respectively. DPPG is compared with UCBVI from [Chowdhury and Zhou, 2022].

state s and the action a. The policy is updated after each episode ($K = 1$) with a single DP policy gradient ascent step (Equation 4.11) Moreover, we consider both DPPG with an L_2 trust region (Proposition 4.6.1), denoted DPPG-L2, and DPPG with a KL trust region (Proposition 4.6.4), denoted DPPG-KL, and use the theoretical values derived in Section 4.6.2 (Propositions 4.6.1 & 4.6.4) to compute the clipping norm S at each update step. For DPPG-KL, S requires the computation of the Fisher information matrix F, so we consider that we can approximate F on public data.

We assess DPPG against UCB-VI [Chowdhury and Zhou, 2022]. Among the several existing DP RL methods, this is the only one with an official implementation, making it the most practical choice for reproducibility and fair comparison. UCB-VI is a private model-based approach for episodic tabular MDPs that leverages private state-action visitation counts to compute a differentially private value function, from which a greedy policy is derived. An exploration bonus is added to the value function, allowing UCB-VI to achieve good performance in an environment like RiverSwim.

Results are presented in Figure 4.8. We consider two different probabilities $p \in \{0.6, 0.9\}$ of obtaining the maximal reward when taking the optimal action at the rightmost state, which influences the difficulty of the task. We also consider two privacy budgets $\epsilon \in \{1.0, 5.0\}$. Despite greater instability, we observe that, on average, both versions of our algorithm far outperform UCB-VI. DPPG-KL also performs slightly better than DPPG-L2, suggesting that considering the policy space geometry is beneficial.

4.6.4.2 Control Tasks

For Gym and MuJoCo control tasks, we use neural network policies and deploy the DPPG algorithm with PPO updates as described in Section 4.6.3. We compare our approach to non-private PPO, expecting a drop in performance as we strengthen the privacy guarantees. To quantify the privacy-utility trade-off for CartPole and Acrobot, we evaluate the

Table 4.11: Asymptotic performance (average episode return over the last 50K steps) of DPPG along with non-private baselines. z is the noise multiplier of the DP Gaussian mechanism. Avg. values and 95% conf. intervals are computed on at least 10 random seeds.

Env.	PPO	A2C	DP Policy Gradient			
CartPole	493.2 ± 9.4	-	$\mathbf{z = 1.0}$	496.4 ± 6.6	$\mathbf{z = 3.0}$	375.5 ± 86.7
Acrobot	-82.1 ± 5.1	-	$\mathbf{z = 1.0}$	-83.0 ± 4.4	$\mathbf{z = 3.0}$	-89.8 ± 7.6
HalfCheetah	2105.5 ± 627.9	1003.3 ± 27.2	$\mathbf{z = 0.05}$	1548.9 ± 399.1	$\mathbf{z = 0.1}$	1328.4 ± 306.5
Hopper	2208.2 ± 477.6	625.0 ± 106.8	$\mathbf{z = 0.05}$	645.0 ± 176.4	$\mathbf{z = 0.1}$	808.9 ± 321.65
Dosing-0	78.9 ± 47.7	-	$\mathbf{z = 0.1}$	109.9 ± 4.8	$\mathbf{z = 0.5}$	17.2 ± 57.6
Dosing-1	99.5 ± 51.9	-	$\mathbf{z = 0.1}$	20.7 ± 53.4	$\mathbf{z = 0.5}$	-24.9 ± 58.8

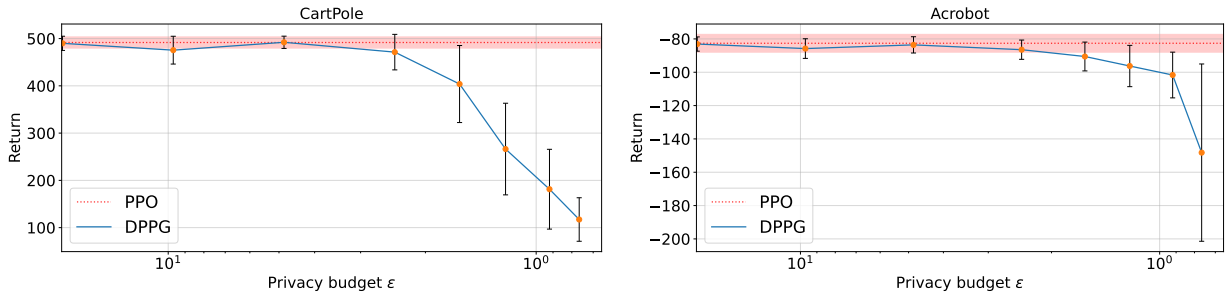


Figure 4.9: Asymptotic performance (average episode return over the last 50K steps) vs. privacy budget ϵ CartPole (*left*) and Acrobot (*right*) in log scale. Average values and 95% confidence intervals are computed on at least 10 seeds.

achieved return for different privacy budgets ϵ (corresponding to various noise multipliers z).

Results are reported in Figure 4.9. We observe that we achieve privacy at almost no cost until ϵ approaches 1. The cost becomes more significant for $\epsilon < 1$, especially on CartPole. Despite offering weak theoretical guarantees, we point out that, given the restrictive worst-case nature of DP, achieving $1 \leq \epsilon \leq 10$ is a realistic and widely accepted practical privacy goal in the DP ML literature (as documented in Ponomareva et al. [2023]). Achieving privacy in the higher-dimensional continuous control tasks Hopper and HalfCheetah proves to be more costly, and we therefore use weaker privacy levels for these environments. Nonetheless, our approach is still able to retain most of the performance of non-private PPO, especially on HalfCheetah. Interestingly, DPPG even outperforms A2C, although the latter offers no privacy guarantee.

Overall, these results show that our approach is able to retain good performance while achieving meaningful privacy levels on standard control tasks with neural policies, which, to the best of our knowledge, has not been reported in the DP RL literature before.

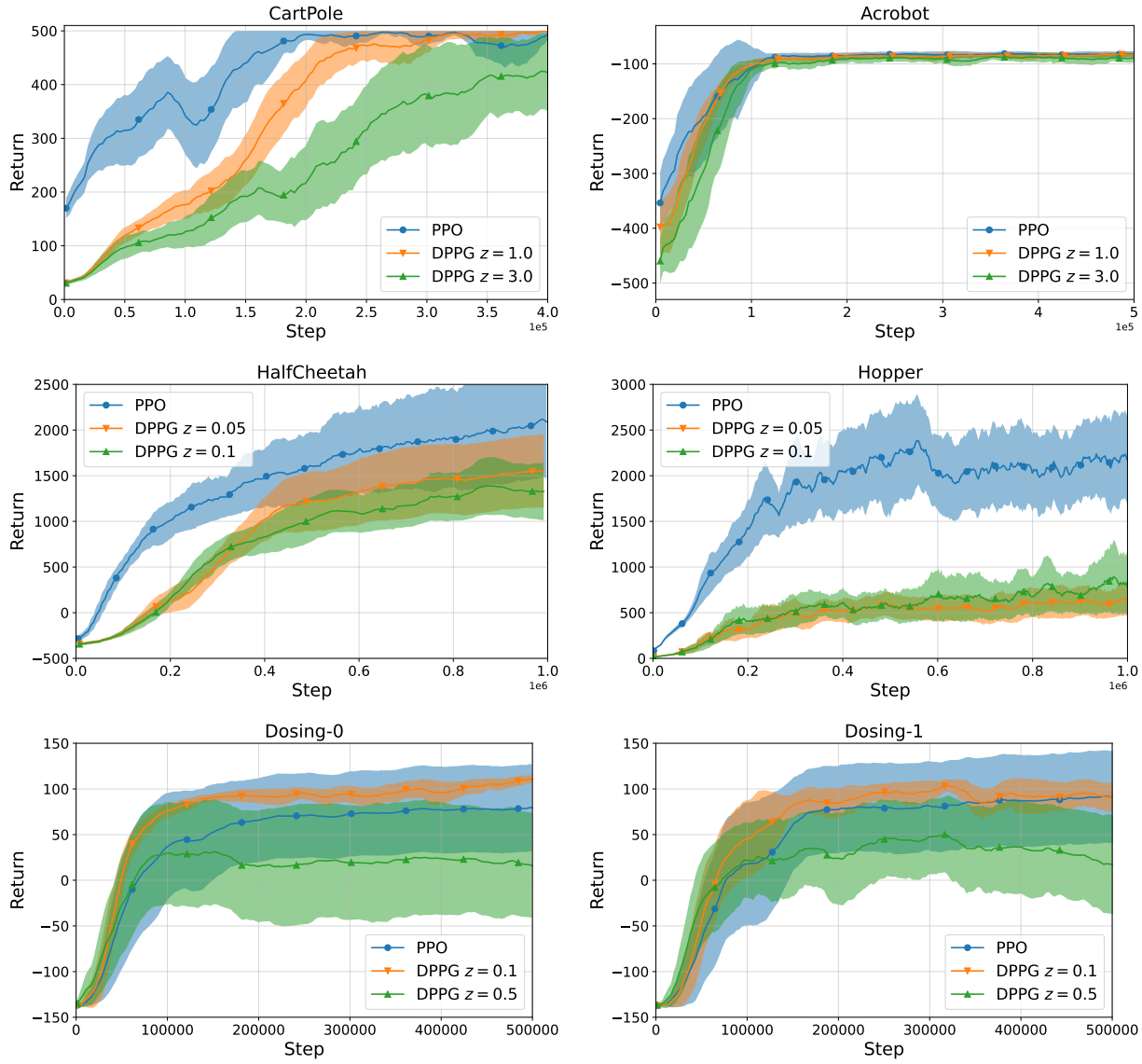


Figure 4.10: Learning curves on Gym Control (*top*), MuJoCo (*middle*) and Dosing (*bottom*) environments. x-axis represents the number of steps in the environment. Average values and 95% confidence intervals are computed on at least 10 random seeds.

4.6.4.3 Medication Dosing Simulations

To better illustrate the real-world relevance of our approach, we learn DP personalized medication policies trained on simulated patient data. In the Dosing task, the goal is to keep blood glucose at tolerable levels in Type-1 diabetic patients, and the RL agent controls insulin uptake. Each trajectory corresponds to a full day of patient measures, and the state contains various health measures such as meal disturbance amounts, insulin levels, and blood glucose levels, which can be considered as sensitive and private information. We use two versions of the environment: Dosing-0, where the meal pattern is constant across episodes, and the more difficult Dosing-1 where each patient (or episode) has a different meal pattern

throughout the day. The simulator is presented in this blog post, and we borrow the official implementation from this GitHub repository.

For this task, we also use a neural network policy with continuous action under two different privacy levels: $z = 0.1$ and $z = 0.5$ corresponding to $\epsilon \approx 38$ and $\epsilon \approx 8$, respectively. While these tasks are more complex than CartPole and Acrobot — with an higher-dimensional state and continuous action space —, results show that our approach achieves these privacy levels with limited impact on performance.

4.6.4.4 Reinforcement Learning from Human Feedback

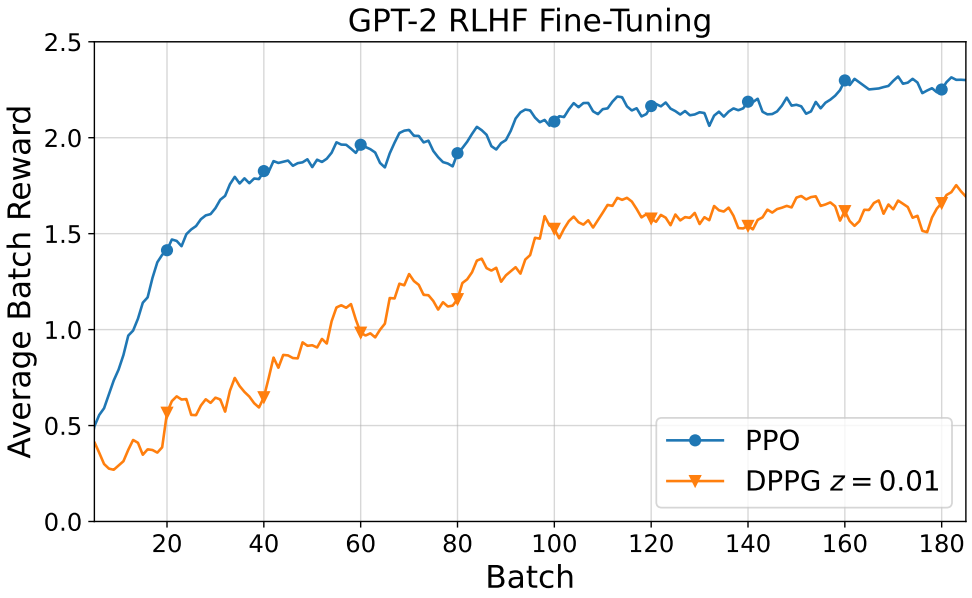


Figure 4.11: Learning curves on the RLHF sentiment tuning task.

Motivated by recent breakthroughs in the field of LLMs where RL played a key role, we also demonstrate the potential of our approach for private LLMs fine-tuning with RLHF. We consider a toy sentiment-tuning task which consists of fine-tuning a GPT-2 model on a dataset containing IMDB reviews. The goal is to encourage the model to produce positive review continuations, using a pre-trained BERT sentiment classifier as a reward model. Precisely, given a prompt s_t (the beginning of a review), the GPT-2 model π_θ completes the review with a sequence $a_t \sim \pi_\theta(\cdot|s_t)$. The reward model R gives a score $r_t = R([s_t, a_t])$ to the resulting review, such that $r_t > 0$ is the review is positive, and $r_t < 0$ otherwise.

A policy gradient algorithm is used to fine-tune the policy parameters θ so that the model achieves larger rewards during training. In the non-private case, we use the classic PPO algorithm for fine-tuning, and use our DPPG algorithm with noise level $z = 10^{-2}$ (that is, $\epsilon \approx 377$) in the private case. We observe in Figure 4.11 that our approach is still able to achieve large rewards despite the DP perturbation. Although this corresponds to weak privacy guarantees, this suggests that our method holds potential for differentially private

fine-tuning of LLMs on sensitive data, for instance, personal data or valuable proprietary data.

4.7 Chapter Summary

While differentially private RL has so far proved challenging, limiting the literature to mainly theoretical work on tabular and linear finite-horizon MDPs, we introduced deep RL algorithms for general γ-discounted MDPs with continuous state and action spaces that protect entire trajectories. In the first part of this chapter, we presented PriMORL, which addresses the offline RL setting with model-based techniques. We empirically show that PriMORL is capable of learning trajectory-level private, neural-based policies in standard control tasks with only limited performance cost. Next, we showed that the introduction of differential privacy in policy gradient methods comes at a limited cost thanks to connections between trust-region approaches and DP-SGD. Both approaches set new empirical standards in differentially private RL, achieving good privacy-utility tradeoffs on a wide range of tasks not yet addressed in the literature, and open encouraging perspectives for the much-needed deployment of private RL agents in the real world. In particular, they are free from the structural limitations of existing methods, applicable to general MDPs, and compatible with neural network approximations, and therefore have the potential to scale to more complex, higher-dimensional problems, such as the most pressing problem of privacy in LLMs.

Overall, this work has highlighted key challenges that need to be addressed if we want to tackle higher-dimensional, real-world applications, and advocates for a paradigm shift in how private RL is approached — from predominantly theoretical research to practical algorithms — setting the stage for future efforts in this direction.

Chapter Five

Promoting Producer-Fairness in Bundle Recommendation

While the previous two chapters focused on privacy, this chapter explores another aspect of trustworthy sequential decision-making: fairness in recommendation. Precisely, we address the problem of bundle recommendation, where recommendations take the form of sets of compatible and complementary items. Benefiting both user experience and platform revenues, this type of recommendation has become very popular in places like e-commerce and media content platforms, thereby attracting attention in research.

However, this setting introduces significant challenges compared to traditional single-item recommendation, hence requiring innovative approaches. In particular, we must rethink how to formalize and introduce fairness in bundle recommendation, a problem that has not yet been addressed in the literature. This chapter attempts to fill this gap by studying fairness of exposure in bundle recommendation, where the exposure of different groups of producers is fairly allocated over a sequence of users.

5.1 Context

Today, an increasing number of marketplaces serve content in the form of bundles, *i.e.*, a composite set of items that are not only **relevant** to the user (or consumer), but also **compatible** with each other (*e.g.*, points of interest within walking distance, compatible accessories for a cell phone), and **complementary** (*e.g.*, movies of different genre, accessories serving different functions). Indeed, bundling can increase the exposure of individual items, trigger purchases through associative links, and eventually increase order sizes, enhancing both user experience and platform revenues [Zhu et al., 2014]. Bundle examples are plentiful and include sets of compatible electronic items, music playlists, books on the same topics, souvenir baskets, and travel packages [Roy et al., 2010, Xie et al., 2012, Amer-Yahia et al., 2014, Benouaret and Lenne, 2016, Chang et al., 2023]. In this work, we are interested in marketplaces where one bundle is served to one user at a time in a sequence [Mehrotra et al.,

2018].

While serving high-quality bundles to users is often the primary goal, ensuring fair visibility of different items is also essential in recommendation systems. Each producer should indeed be afforded an equitable opportunity to capture customer interest and sell their products, and any recommendation algorithm that fails to achieve this objective — such as one that exhibits popularity bias and under-exposes niche items — may unfairly deprive producers of potential revenues. Fairness in single-item top-k recommendations is most commonly defined as fairness of exposure of items in a ranked list [Pitoura et al., 2022, Singh and Joachims, 2018]. In our work, we focus on P(roducer)-fairness in bundles, aiming to achieve fair exposure of items from different producers. While users arrive sequentially and seek the highest-quality bundle, producers expect fair representation of their products over time — resulting in potentially conflicting objectives. In this context, formalizing and enforcing fair exposure of items from different groups is a challenge.

Challenges. Our first challenge **C1** is to devise a problem formulation that 1) combines bundle quality and P-fairness, 2) is expressive enough to cover a wide range of scenarios, and 3) can be effectively tackled. This is not straightforward because we must optimize various dimensions of bundle quality for each user ——relevance, compatibility, and complementarity—— while also accounting for fairness, which is defined over the entire recommendation sequence. Figure 5.1 illustrates the different dimensions of the problem through two examples of movie bundles, highlighting its complexity. In particular, fairness methods designed for single-item recommendation are not suitable: even if a bundle is built from items produced by a fair top-k algorithm, accounting for compatibility and complementarity constraints may lead to choosing items that do not achieve the desired exposure. Our second challenge **C2** is computational. As shown in the literature, building the highest quality bundle is expensive in itself and often NP-hard [Amer-Yahia et al., 2014, Roy et al., 2010], as we must consider pairwise compatibility between items. Adding fairness satisfaction — especially as it is defined across multiple users — further increases its complexity.

Contributions. To address **C1**, we organize items into groups and adapt the notion of demographic parity of exposure [Singh and Joachims, 2018]. To achieve that, we define an exposure vector that reflects the desired exposure of each group. This allows us to formalize the P-Fair Bundle Recommendation Problem which seeks to build the highest quality bundle for each user such that the sequence of recommended bundles satisfies the exposure vector. Producer fairness is thus *amortized* over a sequence of users, similarly to Biega et al. [2018] in the setting of single-item recommendation. This formulation is highly expressive as it covers common cases such as paid inclusion, that is, when a producer pays for a number of listings of their products, or a legal requirement that imposes exposing certain item categories.

To address **C2**, we first propose ILP, an exact Integer Linear Programming solution. This formulation is NP-hard as it generalizes the Quadratic Knapsack Problem [Gallo et al.,

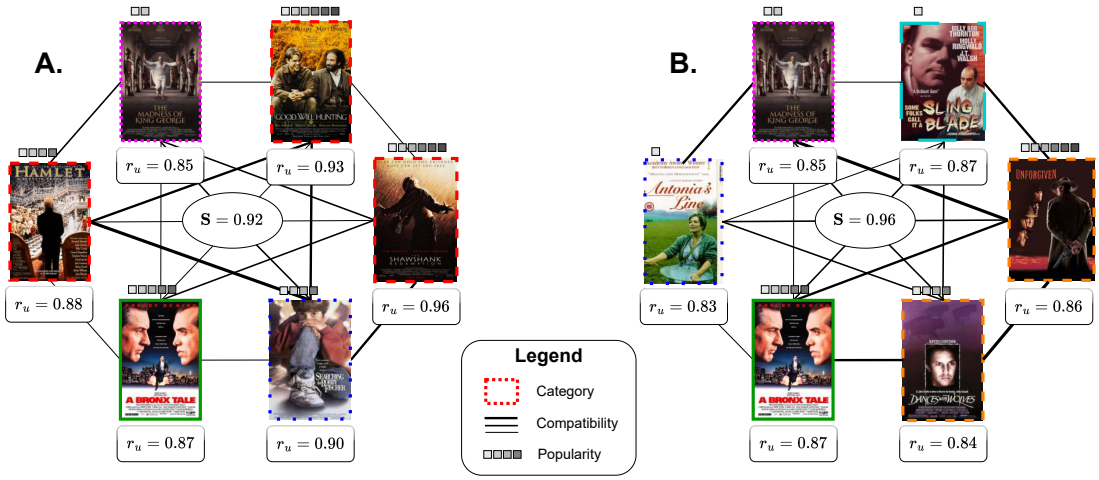


Figure 5.1: Two examples of movie bundles (from *MovieLens*). Here, fairness is defined with respect to movie popularity. The most popular movies may be unfairly over-exposed as a result of the well-known popularity bias of recommendation systems.

1980] with multiple weight constraints and a cardinality constraint. We therefore explore two heuristic-based solutions: F3R, a *fairness-first* approach, and FairWG, a *quality-first* approach. F3R constructs bundles iteratively by first sampling a group of items from the exposure vector distribution, and then greedily choosing an item from this group that best improves bundle quality. FairWG first includes the most relevant item in the bundle and then it uses a linear combination of quality and fairness to complete it. This combination encourages the selection of items from under-exposed groups and reduces the likelihood of choosing items from over-exposed groups. FairWG requires setting a fixed weight to balance bundle quality and fairness. We hence introduce AdaFairWG that dynamically adapts the weight based on how well the current exposures align with the target exposure vector. Given a tolerance level, AdaFairWG chooses to increase or decrease the importance of fairness according to what has been achieved so far in a session.

All three methods are very flexible as they take as input the top-M items (with relevance scores) from any standard, single-item recommendation system. Nonetheless, we regard them as *in-processing* methods, since fairness is directly integrated into the bundle construction process.

We conduct extensive experiments on three datasets: *MovieLens*, *Yelp*, and *Amazon*. Our results investigate the impact of P-fairness on bundle quality and explore response time and scalability. We show that our solutions enhance bundle fairness over fairness-agnostic baselines. As expected, introducing P-fairness adversely impacts quality, revealing the need to achieve a trade-off between constructing bundles and enforcing P-fairness. ILP generally achieves the best quality-fairness trade-off but is only applicable to small instances in practice. We observe that FairWG outperforms F3R and can actually achieve better trade-offs than ILP for comparable response times, while being suitable to large-scale problems. Finally, dynamically adapting the fairness weight in AdaFairWG provides the means to target specific fairness levels over long sessions while achieving high bundle quality.

5.2 Related Work

Bundle Recommendation. A number of works have investigated the recommendation of sets of items, as opposed to traditional single-item recommendation. Two main lines of research emerge from the literature: *pure bundle recommendation* and *bundle generation*. *Pure bundle recommendation* is concerned with the recommendation of existing bundles. Such bundles can be built by the platform themselves, or directly by users (*e.g.*, playlists in music streaming platforms). While earlier methods such as Rendle et al. [2010] only focus on historical user-bundle interactions, treating bundles as atomic units — much like single-item recommendation — modern approaches also exploit past interactions between users and individual items, as well as bundle-item affiliations. This multi-view approach allows the extraction of more powerful patterns, mainly using deep neural networks, ultimately leading to finer recommendations. A popular approach, developed for instance by Chang et al. [2020] and Ma et al. [2024], is to use graph neural networks to effectively propagate information between different views, producing powerful vector embeddings that are then used in a joint optimization process. Other approaches have been proposed, such as Li et al. [2021a] which uses attention networks to capture both intra-bundle and inter-bundle interactions. While such methods achieve impressive results in pure bundle recommendation, they are limited to the recommendation of pre-defined bundles. However, the optimal set of items for a user may not be included in any existing bundles, and in some contexts, pre-defined bundles may not even exist.

On the other hand, *bundle generation* involves the automatic construction of relevant sets of items, possibly from the entire space of available products, allowing for the creation of bundles tailored to users. Some approaches, such as Bai et al. [2019], Hu et al. [2020] and Chang et al. [2023], use deep neural networks to create bundles based purely on user past interactions, including existing bundles or co-purchases. While effective in providing high-quality bundle recommendations, they still rely on the availability of predefined sets of items to extract information from, which can be limiting in contexts where such objects are scarce. Moreover, the black-box nature of these methods complicates the integration of additional structure or constraints to the generated bundle. In particular, introducing meaningful fairness constraints in this context remains a challenge and has yet to be addressed. Closer to our work, Xie et al. [2012], Amer-Yahia et al. [2014] and Benouaret and Lenne [2016] instead consider bundles as sets of relevant and compatible items satisfying some validity constraints such as complementarity and cardinality. Besides not assuming the existence of pre-defined bundles, these formulations offer greater control over the bundle composition, enabling platforms to tailor bundles according to specific criteria. In particular, they provide a suitable framework for incorporating fairness constraints. **Yet, the problem of fair bundle recommendation remains unaddressed even within this framework, and this chapter aims to fill this gap.**

Fairness of Exposure in Recommendation. Achieving fairness has been approached at different stages of a recommendation process: pre-processing [Chen et al., 2023, Fang et al., 2022, Rastegarpanah et al., 2019] such as interventions on input data to achieve a fairness goal [Salimi et al., 2019], in-processing [Burke et al., 2018, Yao and Huang, 2017] such as exploring different weights of a scoring function to achieve a fair ranking [Asudeh et al., 2019], or post-processing where data is re-ranked [Li et al., 2021b, Naghiaei et al., 2022]. The notion of fairness of exposure [Singh and Joachims, 2018], in particular, has received significant attention in the literature. Singh and Joachims [2018] formulates the problem of computing rankings as the optimization of a linear objective over the space of probabilistic rankings, represented by stochastic matrices. By also expressing fairness of exposure as a linear constraint, they apply efficient optimization methods to solve the resulting high-dimensional problem.

Fairness of exposure is often addressed in the context of two-sided platforms, where the goal is to achieve fairness of exposure for the producers, without unfairly impacting the recommendation quality of individual consumers. Biega et al. [2018] map the fair recommendation problem to a fair allocation problem, where the exposure of producer items is fairly allocated to different users, while also guaranteeing a notion of consumer fairness. Usunier et al. [2022] and Do et al. [2021] frame the problem as a concave optimization task and apply the well-known Frank-Wolfe algorithm to efficiently compute rankings that ensure fairness for both producers and consumers. While these works rely on in-processing fairness mechanisms, post-processing approaches as in Wu et al. [2021] and Naghiaei et al. [2022] formulate the fair recommendation problem as a *re-ranking* problem: given a non-fair top-N recommendation, extract a fair ranking of $K < N$ items. Such approaches have the advantage of being applicable to the output of any standard recommendation system.

When users arrive in a sequence, Mehrotra et al. [2018] propose a contextual bandit formulation that integrates producer fairness directly into the reward function. In the context of individual fairness, Biega et al. [2018] formalize the notion of amortized fairness of exposure, where fairness is achieved over a sequence of users.

While our work belongs to in-processing methods and focuses on amortized fairness of exposure, none of the existing proposals applies to high-quality sequential bundle recommendation.

5.3 Preliminaries

We consider a marketplace offering **items** $\mathcal{I}$. Each item $i \in \mathcal{I}$ has a **type** $z \in \{1, ..., Z\}$ (corresponding to values of an attribute or a set of attributes). We set $a_{iz} = 1$ if item i is of type z, and 0 otherwise. Items can belong to more than one type. Moreover, we consider that items come from one of K groups of producers $\mathcal{G} = \{G_1, ..., G_K\}$, and we denote $g_{ik} = \mathbb{1}(i \in G_k)$, as well as $k(i)$ the index of item i's group (*i.e.*, $g_{ik(i)} = 1$).

5.3.1 Bundles

Informally, a bundle $B \subset \mathcal{I}$ is a subset of items that together are relevant to the user, compatible with each other, and valid, i.e., satisfy some application-dependent constraints. We consider any single-item recommender which provides the top-M items $\text{top}(M, u) = (i_{u(1)}, ..., i_{u(M)})$ (with M large enough) as well as their relevance scores $\{r_{ui} | i \in \text{top}(M, u)\}$. We denote $s : \mathcal{I} \times \mathcal{I} \longrightarrow [0, 1]$ a compatibility function over pairs of items. The closer $s_{ij} := s(i, j)$ is to 1, the more compatible i and j are.

Definition 5.3.1. *Bundle Quality.* The quality $\mathbf{Q}(B_u)$ of a bundle B_u served to a user u is a combination of aggregated (user-dependent) relevance $\mathbf{R}(B_u) = f(\{r_{ui} | i \in B_u\})$, and aggregated (user-independent) pairwise item compatibilities $\mathbf{S}(B_u) = g(\{s_{ij} | i, j \in B_u\})$, where f and g are aggregation functions.

Our notion of compatibility covers many practical cases. Consider d a distance $d(i, j) \in [0, 1]$ between two items. If we want bundles of highly similar items, we can use $s_{ij} = 1 - d(i, j)$. In contrast, we can use $s_{ij} = d(i, j)$ if we want bundles of diverse items. s_{ij} can also represent a semantic notion of compatibility such as between a smartphone and a charger.

Definition 5.3.2. *Bundle Validity.* Given $L \in \mathbb{N}$ and $(L_1, ..., L_Z) \in \mathbb{N}^Z$, a bundle B_u is valid if it satisfies:

- **Size constraint:** its size is exactly L: $|B_u| = L$;
- **Complementarity:** it contains at most L_z items of type z: $\sum_{i \in B_u} a_{iz} \leq L_z$.

Adopting bundles of fixed size is a pragmatic choice, as platforms typically present a predetermined number of items simultaneously. The complementarity constraint applies to many scenarios where diversity and minimal redundancy are essential, and also particularly when bundles center on a primary item with complementary accessories — such as a smartphone paired with a compatible case, charger, or headphone.

5.3.2 Exposure and P-Fairness

We consider a recommendation session with a horizon T, denoted $\mathcal{R}_T$, where we serve users $u = 1, ..., T$ sequentially (*i.e.*, one by one). Each user u receives a bundle B_u. We want to achieve fairness of exposure across different groups of producers $G_1, ..., G_K$ over time (*i.e.*, across many users). Our focus here is to formalize this notion of fairness when serving bundles **in a sequence**.

Definition 5.3.3. *Group Exposure and Exposure Vector.* Let $N = \sum_{u=1}^{T} |B_u|$ denote the total number of items recommended during a session $\mathcal{R}_T$, and $N_k = \sum_{u=1}^{T} \sum_{i \in B_u} g_{ik}$ denote the total number of items recommended from group G_k. The exposure $E_k = \frac{N_k}{N}$ of a

group G_k is defined as the proportion of recommended items from G_k among all items recommended during session $\mathcal{R}_T$. An exposure vector $\mathrm{P} = \langle \rho_1, \ldots, \rho_K \rangle$, where $\rho_k \in [0,1]$ such that $\sum_{k=1}^{K} \rho_k = 1$, designates the desired exposure of a set of groups $\mathcal{G}$ where each ρ_k represents the target exposure of group G_k in a recommendation session $\mathcal{R}_T$.

Exposure vectors capture many cases. For instance, a producer could pay for a given number of listings of their products (*i.e.*, *promoted listings*), and ρ_k would therefore represent the share of listings promoted by producers from group k. The targets could also be defined based on a normative idea or a legal requirement about product exposure: for instance, a platform could be willing to promote more items from less popular producers or might be required to expose a certain proportion of domestic products. ρ_k could also simply reflect the number of products offered by G_k.

Definition 5.3.4. *P(Producer)-Fairness.* Given an exposure vector $\mathrm{P} = \langle \rho_1, \ldots, \rho_K \rangle$, a recommendation session $\mathcal{R}_T$ is said to achieve perfect P-fairness iff:

$$\frac{N_k}{\rho_k} = \frac{N_\ell}{\rho_\ell} \iff \frac{E_k}{\rho_k} = \frac{E_\ell}{\rho_\ell} \ , \quad \text{where } k, \ell = 1, ..., K \ .$$

Our P-fairness adapts the standard fairness notion of *demographic parity* with respect to an exposure vector P [Singh and Joachims, 2018]. This is equivalent to setting $E_k = \rho_k$ for all $k = 1, .., K$, which is itself equivalent to $E_k \geq \rho_k$ for all $k = 1, .., K$.

Proof. Indeed, if $E_k = \rho_k$ for every k, then $E_k/\rho_k = 1$ for every k, and therefore $\frac{E_k}{\rho_k} = \frac{E_\ell}{\rho_\ell}$ for any pair (k, ℓ). Conversely, suppose that we have demographic parity. We have:

$$\begin{aligned} \frac{E_k}{\rho_k} = \frac{1 - \sum_{\ell \neq k} E_\ell}{1 - \sum_{\ell \neq k} \rho_\ell} &\iff E_k - E_k \sum_{\ell \neq k} \rho_\ell = \rho_k - \rho_k \sum_{\ell \neq k} E_\ell \\ &\iff E_k = \rho_k + E_k \sum_{\ell \neq k} \rho_\ell - \rho_k \sum_{\ell \neq k} E_\ell \ . \end{aligned}$$

Now we show that $E_k \sum_{\ell \neq k} \rho_\ell - \rho_k \sum_{\ell \neq k} E_\ell = 0$. Therefore:

$$\begin{aligned} E_k \sum_{\ell \neq k} \rho_\ell - \rho_k \sum_{\ell \neq k} E_\ell &= E_k \rho_k \left(\sum_{\ell \neq k} \frac{\rho_\ell}{\rho_k} - \sum_{\ell \neq k} \frac{E_\ell}{E_k} \right) \\ &= E_k \rho_k \left(\sum_{\ell \neq k} \frac{\rho_\ell}{\rho_k} - \frac{E_\ell}{E_k} \right) \ . \end{aligned}$$

But, from demographic parity, $\frac{E_\ell}{E_k} = \frac{\rho_\ell}{\rho_k}$ and therefore $\sum_{\ell \neq k} \frac{\rho_\ell}{\rho_k} - \frac{E_\ell}{E_k} = 0$, which yields the first equivalence.

Now, we need to show that:

$$E_k = \rho_k \ , \ \ k = 1, ..., K \iff E_k \geq \rho_k \ , \ \ k = 1, ..., K \ .$$

We just have to show that $\{E_k \geq \rho_k \ , \ k = 1, ..., K\} \implies \{E_k = \rho_k \ , \ k = 1, ..., K\}$, as the other implication is obvious. We have $E_k = 1 - \sum_{\ell \neq k} E_\ell$. But since $E_\ell \geq \rho_\ell$ for every $\ell \neq k$, then $\sum_{\ell \neq k} E_\ell \geq \sum_{\ell \neq k} \rho_\ell$, and thus $E_k \leq 1 - \sum_{\ell \neq k} \rho_\ell = \rho_k$. Therefore, for any k, we have both $E_k \geq \rho_k$ and $E_k \leq \rho_k$, that is $E_k = \rho_k$.

□

Therefore, a recommendation session is perfectly fair if the realized exposure of any group G_k is equal to its target exposure. However, achieving perfect fairness is unrealistic. We hence introduce a tolerance level ϵ, such that we achieve P-fairness iff:

$$E_k \geq \rho_k(1 - \epsilon) \ , \ k = 1, ..., K \ ,$$

where ϵ could, for instance, be chosen based on legal regulations.

5.4 The P-Fair Bundle Recommendation Problem

5.4.1 General Problem

In many recommendation applications, users arrive sequentially and must be served in real time. Given a time horizon T and a user u, the served bundle B_u must be of highest relevance and compatibility while being valid:

$$\max_{B_u} \mathbf{R}(B_u) \quad \textit{(Relevance)} \tag{5.1}$$

$$\max_{B_u} \mathbf{S}(B_u) \quad \textit{(Compatibility)} \tag{5.2}$$

$$\text{s.t. } B_u \text{ is valid} \quad \textit{(Validity)} \ . \tag{5.3}$$

Moreover, given an exposure vector $\mathrm{P} = (\rho_1, ..., \rho_K)$, we must ensure the following P-fairness constraint over the entire recommendation session $\mathcal{R}_T$:

$$E_k \geq \rho_k(1 - \epsilon) \ , \quad \forall k = 1, ..., K \quad \textit{(P-fairness)} \ . \tag{5.4}$$

While (5.1)-(5.3) only depend on u, (5.4) is defined across all users, and cannot be addressed in its present form since ensuring (5.4) would require prior knowledge of all users in the session and the ability to compute their recommendations simultaneously. Moreover, the conflicting nature of the two objectives (5.1) and (5.2) adds considerable complexity to solving the problem. To address that, we propose a relaxed version of the problem that aligns with real-world circumstances.

5.4.2 A Relaxed Formulation

We propose to combine bundle relevance and compatibility into a scalarized *quality* score, transforming the task into a single-objective optimization problem:

$$\begin{aligned}\mathbf{Q}(B_u) &= (1-\gamma) \times \mathbf{R}(B_u) + \gamma \times \mathbf{S}(B_u) \\ &= (1-\gamma) \times \frac{1}{L} \sum_{i \in B_u} r_{ui} + \gamma \times \frac{2}{L\,(L-1)} \sum_{\substack{i,j \in B_u \\ i<j}} s_{ij},\end{aligned}$$

where $\gamma \in [0,1]$ is used to balance compatibility and relevance.

We also transform the P-fairness constraint (5.4) into an online stepwise conditional constraint. We introduce the following quantities:

$$\begin{aligned}\Delta_k(u) &= \sum_{i \in B_u} g_{ik} - |B_u| \rho_k (1-\epsilon) \ , \\ \Delta_k(1\!:\!u-1) &= \sum_{t=1}^{u-1} \sum_{i \in B_t} g_{ik} - \sum_{t=1}^{u-1} |B_t| \rho_k (1-\epsilon) \ .\end{aligned}$$

$\Delta_k(u)$ and $\Delta_k(1\!:\!u-1)$ are the gaps between the actual and the required number of recommended items from G_k if we wanted to strictly meet the target exposure $\rho_k(1-\epsilon)$, respectively for user u (current recommendation) and for users 1 to $u-1$ (past recommendations). We show that if the constraint:

$$\Delta_k(u) + \Delta_k(1\!:\!u-1) \geq 0 \ , \quad \forall k = 1, ..., K \ , \tag{5.5}$$

is met for every user $u = 1, ..., T$, then the constraint (5.4) holds. Indeed, since $\Delta_k(u) + \Delta_k(1\!:\!u-1) = \Delta_k(1\!:\!u)$, (5.5) is equivalent to $E_k(1\!:\!u) \geq \rho_k(1-\epsilon)$, where $E_k(1\!:\!u)$ is the exposure of G_k up to user u. As it holds for every u, in particular $E_k(1\!:\!T) = E_k \geq \rho_k(1-\epsilon)$.

Proof. **Derivation of the online constraint (5.5) from the global constraint (5.4).** Let $N(u) = |B_u|$ and $N_k(u) = \sum_{i \in B_u} g_{ik}$ denote respectively the total number of items in B_u and number of items from group G_k in B_u, and $E_k(u) = \frac{N_k(u)}{|B_u|}$ be the exposure of group G_k in B_u. Let us further define $N_k(1:u-1) = \sum_{t=1}^{u-1} N_k(t)$, $N(1:u) = \sum_{t=1}^{u-1} N(t)$, and $E_k(1\!:\!u-1) = \frac{N_k(1:u-1)}{N(1:u-1)}$.

We have:

$$\begin{aligned}E_k(1\!:\!u) &= \frac{N(1\!:\!u-1)}{N(1\!:\!u)} E_k(1\!:\!u-1) + \frac{|B_u|}{N(1\!:\!u)} E_k(u) \\ &= \frac{N_k(1\!:\!u-1)}{N(1\!:\!u)} + \frac{N_k(u)}{N(1\!:\!u)} \ .\end{aligned}$$

Thus, if we want the recommendations to comply with the P-fairness constraint at step u,

we want $E_k(1:u) \geq \rho_k(1-\epsilon)$, which yields:

$$\begin{aligned} & N_k(1:u-1) + N_k(u) \geq N(1:u)\rho_k(1-\epsilon) \\ \iff & N_k(1:u-1) + N_k(u) \geq N(1:u-1)\rho_k(1-\epsilon) + |B_u|\rho_k(1-\epsilon) \\ \iff & N_k(u) - |B_u|\rho_k(1-\epsilon) \geq N(1:u-1)\rho_k(1-\epsilon) - N_k(1:u-1) \ . \end{aligned}$$

Denoting $\Delta_k(u) = N_k(u) - |B_u|\rho_k(1-\epsilon)$ and $\Delta_k(1:u-1) = N_k(1:u-1) - N(1:u-1)\rho_k(1-\epsilon)$, we can rewrite the P-fairness constraint as:

$$\Delta_k(u) + \Delta_k(1:u-1) \geq 0.$$

□

We can further interpret (5.5) as follows: the more the group G_k has been over-exposed up to the current round of recommendation (*i.e.*, large $\Delta_k(1:u-1) \geq 0$), the less items from G_k have to be included into u's bundle to meet the P-fairness constraint, relatively to the target exposure $\rho_k(1-\epsilon)$. Therefore, this constraint leaves room to prioritize bundle relevance and compatibility over fairness when possible.

Finally, we replace (5.4) by the conditional constraint (5.5) which can be computed at every round since it only depends on the current user's information and the recommendations of previous users. We thus obtain an online optimization problem:

The P-fair Bundle Recommendation Problem

For every user $u = 1, ..., T$, solve:

$$\begin{aligned} & \max_{B_u} \mathbf{Q}(B_u) \\ \text{s.t. } & B_u \text{ is valid,} \\ & \Delta_k(u) + \Delta_k(1:u-1) \geq 0 \ , \quad \forall k = 1, ..., K \ . \end{aligned} \tag{5.6}$$

Time-dependent tolerance. Since our goal is to achieve P-fairness over the entire recommendation session, it is not needed that $E_k(1:u) \geq \rho_k(1-\epsilon)$ for $u < T$. It may in fact be suboptimal to impose such a condition, as recommendations served earlier in a session $\mathcal{R}_T$ may benefit from more emphasis on bundle quality at the expense of fairness. To allow this while still meeting the P-fairness requirements at horizon T, we propose a time-dependent, monotonically decreasing tolerance $\epsilon(t)$ such that $\epsilon(t) \geq \epsilon(t+1)$ and $\epsilon(T) = \epsilon$. Given $\alpha > 0$:

$$\epsilon(t) = \epsilon \cdot \left(1 + \left(\frac{T-t}{T}\right)^{\alpha}\right) \ .$$

Infinite horizon. Up until now, we have assumed a fixed known horizon T for a recommendation session $\mathcal{R}_T$. This is not limiting in practice: if we want to ensure P-fairness over

an infinite horizon, it is enough to zero out the P-fairness constraints every T users and start over for the next subsequence of T users.

5.5 Exact and Heuristic Solutions

5.5.1 An Integer Linear Programming Formulation

We first propose an Integer Linear Programming (ILP) formulation of the P-fair Bundle Recommendation Problem. To be efficient, it solves the problem over the top-M items provided by a black-box single-item recommender (as mentioned in Section 5.3.1) instead of the entire item space $\mathcal{I}$.

$$
\begin{aligned}
&\max_{X,Y} \frac{1-\gamma}{L} \sum_{i=1}^{M} x_i r_{ui} + \frac{2\gamma}{L(L-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} y_{ij} s_{ij} \\
\text{s.t. } &\sum_{i=1}^{M} x_i = L \\
&\sum_{i=1}^{M} x_i a_{iz} \leq L_z, \quad \forall z = 1, ..., Z \\
&\sum_{i=1}^{M} x_i (g_{ik} - \rho_k(1-\epsilon)) + \Delta_k(1\!:\!u-1) \geq 0 \quad \forall k = 1, ..., K \\
&y_{ij} \leq x_i, \;\; y_{ij} \leq x_j \quad \forall i = 1, ..., M, \quad j = i+1, ..., M \\
&y_{ij} \geq x_i + x_j - 1 \quad \forall i = 1, ..., M, \quad j = i+1, ..., M \\
&x_i, y_{ij} \in \{0, 1\} \;\; .
\end{aligned}
$$

Solving the program above will provide the sequence of highest quality bundles that strictly satisfies all constraints, including the P-fairness requirements; in other words, the optimal sequence of bundles given M. However, this problem is NP-hard: indeed, it can be seen as a generalization of the Quadratic Knapsack Problem [Gallo et al., 1980], which is itself NP-hard, with multiple weight constraints and a cardinality constraint. Therefore, it may not be possible to compute an ILP solution in a reasonable time, especially if M is large, which we assess in our experiments. This may limit the applicability of this approach. For these reasons, we investigate faster and more scalable approaches.

5.5.2 Heuristic Solutions

Our first heuristic method, F3R (Alg. 13) (for Fair Randomized Round Robin), is a *fairness-first* approach. F3R constructs the bundle iteratively by first sampling a group G_k

Algorithm 13 F3R

Input: User u, items $\mathcal{I}$
Output: Bundle B

1: $(I_1^a, ..., I_K^a) \leftarrow (G_1, ..., G_K)$ {Initialize active sets}
2: $B = \{\}$
3: **while** $|B| < L$ **do**
4: $z \sim \mathcal{U}([0, 1])$.
5: **if** $z > \epsilon$ **then**
6: Sample $k \sim (\rho_1, ..., \rho_K)$
7: $I^a \leftarrow I_k^a$
8: **else**
9: $I^a \leftarrow \bigcup_{k=1}^{K} I_k^a$
10: **end if**
11: $i \leftarrow \arg\max_{j \in I^a} \mathbf{Q}(B \cup \{j\})$
12: $B \leftarrow B \cup \{i\}$
13: **for** all z s.t. $a_{iz} = 1$ **do**
14: **if** $\sum_{i \in B} a_{iz} = L_z$ **then**
15: $I_k^a \leftarrow I_k^a \setminus \{j \in I_k^a | a_{jz} = 1\}$ for all $\ell = 1, ...K$
16: **end if**
17: **end for**
18: **end while**
19: **return** B

} CheckComp

from the exposure vector distribution (line 5 to 7), and then greedily choosing the item from G_k that improves the bundle score the most (line 11). Sampling according to the exposure vector ensures that the exposure targets will be met in expectation, hence F3R prioritizes *fairness* over bundle quality. However, we can relax the fairness mechanism by choosing the best overall active item with probability ϵ (lines 8-9). ϵ therefore controls the fairness-quality trade-off. The validity constraints are then ensured, first by stopping the algorithm when the bundle size reaches L (line 3), and second by removing items whose type has been covered from the active sets to meet the complementarity requirements (lines 13 to 17, subroutine referred to as CheckComp).

The second method, FairWG (Alg. 14) (for Fairness-Weighted Greedy), is a *quality-first* approach that extends the BOBO-Pick algorithm from [Amer-Yahia et al., 2014], later taken up by Benouaret and Lenne [2016]. It first includes the most relevant item in the bundle, hence prioritizing relevance (lines 1 to 3). Then, FairWG fills the bundle iteratively similarly to F3R, but augments the bundle quality function with a weighted *fairness term* (line 6). In line with the problem formulation from Section 5.4.2, we define the fairness term as:

$$\mathcal{L}(E_k(1\!:\!u-1), \rho_k) = -\frac{\Delta_k(1\!:\!u-1)}{N(1\!:\!u-1)} = \rho_k - E_k(1\!:\!u-1) \ ,$$

where $N(1\!:\!u-1) = \sum_{t=1}^{u-1} |B_t|$ is the total number of items recommended up to user $u-1$. This term encourages the selection of items from under-exposed groups and reduces the likelihood of choosing items from over-exposed groups.

Algorithm 14 FairWG

Input: User u, items $\mathcal{I}$, fairness weight λ
Output: Bundle B

$w \leftarrow \arg\max_{i \in I} r_{ui}$
$B \leftarrow \{w\}$
$I^a \leftarrow \mathcal{I} \setminus \{w\}$
CheckComp
while $|B| < L$ **do**
  $i \leftarrow \arg\max_{j \in I^a} \Big(Q(B \cup \{j\}) + \lambda \times \mathcal{L}(E_{k(j)}(1\!:\!u-1), \rho_{k(j)})\Big)$
  $B \leftarrow B \cup \{i\}$
  CheckComp
end while
return B

The fairness weight λ controls the trade-off between quality and fairness: the larger λ, the more we emphasize fairness at the detriment of bundle quality. However, it is difficult to anticipate how a specific λ value will balance quality and fairness *a priori* for a given problem, and finding the most suitable value may require several rounds of trial and error. To address this issue and make FairWG more widely applicable, we introduce AdaFairWG that dynamically adapts $\lambda := \lambda_u$ at each step u based on how well the current exposures $E_k(1\!:\!u)$ align with the target exposure vector P. More precisely, for a given tolerance level ϵ, we aim to achieve $\mathbf{F}(B_1, ..., B_T) \geq 1 - \epsilon$ through the following mechanism: if $\mathbf{F}(B_1, ..., B_u) < 1 - \epsilon$, we increase the fairness weight, *i.e.*, $\lambda_{u+1} = \lambda_u \times 2$, encouraging more fairness, and if $\mathbf{F}(B_1, ..., B_u) > 1 - \epsilon/2$, we decrease it, *i.e.*, $\lambda_{u+1} = \lambda_u/2$, to put more emphasis on quality. With AdaFairWG, the recommendation provider thus only needs to specify their chosen tolerance ϵ.

5.6 Experiments

We now empirically evaluate our methods. We examine their response time and scalability, and assess how the introduction of P-fairness affects bundle quality.

5.6.1 Experimental Setup

5.6.1.1 Datasets.

We build semi-synthetic datasets from three real-world datasets with user ratings: *MovieLens-100K* [Harper and Konstan, 2015], *Yelp* [Yelp, 2023], and *Amazon Electronics Retail* [Ni et al., 2019]. For each dataset, we use item features to construct relevant and realistic producer groups, compatibility functions and complementarity attributes, as defined in our bundle recommendation problem. Without loss of generality, we use the SVD matrix factor-

Table 5.1: Description of the datasets.

	MovieLens	Yelp	Amazon
Description	Movies	Restaurants	Cell Phones & Acces.
#Users	943	3,838	64,187
#Items	1,681	2,833	1,290
#Ratings	99,999	153,362	78,017
Compatibility	Periode, Genre	Distance	Customer interactions
Producer Groups	Popularity, $K = 2$	Popularity, $K = 3$	Country, $K = 4$
Complementarity	✗	Type (*e.g.*, Italian)	Type (*e.g.*, Headphones)

Table 5.2: Example of a bundle (*MovieLens*, user $u = 486$) produced by our recommendation algorithm: the Optimal Bundle (*left*) is returned by ILP **without** the fairness constraint and the Fair Bundle (*right*) is returned by ILP **with** the fairness constraint (with $\epsilon = 0$). Items in **bold** reflects differences between the Optimal and Fair solutions.

	Optimal Bundle ($\mathbf{R} = 0.90$, $\mathbf{S} = 0.92$)			Fair Bundle ($\mathbf{R} = 0.88$, $\mathbf{S} = 0.84$)		
	Movie Title	# votes	r_{ui}	Title	# votes	r_{ui}
i_1	Madness of King George	19K	0.85	Madness of King George	19K	0.85
i_2	A Bronx Tale	167K	0.87	A Bronx Tale	167K	0.87
i_3	**Hamlet**	**41K**	**0.88**	**A Time to Kill**	**1K**	**0.83**
i_4	Searching for Bobby F...	43K	0.90	Searching for Bobby F...	43K	0.90
i_5	**Good Will Hunting**	**1,100K**	**0.93**	**Some Folks Call It...**	**2K**	**0.87**
i_6	Shawshank Redemption	2,970K	0.96	Shawshank Redemption	2,970K	0.96

ization algorithm [Salakhutdinov and Mnih, 2007] to infer relevance scores from explicit user ratings. Producer groups are formed such that a fairness-agnostic algorithm would tend to over-recommend (in a normative sense) items from some groups to the detriment of others. In *MovieLens* and *Yelp*, we categorize items into popularity tiers to correct popularity bias and promote balanced recommendations. For *Amazon*, items are split on the producer's country of origin, with a majority from US-based producers; our experiments thus aim to ensure a more balanced representation of items from different origins, for instance, to align with international trade requirements.

Table 5.1 summarizes our datasets along with the features used to construct the different dimensions of our problem, and more detailed information is provided below.

MovieLens. Compatibility: We use a notion of similarity between movies as compatibility. We consider that two movies are similar if they are from the same period and have similar genres. We compute the similarity between two movies i and j w.r.t. pe-

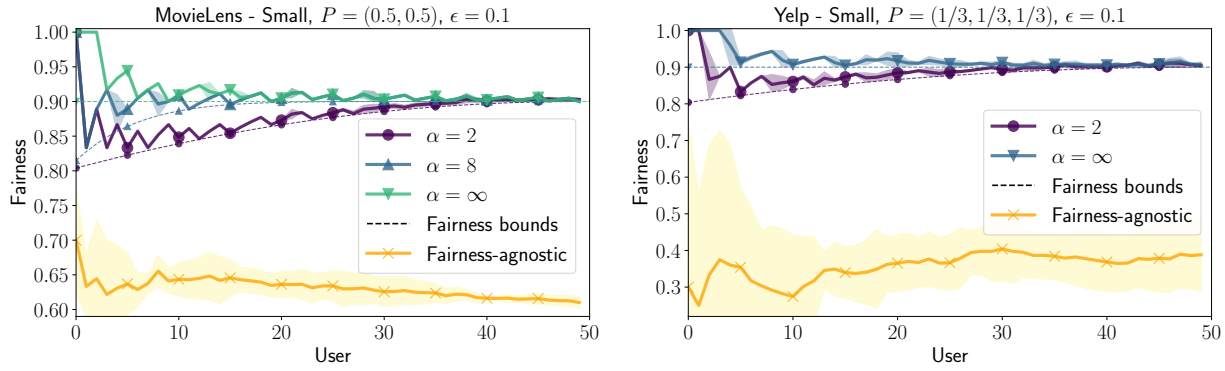


Figure 5.2: Fairness evolution in a session for ILP on *MovieLens* and *Yelp* ($\epsilon = 10\%$). Results are similar on *Amazon*.

riod as $\text{sim}^P(i,j) = \exp(-|\text{year}_i - \text{year}_j|/10)$, where year_i is the year of movie i. As movies have multiple genres, we one-hot encode the genres of each movie into a vector genre_i and compute "genre similarity" as the Jaccard similarity between two genre vectors: $\text{sim}^G(i,j) = \text{Jaccard}(\text{genre}_i, \text{genre}_j)$. Finally, we compute the overall similarity as $\text{sim}(i,j) = 0.5 \times \text{sim}^P(i,j) + 0.5 \times \text{sim}^G(i,j)$. Fairness: We consider fairness of exposure with respect to movie popularity, illustrating the *popularity bias* discussed in Chapter 2. To build the fairness groups G_0, G_1 based on movie popularity, we fetch additional data from the *iMDb* API, in particular the number of users who have rated a movie. Based on preliminary observations, we set a threshold: movies with more than 20,000 votes belong to G_0, and movies with less than 20,000 votes belong to G_1 (the "discriminated" group).

Yelp. Pre-processing: We limit the scope to open businesses (restaurants, bars, etc.) in the city of Philadelphia that have received more than 20 reviews. Moreover, we only keep users who have reviewed more than 20 businesses. Compatibility: We measure the compatibility between two businesses as the distance between their locations. We compute the distance between two locations from their longitude and latitude using the Haversine formula. Denoting $d(i,j)$ the distance in kilometers between two businesses i and j, we compute compatibility as $\exp(-|d(i,j)|/3)$ to obtain a compatibility score between 0 and 1. Fairness: As for *MovieLens*, businesses are split into different popularity tiers G_0, G_1, G_2. We measure popularity based on the number of reviews for a given business: businesses with more than 500 reviews are assigned to G_0, businesses between 100 and 500 reviews are assigned to G_1, and businesses with less than 100 reviews are assigned to G_2. Complementarity: We use the type of businesses from the dataset to get 12 large categories of businesses: *Coffee*, *Bar*, *Brunch*, *Italian*, *Sandwich*, *Meat*, *Seafood*, *Asian*, *TexMex*, *Vegetarian*, *Healthy*, *Sweet*. We use the following complementarity constraints L_z's: $(2,2,1,1,1,1,1,1,1,1,2,1)$ and $(3,3,2,2,2,1,1,2,1,1,3,1)$, for *Small* ($L = 6$) and *Large* ($L = 12$) bundles, respectively.

Amazon. Pre-processing: We limit the scope to items from the category "Cell Phones & Accessories" and from the most common brands (those with more than 30 items) from the

Table 5.3: ILP runtime (per user) and bundle quality for different values of M.

ILP						
	MovieLens		Yelp		Amazon	
M	*Runtime (s)*	Δ Quality	Runtime (s)	Δ Quality	Runtime (s)	Δ Quality
50	0.68 ± 0.29	—	1.03 ± 0.37	—	0.14 ± 0.04	—
100	3.39 ± 2.74	$\nearrow$ 10.53% ± 3.82%	12.85 ± 3.80	$\nearrow$ 3.07% ± 3.40%	0.46 ± 0.11	$\nearrow$ 5.35% ± 3.52%
200	88.61 ± 23.24	$\nearrow$ 14.04% ± 5.12%	184.03 ± 126.99	$\nearrow$ 6.15% ± 2.24%	2.67 ± 0.44	$\nearrow$ 9.44% ± 4.67%
FairWG						
	MovieLens		Yelp		Amazon	
M	*Runtime (s)*	Δ Quality	Runtime (s)	Δ Quality	Runtime (s)	Δ Quality
50		—		—		—
100	$< 10^{-2}$	$\nearrow$ 2.56% ± 4.87%	$< 10^{-2}$	$\nearrow$ 2.36% ± 4.17%	$< 10^{-2}$	$\nearrow$ 0.73% ± 2.71%
200		$\nearrow$ 4.79% ± 5.18%		$\nearrow$ 3.75% ± 4.62%		$\nearrow$ 2.36% ± 4.16%
F3R						
	MovieLens		Yelp		Amazon	
M	*Runtime (s)*	Δ Quality	Runtime (s)	Δ Quality	Runtime (s)	Δ Quality
50		—		—		—
100	$< 10^{-2}$	$\nearrow$ 1.28% ± 1.92%	$< 10^{-2}$	$\nearrow$ 2.95% ± 2.85%	$< 10^{-2}$	$\nearrow$ 0.60% ± 3.68%
200		$\nearrow$ 2.62% ± 2.03%		$\nearrow$ 4.85% ± 3.00%		$\nearrow$ 2.95% ± 4.45%

following countries: USA, China, Japan and South Korea. The majority of items come from these countries. Compatibility: We use past interaction data to measure compatibility between items. Specifically, the compatibility scores $s(i, j)$ reflect whether products i and j have been bought or viewed by the same users. Fairness: We determine the $K = 4$ groups based on their country of origin. G_0, G_1, G_2 and G_3 corresponds to items from Chinese, Japanese, South Korean and American brands, respectively. We have $|G_0| = 176$, $|G_1| = 179$, $|G_2| = 142$, $|G_3| = 793$: there is a majority of items from American brands (61%). Our fairness goal for this dataset is therefore to ensure a more balanced representation of items from different origins (for instance, at least 50% of items from China, Japan, and South Korea). Complementarity: We use the product types given in the dataset, *e.g.*, *Cases*, *Batteries*, *Headphones*, *Speakers*. There are $Z = 62$ different types of items. We use $L_z = 1$ for each product type, such that a bundle is made of items from strictly different types.

5.6.1.2 Methods and Implementation.

We assess our three recommendation methods: ILP, F3R and FairWG. All methods are implemented in Python and experiments are run on Intel i7 CPUs. For ILP, we implement the program from Section 5.5.1 using the Gurobi solver. We also use the ILP method without P-fairness (or, equivalently, with tolerance $\epsilon = 1.0$) as a baseline. Indeed, given the M most relevant items, we can build the highest-quality valid bundle for each user. We then compare this "optimal" bundle with the fair bundle returned by ILP, F3R and FairWG, to evaluate the impact of P-fairness on quality. We also compare them with a random baseline to ensure that our methods indeed provide meaningful recommendations. We further explore the capabilities of our methods by varying the different problem parameters. We consider two bundle sizes, *Small* ($L = 6$) and *Large* ($L = 12$), and, unless stated otherwise, set $\gamma = 1/3$.

Table 5.4: ILP bundle quality on different problem instances, for different values of α and different exposure vectors P. The tolerance is fixed to $\epsilon = 0.1$ (except for *Amazon* for which it is $\epsilon = 0.2$, since no solution is found even with $\alpha = 2$ for $\epsilon = 0.1$).

	MovieLens		
P	$(0.5, 0.5)$	$(0.3, 0.7)$	$(0.2, 0.8)$
$\alpha = \infty$	0.84 ± 0.05	0.80 ± 0.12	0.00 ± 0.00
$\alpha = 2$	0.84 ± 0.11	0.80 ± 0.11	$\mathbf{0.79 \pm 0.12}$
	Yelp		
P	$(1/3, 1/3, 1/3)$	$(0.25, 0.35, 0.4)$	$(0.2, 0.3, 0.5)$
$\alpha = \infty$	0.88 ± 0.19	0.00 ± 0.00	0.00 ± 0.00
$\alpha = 2$	$\mathbf{0.91 \pm 0.13}$	$\mathbf{0.89 \pm 0.16}$	$\mathbf{0.89 \pm 0.15}$
	Amazon		
P	$(0.25, 0.25, 0.25, 0.25)$	$(0.25, 0.25, 0.2, 0.3)$	$(0.2, 0.2, 0.1, 0.5)$
$\alpha = \infty$	0.00 ± 0.00	0.00 ± 0.00	0.72 ± 0.03
$\alpha = 2$	$\mathbf{0.72 \pm 0.03}$	$\mathbf{0.72 \pm 0.03}$	0.72 ± 0.04

We also vary algorithm-specific parameters α (ILP), ϵ (ILP, F3R) and λ (FairWG).

Our results are averages of five recommendation sessions with different random seeds, and we provide confidence intervals of the form $[m - \sigma, m + \sigma]$, where m is the mean and σ the standard deviation. Since F3R introduces additional randomness, we also run five random seeds for each recommendation session for this algorithm.

5.6.1.3 Measures.

- *Quality:* the quality of a bundle (a combination of relevance and compatibility) as given by the bundle scoring function $\mathbf{Q}(B)$. We use the term *Relative Quality* to refer to the ratio $\mathbf{Q}(B)/\mathbf{Q}(B^\star)$, where $B^\star$ denotes the non-fair optimal bundle obtained with the fairness-free ILP program (for the same user and same M).
- *P-Fairness:* Producer-fairness ranges from 0 to 1 and is equal to 1 in case of perfect fairness, *i.e.*, if $E_k = \rho_k$: $\mathbf{F}(B_1, ..., B_T) = 1 - \max_{k=1,..,K} \max(0, (\rho_k - E_k)/\rho_k))$. It is built such that, if we allow a tolerance ϵ, we expect $\mathbf{F}(B_1, ..., B_T) \geq 1 - \epsilon$.
- *Runtime:* measured as response time per user, a key component of user satisfaction in interactive recommendation settings.

5.6.2 Empirical Findings

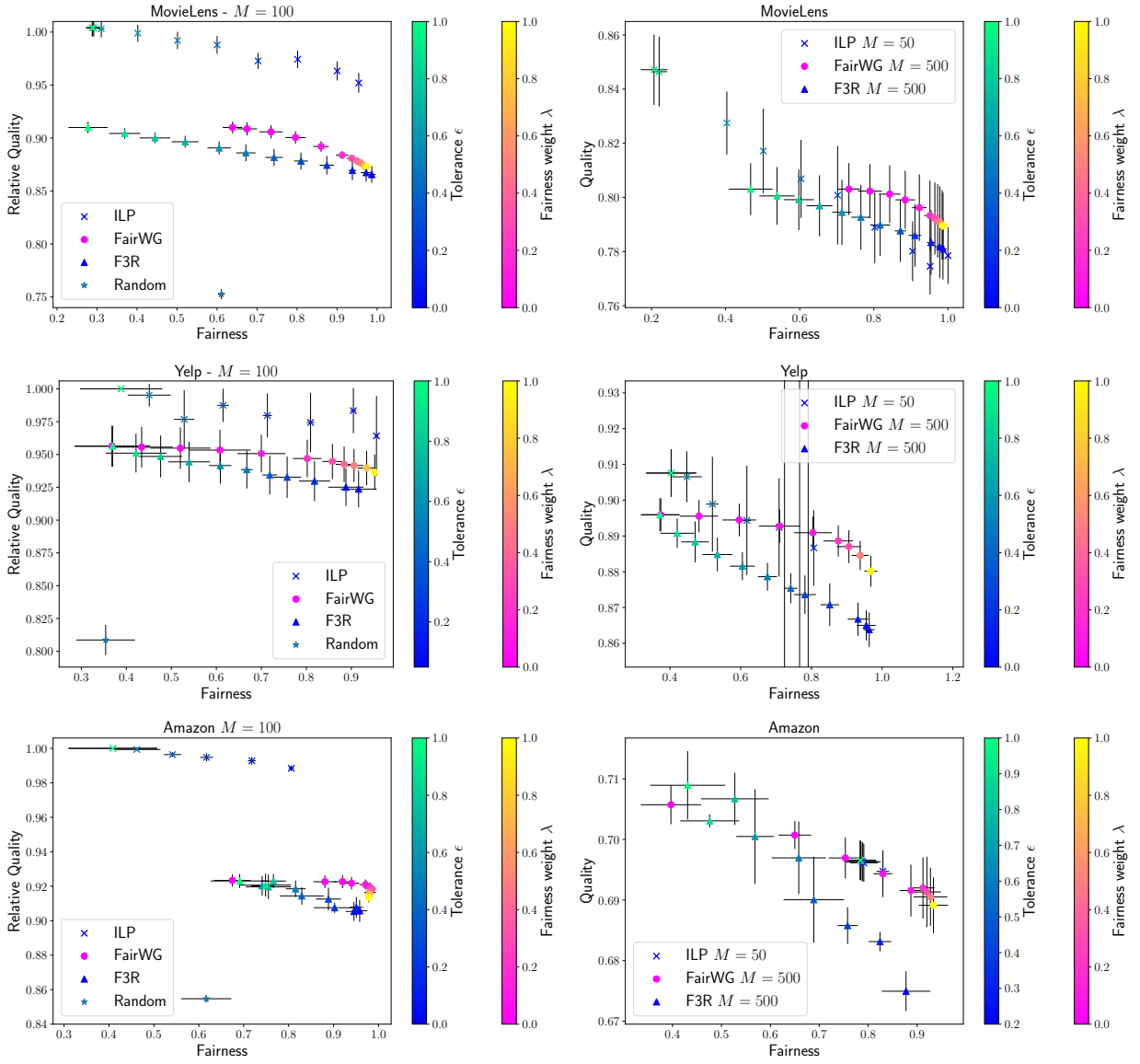


Figure 5.3: Fairness-quality trade-offs for all methods on *MovieLens* (*top*), *Yelp* (*middle*) and *Amazon* (*bottom*). ***Left:*** Trade-offs for a fixed top-M with $M = 100$. Quality relative to the optimal bundle produced by ILP w/o P-fairness (*i.e.*, with $\epsilon = 1.0$). ***Right:*** Trade-offs for comparable runtimes ($\approx 1s$/user for ILP with $M = 50$, $\lessapprox 10^{-2}s$/user for FairWG and F3R with $M = 500$). For ILP, low ϵ's for which solutions cannot be reached are not represented.

Proposed methods enhance recommendation fairness over fairness-agnostic baseline. As illustrated in Figure 5.2, ILP consistently achieves the fairness objective by enforcing the P-fairness constraint at each step. Indeed, **F** (plain lines) never breaches the imposed bound (dashed lines). In contrast, the fairness-agnostic baseline delivers a very unfair solution. Table 5.2 illustrates a *MovieLens* bundle from our ILP solution, with and without P-fairness awareness. To increase exposure for the least popular movie group G_1, the fair solution includes *A Time to Kill* and *Some Folks Call It a Slim Blade*, which are nearly as

relevant as the more popular alternatives (*Hamlet* and *Good Hunting*) selected without the P-fairness constraint. This selection enhances G_1's exposure and meets P-fairness requirements. Although not ensuring strict P-fairness like ILP, our heuristic approaches FairWG and F3R successfully provide fairer recommendations than the baseline (*i.e.*, the ILP method with tolerance $\epsilon = 1.0$), as demonstrated in Figure 5.3 (*left*). Results are consistent across all datasets.

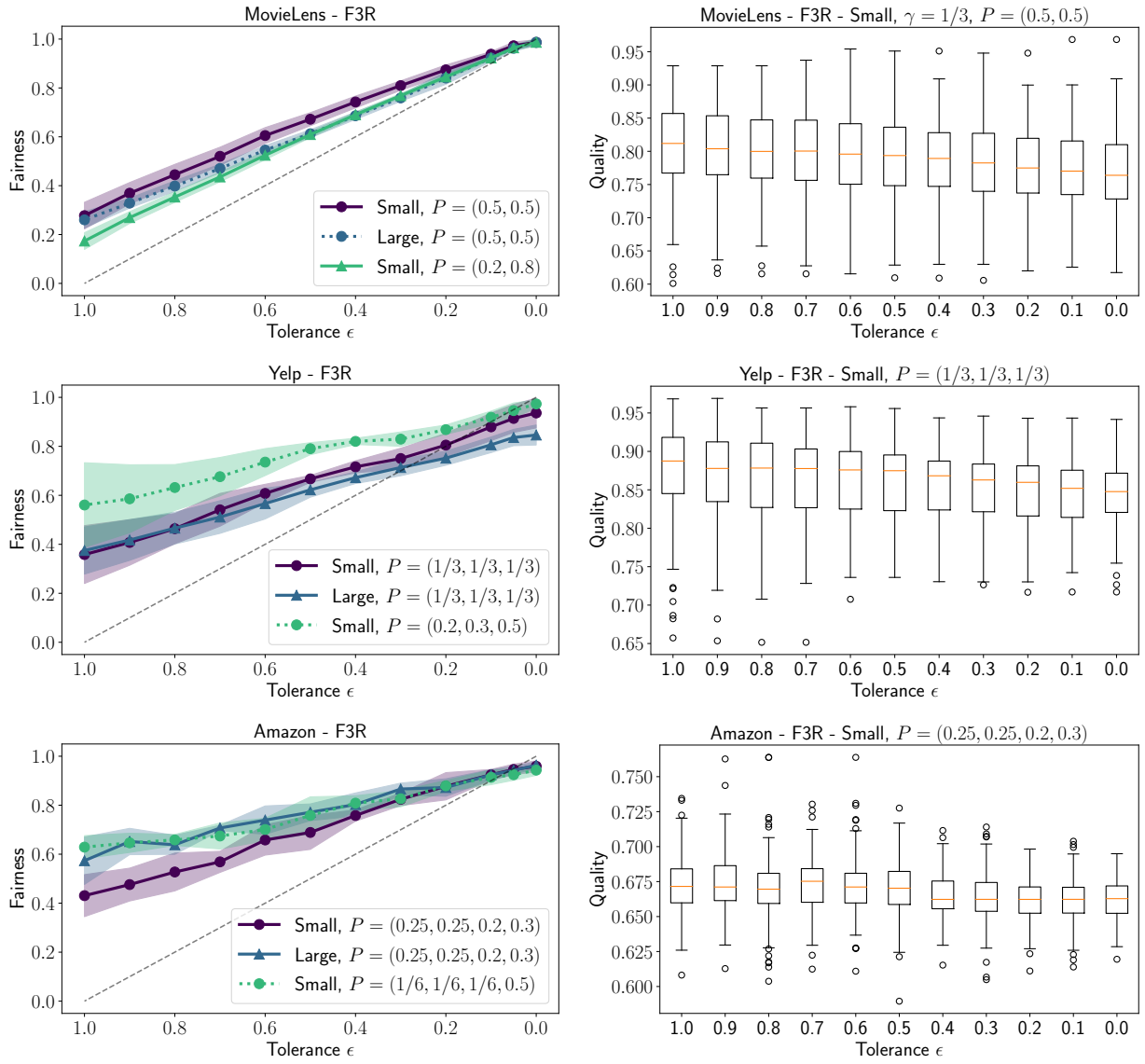


Figure 5.4: Fairness (*left*) and bundle quality (*right*) for F3R for *MovieLens* (*top*), *Yelp* (*middle*) and *Amazon* (*bottom*). The fairness-quality trade-off is achieved with ϵ

Introducing P-fairness adversely impacts quality, revealing a trade-off. Figures 5.3, 5.4 and 5.5 show that the introduction of P-fairness comes at a cost and that one generally cannot obtain a solution that is simultaneously more fair and of higher quality. In fact, for all three methods, increasing the fairness metric leads to a decrease in quality. For ILP and F3R, this quality-fairness trade-off is controlled by the tolerance parameter $\epsilon \in (0, 1)$,

whereas for FairWG, the trade-off is controlled by the weight $\lambda \geq 0$: by increasing λ or decreasing ϵ, we put more emphasis on fairness to the detriment of quality.

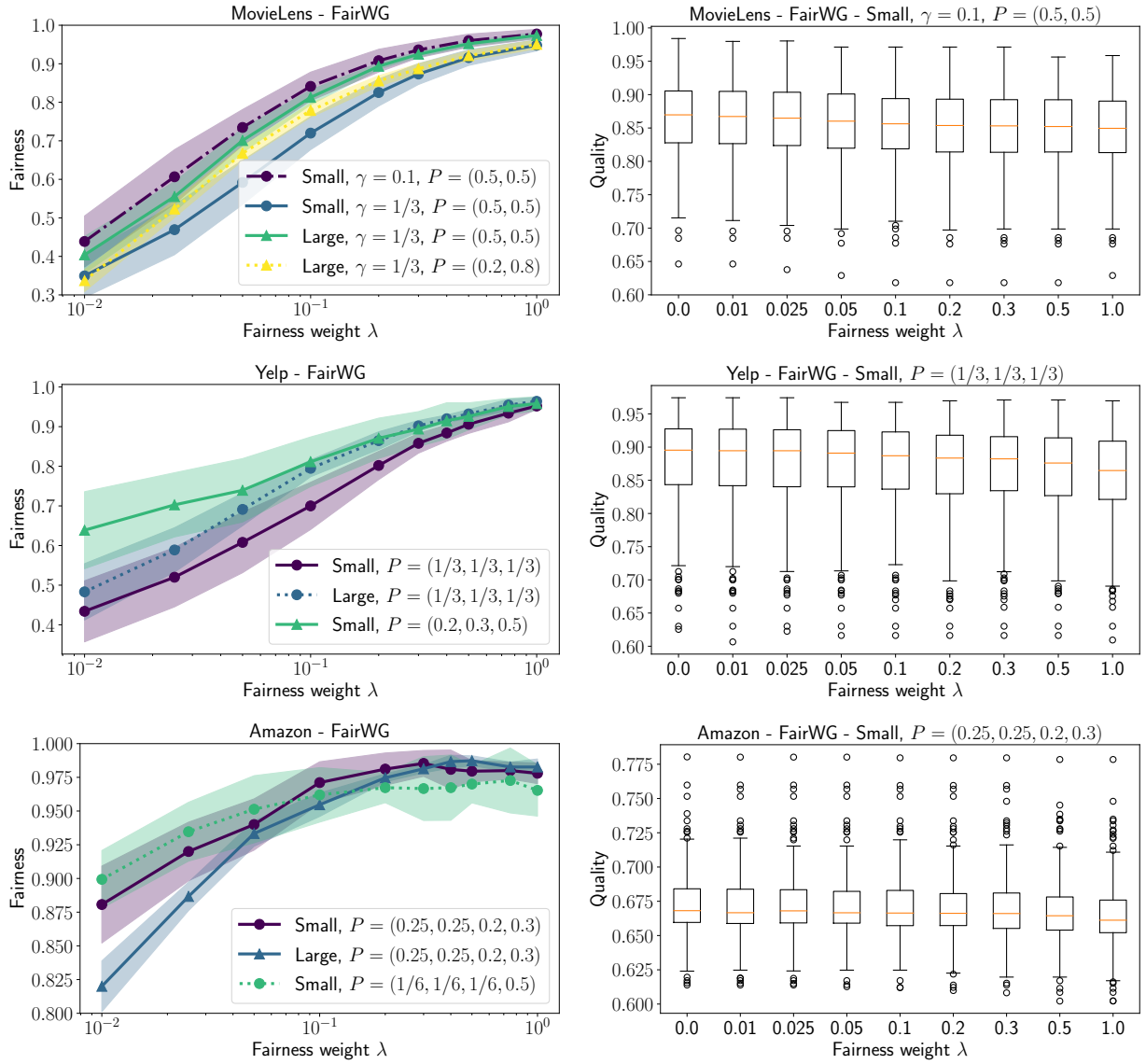


Figure 5.5: Fairness (*left*) and bundle quality (*right*) for FairWG for *MovieLens* (*top*), *Yelp* (*middle*) and *Amazon* (*bottom*). The fairness-quality trade-off is achieved with λ.

ILP generally achieves the best quality-fairness trade-off for small instances. Figure 5.3 (*left*) shows that, for a fixed M, ILP yields superior quality trade-offs compared to the two proposed heuristics. However, this is contingent on the feasibility of the problem. On more complex problems involving complementarity constraints (*Amazon* and *Yelp*), ILP fails to provide solutions for low tolerance values and is therefore unable to reach high levels of P-fairness. This is particularly noticeable on *Amazon* where the program is not solved for $\epsilon < 0.2$. In some of these difficult cases, using adaptive exposure vectors (illustrated in Figure 5.2) can help reach solutions by relaxing the fairness constraint in the earlier stages of the recommendation session. In Table 5.4, we indeed observe that the ILP program fails

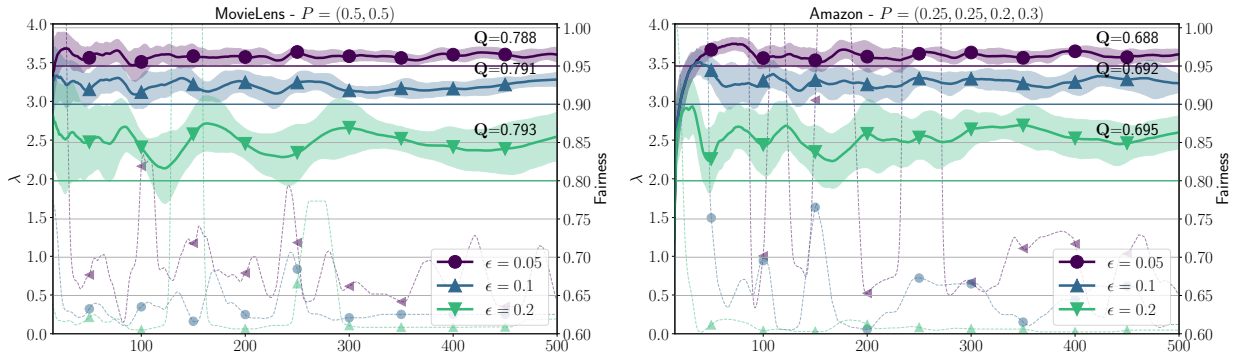


Figure 5.6: Results for AdaFairWG with tolerance levels $\epsilon \in \{5\%, 10\%, 20\%\}$ and horizon $T = 500$, on *MovieLens* (*left*), and *Amazon* (*right*). Plain lines: fairness **F**. Dashed lines: adaptive fairness weight λ. Beside each curve, the average quality **Q** is displayed. Results are similar for *Yelp*.

in several problem instances with $\alpha = \infty$ (*i.e.*, fixed exposure vector) but can recover high-quality solutions with $\alpha = 2$ (*i.e.*, adaptive exposure vector). Moreover, ILP is by far the slowest approach (as shown in Table 5.3), and despite the increase in quality associated with larger M, it has to remain small if we consider deploying it in real-time settings.

Heuristics achieve competitive trade-offs at minimal cost compared to ILP. While slightly inferior to ILP for a given M, F3R and FairWG still provide high quality bundles while encouraging fairness (Figure 5.3). As opposed to ILP, these heuristics also enjoy a near-constant runtime with respect to M (Table 5.3), while increasing M can lead to significant improvements on quality. Therefore, if we contrast all methods for comparable runtimes instead of a fixed M, as done in Figure 5.3 (*right*), we observe that they are actually very close in performance. In fact, FairWG can even achieve better quality-fairness trade-offs than ILP at high fairness levels.

While F3R can target specific fairness levels, it is generally outperformed by FairWG. Meanwhile, AdaFairWG successfully addresses the limitations of FairWG. Given a tolerance level ϵ, F3R generally achieves the fairness goal $\mathbf{F}(B_1, ..., B_T) \geq 1 - \epsilon$ as expected (see Figure 5.4). However, it might not meet this objective for ϵ close to zero, especially in the more difficult *Yelp* and *Amazon* problems. Indeed, when fairness is hard to achieve, there may not be enough available items in each group G_k to keep sampling according to the exposure vector distribution until the bundle is filled, resulting in a mismatch between the achieved and the expected fairness level. This can be alleviated by increasing M, expanding the range of available items, as can be seen in the *Amazon* experiments. On the other hand, FairWG generally performs better, delivering higher quality recommendations across all fairness levels (see Figures 5.3), but requires choosing λ through trial and error. As shown in Figure 5.6, AdaFairWG, which dynamically adjusts λ (in *dashed lines*) during the session, overcomes this difficulty and can successfully achieve a given target tolerance

level ϵ. Using a large horizon $T = 500$, we observe that the fairness level ends up stabilizing just over the target level. This suggests that this method is highly effective in controlling the fairness level over large horizons.

5.7 Chapter Summary

In this work, we investigated P(roducer)-fairness in the context of sequential bundle recommendation. To the best of our knowledge, our work is the first to combine quality and fairness in bundle recommendation. The need to account for multiple dimensions, pairwise item interactions, and P-fairness across the entire recommendation sequence makes this problem challenging from both formalization and computational perspectives. We formulate the P-fair bundle recommendation problem in an expressive and feasible manner and propose both exact and heuristic solutions. Empirical investigations show that our methods successfully enhance recommendation fairness in bundles and highlight the strengths and limitations of each approach. In particular, we show that our heuristic algorithms can be as effective as the expensive ILP solution, especially when dynamically balancing fairness and quality during the recommendation session.

Chapter Six

Conclusion

6.1 Research Summary

As automatic decision-making increasingly impacts society, ethical concerns — particularly fairness and privacy — have been growing in parallel. Addressing these issues is both a societal imperative and a strategic matter for technology organizations, but reconciling ethics and performance brings many challenges. This thesis addresses the problem of introducing privacy and fairness into decision-making without sacrificing performance, hence contributing to the ongoing efforts towards the deployment of trustworthy solutions in real-world settings. We approached this effort from the perspective of three sequential decision-making problems with significant practical implications: best arm identification in multi-agent bandits, deep reinforcement learning, and bundle recommendation. After identifying and formalizing the privacy and fairness challenges in these tasks, we propose solutions that efficiently balance ethics and performance, taking into account the practical constraints of each setting.

In the following, we summarize the key contributions from each chapter.

Chapter 3 addressed best arm identification in multi-armed bandits, a framework in which important problems, such as the optimization of telecommunication networks and clinical trials, can be formalized. We identified that the necessary communications between the agents could leak private information, putting both the input data and the output best arm at risk. In response, we developed two distinct privacy-preserving solutions: the first algorithm, DP-MASE, protects the input data using differentially private communications, while the second method, Corrupted Elimination, protects the best arm identity by combining suboptimal local agents, message corruption and a well-calibrated majority voting rule. We demonstrated the effectiveness of such approaches on toy examples, successfully balancing between sample complexity and formal privacy guarantees.

Chapter 4 tackled the important yet relatively overlooked problem of privacy-preserving deep reinforcement learning, which holds significant implications on timely topics such as LLM fine-tuning. After identifying the general requirements of privacy-preserving RL, par-

ticularly the need for trajectory-level privacy guarantees, our contribution to this problem is twofold. The first part of this chapter addressed the offline setting, which is relevant for real-world problems where extensive interaction with the system is not feasible and reinforcement learning is applied to a fixed private dataset. There, we developed PriMORL, a deep model-based approach that leverages trajectory-level updates to learn a differentially private model of the environment. The second part of the chapter investigated the introduction of privacy in policy gradient methods — which achieve state-of-the-art results in many benchmarks and are key components of recent advances in LLMs. Leveraging connections between trust-region approaches and gradient clipping, we showed that privacy could be introduced at a limited cost in policy gradient approaches, and proposed a simple practical algorithm based on PPO. Both methods demonstrated promising privacy-performance trade-offs in previously unexplored deep RL benchmarks, laying the foundation for deploying privacy-preserving RL agents in complex tasks.

Chapter 5 considered the problem of recommending bundles of items, *i.e.*, unordered lists of compatible and complementary items, with applications ranging from e-commerce to media content platforms. In this setting, we formalized the problem of maximizing bundle quality while guaranteeing producer-fairness of exposure over a sequence of recommendations as a high-dimensional constrained optimization process. To tackle this challenging, NP-hard problem combining pairwise item interactions and long-term user recommendation dependencies, we took an approach that lies between in-processing and post-processing methods, building bundles from the output of a generic single-item recommendation system, but integrating fairness directly into the bundle construction process. We provided an exact Integer Linear Programming (ILP) solution as well as two heuristic-based solutions. We showed on three datasets, based on real-world data, that all our methods adequately balance producer fairness and recommendation quality. The ILP approach can reach optimal solutions in reasonable response time for small instances, while heuristic-based solutions achieve fast and high-quality recommendations in large problems.

6.2 Perspectives

This thesis has highlighted important challenges in multi-agent bandits, privacy-preserving deep reinforcement learning and fair bundle recommendation, hence offering valuable insights and opportunities for further research in these fields. This section outlines several promising avenues for each domain.

Multi-Agent Bandits with Private Communications.

Extension to other settings. Given the potential of multi-agent bandits to optimize large-scale real-world distributed systems — *e.g.*, telecommunication networks, energy grids, advertising, clinical trials — and associated privacy threats, extending the approaches de-

veloped in Chapter 3 to settings beyond fixed-confidence best arm identification with homogeneous arms appears to be a valuable direction for future research. Examples of such settings include multi-agent best arm identification with fixed budget [Karpov and Zhang, 2023], collaborative regret minimization [Karpov and Zhang, 2024], multi-agent linear bandits [Moradipari et al., 2022], and multi-agent bandits with heterogeneous arms [Chawla et al., 2023]. Each of these settings has specific characteristics that prevent straightforward applications of our methods and may require different communication schemes. In particular, exploiting the linear structure of linear bandits in the private setting is nontrivial; and the voting schemes from Chapter 3 may be insufficient for effective collaborative regret minimization, potentially requiring the privatization of higher-dimensional messages (see, *e.g.*, the CFTRL and DFTRL algorithms from Yi [2023]).

New communication strategies. Since the efficiency of multi-agent bandit methods heavily depends on how agents share information, exploring different private communication strategies is another promising avenue. In Chapter 3, communications were restricted to binary messages, indicating whether a given arm had been eliminated locally or not. If such messages are well-suited for bandwidth-constrained systems, we believe that allowing the agents to send more informative messages, like continuous signals, might be more efficient. Intuitively, if messages are more informative, the multi-agent collaboration process gets more efficient, but this is at the expense of privacy. This creates a spectrum ranging from binary messages — which provide minimal information and require less perturbation to achieve privacy — to messages sharing raw rewards — which maximize information but demand more noise to maintain privacy. In private settings, there might be an optimal trade-off between these extremes, where messages are sufficiently informative without requiring excessive noise. Such a trade-off may be worth investigating, for instance from an information-theoretic perspective.

Privacy-Preserving Deep Reinforcement Learning.

While Chapter 4 laid solid foundations for privacy-preserving reinforcement learning in complex settings, several challenges remain before it can be applied in real-world scenarios. We identify key research directions to bridge this gap.

Scale to higher-dimensional tasks. Experiments from Chapter 4 showed that, despite promising results on challenging deep reinforcement learning benchmarks, our methods struggled in higher-dimensional tasks. Future research should therefore aim at alleviating the negative impact of problem dimensionality. To this end, we believe a promising direction is privately learning compact representations of high-dimensional inputs and performing planning directly in the latent space, as explored by Jiang et al. [2023] in the non-private setting. Another avenue to enhance the scalability of our methods is to implement differential privacy at the level of trajectory segments. As argued in the recent position paper by Fan et al.

[2025], this approach may offer improved privacy-utility trade-offs compared to protecting entire trajectories, while still addressing the primary privacy concerns in reinforcement learning.

Empirical privacy auditing. Even if reinforcement learning algorithms have been shown to be vulnerable to membership inference attacks [Gomrokchi et al., 2023], these attacks remain underexplored and often rely on unrealistic assumptions about the adversary. In practice, privacy attacks in reinforcement learning are likely very challenging, particularly due to the noisy dynamics of the environment, the stochastic nature of training algorithms, and the high dimensionality of the tasks. Therefore, achieving theoretically-sound privacy budget ϵ in complex RL tasks is probably an unrealistic goal, and the field should aim for more pragmatic targets — for instance, ϵ in the 1 to 10 range, or even larger. However, as pointed out in Ponomareva et al. [2023], such privacy guarantees must be backed by empirical privacy audits to ensure they provide sufficient protection. Therefore, an important research avenue is the development of practical attacks based on realistic adversary assumptions, as well as rigorous and standardized privacy-auditing benchmarks for a strong empirical evaluation of privacy-preserving RL algorithms.

Off-policy methods. After the offline and on-policy settings studied in this thesis, a natural research direction is to develop comparable off-policy methods for privacy-preserving deep online reinforcement learning. Since off-policy methods often reuse past data from replay buffers, they introduce additional privacy challenges. Nevertheless, studying this setting is valuable due to its broad applicability and its advantages over on-policy approaches, such as better sample efficiency. In particular, off-policy methods may also be used for aligning LLMs [Flet-Berliac et al., 2024].

Fair Bundle Recommendation.

With bundle-based recommendation gaining popularity as a strategy that benefits both producers and consumers, the fair bundle recommendation problem studied in Chapter 5 also opens a number of worthwhile avenues in research.

Fairness in multi-bundle recommendation. Chapter 5 considered recommending one bundle at a time, which aligns well with many real-world recommendation scenarios. In contrast, Amer-Yahia et al. [2014] studied the recommendation of sets of diverse and complementary bundles, which find applications in information retrieval. Moreover, although more complex to define than in the single-item setting — particularly due to the potential for overlap between bundles — the problem of ranking bundles has also attracted attention [Roy et al., 2010]. These settings may require other notions of exposure, such as accounting

for position bias, as in Singh and Joachims [2018]. Formalizing and addressing fairness in multi-bundle recommendation thus represents a promising direction for future research.

Address consumer fairness. Chapter 5 focused on producer fairness, without delving into how producer fairness may impact consumers individually, *i.e.*, *consumer fairness*. However, it is well-established that promoting producer fairness can inadvertently lead to consumer unfairness, potentially reducing user satisfaction and engagement. The literature on single-item recommendation systems has long explored the balance between consumer and producer fairness — often referred to as CP-fairness — in two-sided marketplaces, for instance in [Naghiaei et al., 2022, Mehrotra et al., 2018, Patro et al., 2020]. Another avenue for future work is thus to formalize and account for consumer fairness in bundle recommendation.

Fairness in black-box methods. Finally, it seems important to investigate how fairness could be integrated into deep learning-based bundle recommendation and generation methods such as [Chang et al., 2023, Ma et al., 2024]. While these black-box approaches inherently lack interpretability and transparency, making it challenging to introduce and control fairness in-process, they may be better suited than the methods in Chapter 5 for large-scale, multi-modal settings involving complex interactions between users, items, and bundles. Studying fairness in such approaches is therefore essential.

References

Martín Abadi, Andy Chu, Ian J. Goodfellow, Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of SIGSAC*, 2016.

Mohammad Mehdi Afsar, Trafford Crump, and Behrouz H. Far. Reinforcement learning based recommender systems: A survey. *ACM Comput. Surv.*, 55(7):145:1–145:38, 2023.

Mridul Agarwal, Vaneet Aggarwal, and Kamyar Azizzadenesheli. Multi-Agent Multi-Armed Bandits with Limited Communication. *JMLR*, 23:1–24, 2021.

Sihem Amer-Yahia, Francesco Bonchi, Carlos Castillo, Esteban Feuerstein, Isabel Méndez-Díaz, and Paula Zabala. Composite retrieval of diverse and complementary bundles. *IEEE Trans. Knowl. Data Eng.*, 26(11):2662–2675, 2014.

Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. In *Proceedings of ICLR*, 2021.

Abolfazl Asudeh, H. V. Jagadish, Julia Stoyanovich, and Gautam Das. Designing fair ranking schemes. In *Proceedings of SIGMOD*, 2019.

Ali Ateeq and Marwan Milhem. *Integrating Ethical Principles in Corporate Strategy: A Comprehensive Analysis of Business Ethics in Contemporary Enterprises*, pages 201–210. 2024.

Jean-Yves Audibert and Sébastien Bubeck. Best Arm Identification in Multi-Armed Bandits. In *Proceedings of COLT*, 2010.

Jinze Bai, Chang Zhou, Junshuai Song, Xiaoru Qu, Weiting An, Zhao Li, and Jun Gao. Personalized bundle list recommendation. In *Proceedings of WWW*, 2019.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *Proceedings of NeurIPS*, 2018.

Solon Barocas and Andrew D. Selbst. Big data's disparate impact. *California Law Review*, 104:671, 2016.

Raef Bassily, Adam D. Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the Annual Symposium on Foundations of Computer Science*. IEEE, 2014.

Debabrota Basu, Christos Dimitrakakis, and Aristide C. Y. Tossou. Differential privacy for multi-armed bandits: What is it and what is its cost? *CoRR*, abs/1905.12298, 2019.

Alejandro Bellogín, Pablo Castells, and Iván Cantador. Statistical biases in information retrieval metrics for recommender systems. *Inf. Retr. J.*, 20(6):606–634, 2017.

Idir Benouaret and Dominique Lenne. A package recommendation framework for trip planning activities. In *Proceedings of RecSys*, 2016.

Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. In *Proceedings of SIGIR*, 2018.

Arpita Biswas, Gourab K. Patro, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. Towards fair recommendation in two-sided platforms. *CoRR*, abs/2201.01180, 2022.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *CoRR*, abs/1606.01540, 2016.

Robin Burke, Nasim Sonboli, and Aldo Ordonez-Gauger. Balanced neighborhoods for multi-sided fairness in recommendation. In *Proceedings of FAccT*, 2018.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the USENIX Security Symposium*, pages 267–284, 2019.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *Proceedings of the USENIX Security Symposium*, 2021.

Tudor Cebere, Aurélien Bellet, and Nicolas Papernot. Tighter privacy auditing of DP-SGD in the hidden state threat model. In *Proceedings of ICLR*, 2025.

Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. Bundle recommendation with graph convolutional networks. In *Proceedings of SIGIR*, 2020.

Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. Bundle recommendation and generation with graph neural networks. *IEEE Trans. Knowl. Data Eng.*, 35(3):2326–2340, 2023.

Ronshee Chawla, Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. The Gossiping Insert-Eliminate Algorithm for Multi-Agent Bandits. In *Proceedings of AISTATS*, 2020.

Ronshee Chawla, Daniel Vial, Sanjay Shakkottai, and R. Srikant. Collaborative multi-agent heterogeneous multi-armed bandits. In *Proceedings of ICML*, 2023.

Lei Chen, Le Wu, Kun Zhang, Richang Hong, Defu Lian, Zhiqiang Zhang, Jun Zhou, and Meng Wang. Improving recommendation fairness via data augmentation. In *Proceedings of WWW*, 2023.

Sayak Ray Chowdhury and Xingyu Zhou. Differentially private regret minimization in episodic markov decision processes. In *Proceedings of AAAI*, 2022.

Filippos Christianos, Georgios Papoudakis, Matthieu Zimmer, Thomas Coste, Zhihao Wu, Jingxuan Chen, Khyati Khandelwal, James Doran, Xidong Feng, Jiacheng Liu, Zheng Xiong, Yicheng Luo, Jianye Hao, Kun Shao, Haitham Bou-Ammar, and Jun Wang. Pangu-agent: A fine-tunable generalist agent with structured reasoning. *CoRR*, abs/2312.14878, 2023.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Proceedings of NeurIPS*, 2018.

Radu Ciucanu, Pascal Lafourcade, Gael Marcadet, and Marta Soare. SAMBA: A Generic Framework for Secure Federated Multi-Armed Bandits. *Journal of Artificial Intelligence Research*, 73:737–765, 2022.

Chris Cundy, Rishi Desai, and Stefano Ermon. Privacy-constrained policies via mutual information regularized policy gradients. In *Proceedings of AISTATS*, 2024.

Edwige Cyffers and Aurélien Bellet. Privacy amplification by decentralization. In *Proceedings of AISTATS*, 2022.

Anirban DasGupta. *Probability for Statistics and Machine Learning: Fundamentals and Advanced Topics*. Springer Texts in Statistics. Springer, 2011.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

Floris den Hengst, Eoin Grua, Ali el Hassouni, and Mark Hoogendoorn. Reinforcement learning for personalization: A systematic literature review. *Data Science*, 3:1–41, 04 2020.

Damien Desfontaines and Balázs Pejó. Sok: Differential privacies. *Proc. Priv. Enhancing Technol.*, 2020(2):288–313, 2020.

Virginie Do, Sam Corbett-Davies, Jamal Atif, and Nicolas Usunier. Two-sided fairness in rankings via lorenz dominance. In *Proceedings of NeurIPS*, 2021.

Yanqing Duan, John S. Edwards, and Yogesh K Dwivedi. Artificial intelligence for decision making in the era of big data – evolution, challenges and research agenda. *International Journal of Information Management*, 48:63–71, 2019.

Abhimanyu Dubey and Alex Pentland. Private and Byzantine-Proof Cooperative Decision-Making. In *Proceedings of AAMAS*, 2020.

Cynthia Dwork. Differential Privacy. In *Proceedings of ICALP*, 2006.

Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, 2014.

Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. On the Complexity of Differentially Private Data Release: Efficient Algorithms and Hardness Results. In *Proceedings of STOC*, 2009.

Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *Proceedings of the Symposium on Foundations of Computer Science*. IEEE, 2010.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC bounds for multi-armed bandit and markov decision processes. In *Proceedings of COLT*, 2002.

Eyal Even-Dar, Shie Mannor, and Y. Mansour. Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems. *JMLR*, 2006.

Alessandro Fabris, Nina Baranowska, Matthew J. Dennis, David Graus, Philipp Hacker, Jorge Saldivar, Frederik J. Zuiderveen Borgesius, and Asia J. Biega. Fairness and bias in algorithmic hiring: A multidisciplinary survey. *ACM Trans. Intell. Syst. Technol.*, 16(1): 16:1–16:54, 2025.

Flint Xiaofeng Fan, Cheston Tan, Roger Wattenhofer, and Yew-Soon Ong. Position paper: Rethinking privacy in RL for sequential decision-making in the age of LLMs. *CoRR*, abs/2504.11511, 2025.

Minghong Fang, Jia Liu, Michinari Momma, and Yi Sun. Fairroad: Achieving fairness for recommender systems with optimized antidote data. In *Proceedings of SACMAT*, 2022.

Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy amplification by iteration. *CoRR*, abs/1808.06651, 2018.

Raphaël Féraud, Réda Alami, and Romain Laroche. Decentralized Exploration in Multi-Armed Bandits. In *Proceedings of ICML*, 2019.

Yannis Flet-Berliac, Nathan Grinsztajn, Florian Strub, Eugene Choi, Bill Wu, Chris Cremer, Arash Ahmadian, Yash Chandak, Mohammad Gheshlaghi Azar, Olivier Pietquin, and Matthieu Geist. Contrastive policy gradient: Aligning llms on sequence-level scores in a supervised-friendly fashion. In *Proceedings of EMNLP*, 2024.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020.

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. Practical membership inference attacks against fine-tuned large language models via self-prompt calibration. *CoRR*, abs/2311.06062, 2023.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of ICML*, 2019.

Victor Gabillon, Mohammad Ghavamzadeh, Alessandro Lazaric, and Sébastien Bubeck. Multi-Bandit Best Arm Identification. In *Proceedings of NeurIPS*, 2011.

G. Gallo, P. L. Hammer, and B. Simeone. *Quadratic knapsack problems*, pages 132–149. Springer Berlin Heidelberg, 1980.

Evrard Garcelon, Vianney Perchet, Ciara Pike-Burke, and Matteo Pirotta. Local differential privacy for regret minimization in reinforcement learning. In *Proceedings of NeurIPS*, 2021.

Evrard Garcelon, Matteo Pirotta, and Vianney Perchet. Encrypted Linear Contextual Bandit. In *Proceedings of AISTATS*, 2022.

Maziar Gomrokchi, Susan Amin, Hossein Aboutalebi, Alexander Wong, and Doina Precup. Membership inference attacks against temporally correlated data in deep reinforcement learning. *IEEE Access*, 11:42796–42808, 2023.

Çaglar Gülçehre, Ziyu Wang, Alexander Novikov, Thomas Paine, Sergio Gómez Colmenarejo, Konrad Zolna, Rishabh Agarwal, Josh Merel, Daniel J. Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, Jerry Li, Mohammad Norouzi, Matthew Hoffman, Nicolas Heess, and Nando de Freitas. RL unplugged: A collection of benchmarks for offline reinforcement learning. In *Proceedings of NeurIPS*, 2020.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of ICML*, 2018.

Ben M. Hambly, Renyuan Xu, and Huining Yang. Recent advances in reinforcement learning in finance. *CoRR*, abs/2112.04553, 2021.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):1–19, 2015.

Eshcar Hillel, Zohar Karnin, Tomer Koren, Ronny Lempel, and Oren Somekh. Distributed Exploration in Multi-Armed Bandits. In *Proceedings of NeurIPS*, 2013.

Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. Modeling personalized item frequency information for next-basket recommendation. In *Proceedings of SIGIR*, 2020.

Matthew Jagielski, Jonathan R. Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? In *Proceedings of NeurIPS*, 2020.

Kevin Jamieson and Robert Nowak. Best-arm Identification Algorithms for Multi-Armed Bandits in the Fixed Confidence Setting. In *Proceedings of CISS*, 2014.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Proceedings of NeurIPS*, 2019.

Amit Anand Jha, Nazal Mohamed, and Krishna Jagannathan. Collaborative Best Arm Identification in Multi-armed Bandits. In *Proceedings of COMSNETS*, 2022.

Zhengyao Jiang, Tianjun Zhang, Michael Janner, Yueying Li, Tim Rocktäschel, Edward Grefenstette, and Yuandong Tian. Efficient planning in a compact latent action space. In *Proceedings of ICLR*, 2023.

Sham M. Kakade. A natural policy gradient. In *Proceedings of NIPS*, 2001.

Nikolai Karpov and Qin Zhang. Communication-effcient collaborative best arm identifcation. In *Proceedings of AAAI*, 2023.

Nikolai Karpov and Qin Zhang. Communication-efficient collaborative regret minimization in multi-armed bandits. In *Proceedings of AAAI*, 2024.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *CoRR*, abs/2312.14925, 2023.

Michael J. Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Mach. Learn.*, 49(2-3):209–232, 2002.

Balázs Kégl, Gabriel Hurtado, and Albert Thomas. Model-based micro-data reinforcement learning: what are the crucial model properties and which model to choose? In *Proceedings of ICLR*, 2021.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *Proceedings of NeurIPS*, 2020.

Byeongchan Kim and Min Hwan Oh. Model-based offline reinforcement learning with count-based conservatism. In *Proceedings of ICML*, 2023.

Bangalore Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Kumar Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *CoRR*, abs/2002.00444, 2020.

Nima Kordzadeh and Maryam Ghasemaghaei. Algorithmic bias: review, synthesis, and future research directions. *Eur. J. Inf. Syst.*, 31(3):388–409, 2022.

Ravi Kumar Kolla, Krishna Jagannathan, and Aditya Gopalan. Collaborative Learning of Stochastic Bandits Over a Social Network. *Transactions on Networking*, 26(4):1782–1795, 2018.

T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

David Leslie. Understanding artificial intelligence ethics and safety: A guide for the responsible design and implementation of ai systems in the public sector, 2019.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020.

Chen Li, Yuanfu Lu, Wei Wang, Chuan Shi, Ruobing Xie, Haili Yang, Cheng Yang, Xu Zhang, and Leyu Lin. Package recommendation with intra- and inter-package attention networks. In *Proceedings of SIGIR*, 2021a.

Tan Li and Linqi Song. Privacy-Preserving Communication-Efficient Federated Multi-Armed Bandits. *IEEE Journal on Selected Areas in Communications*, 40(3):773–787, 2022.

Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. User-oriented fairness in recommendation. In *Proceedings of WWW*, 2021b.

Chonghua Liao, Jiafan He, and Quanquan Gu. Locally differentially private reinforcement learning for linear mixture markov decision processes. *CoRR*, abs/2110.10133, 2021.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *Proceedings of ICLR*, 2016.

Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. Dependence makes you vulnberable: Differential privacy under dependent tuples. In *Proceedings of the Network and Distributed System Security Symposium*, 2016.

Mingyang Liu, Xiaotong Shen, and Wei Pan. Deep reinforcement learning for personalized treatment recommendation. *Statistics in Medicine*, 41, 2022.

Siqi Liu, Kay Choong See, Kee Yuan Ngiam, Leo Anthony Celi, Xingzhi Sun, and Mengling Feng. Reinforcement learning for clinical decision support in critical care: Comprehensive review. *J Med Internet Res*, 22(7):18477, 2020.

Suyun Liu and Luís Nunes Vicente. Accuracy and fairness trade-offs in machine learning: a stochastic multi-objective approach. *Comput. Manag. Sci.*, 19(3):513–537, 2022.

Cong Lu, Philip J. Ball, Jack Parker-Holder, Michael A. Osborne, and Stephen J. Roberts. Revisiting design choices in offline model based reinforcement learning. In *Proceedings of ICLR*, 2022.

Andrew Luttrell and Richard Petty. Morality matters in the marketplace: The role of moral metacognition in consumer purchasing. *Social Cognition*, 39:328–351, 06 2021.

Paul Luyo, Evrard Garcelon, Alessandro Lazaric, and Matteo Pirotta. Differentially private exploration in reinforcement learning with linear representation. *CoRR*, abs/2112.01585, 2021.

Yunshan Ma, Yingzhi He, Xiang Wang, Yinwei Wei, Xiaoyu Du, Yuyangzi Fu, and Tat-Seng Chua. Multicbr: Multi-view contrastive learning for bundle recommendation. *ACM Trans. Inf. Syst.*, 42(4):100:1–100:23, 2024.

Udari Madhushani and Naomi Leonard. It Doesn't Get Better and Here's Why: A Fundamental Drawback in Natural Extensions of UCB to Multi-agent Bandits. In *Proceedings of "I Can't Believe It's Not Better!" NeurIPS 2020 Workshop*, 2020.

Juan-Gabriel Martínez-Navalón, Vera Gelashvili, and Alba Gómez-Ortega. Evaluation of user satisfaction and trust of review platforms: Analysis of the impact of privacy and e-wom in the case of tripadvisor. *Frontiers in Psychology*, Volume 12, 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of AISTATS*, 2017.

Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *Proceedings of ICLR*, 2018.

Frank D. McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2009.

Rishabh Mehrotra, James McInerney, Hugues Bouchard, Mounia Lalmas, and Fernando Diaz. Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In *Proceedings of CIKM*, 2018.

Ilya Mironov. Renyi Differential Privacy. In *Proceedings of CSF*. IEEE, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of ICML*, 2016.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *Found. Trends Mach. Learn.*, 16(1):1–118, 2023.

Jakob Mökander, Jessica Morley, Mariarosaria Taddeo, and Luciano Floridi. Ethics-based auditing of automated decision-making systems: Nature, scope, and limitations. *Sci. Eng. Ethics*, 27(4):44, 2021.

Ahmadreza Moradipari, Mohammad Ghavamzadeh, and Mahnoosh Alizadeh. Collaborative multi-agent stochastic linear bandits. In *Proceedings of ACC*, 2022.

Mohammadmehdi Naghiaei, Hossein A. Rahmani, and Yashar Deldjoo. CPFair: Personalized Consumer and Producer Fairness Re-ranking for Recommender Systems. In *Proceedings of SIGIR*, 2022.

Dung Daniel T. Ngo, Giuseppe Vietri, and Steven Wu. Improved regret for differentially private exploration in linear MDP. In *Proceedings of ICML*, 2022.

Jianmo Ni, Jiacheng Li, and Julian J. McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of EMNLP-IJCNLP*, 2019.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Proceedings of NIPS*, 2013.

Xinlei Pan, Weiyao Wang, Xiaoshuai Zhang, Bo Li, Jinfeng Yi, and Dawn Song. How You Act Tells a Lot: Privacy-Leaking Attack on Deep Reinforcement Learning. In *Proceeding of AAMAS*, 2019.

Marc S. Paolella. *Linear Models and Time-Series Analysis: Regression, ANOVA, ARMA and GARCH*. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd., 2019.

Gourab K. Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of WWW*, 2020.

Dana Pessach and Erez Shmueli. Algorithmic fairness. *CoRR*, abs/2001.09784, 2020.

Evaggelia Pitoura, Kostas Stefanidis, and Georgia Koutrika. Fairness in rankings and recommendations: an overview. *VLDB*, 31(3):431–458, 2022.

Natalia Ponomareva, Jasmijn Bastings, and Sergei Vassilvitskii. Training text-to-text transformers with privacy guarantees. In *Proceedings of ACL*, 2022.

Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. How to dp-fy ML: A practical guide to machine learning with differential privacy. *J. Artif. Intell. Res.*, 77:1113–1201, 2023.

Kritika Prakash, Fiza Husain, Praveen Paruchuri, and Sujit Gujar. How Private Is Your RL Policy? An Inverse RL Based Analysis Framework. In *Proceeding of AAAI*, 2022.

Dan Qiao and Yu-Xiang Wang. Offline reinforcement learning with differential privacy. In *Proceedings of NeurIPS*, 2023a.

Dan Qiao and Yu-Xiang Wang. Near-optimal differentially private reinforcement learning. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent, editors, *Proceedings of AISTATS*, 2023b.

Zhiwei (Tony) Qin, Hongtu Zhu, and Jieping Ye. Reinforcement learning for ridesharing: A survey. *CoRR*, abs/2105.01099, 2021.

Bashir Rastegarpanah, Krishna P Gummadi, and Mark Crovella. Fighting fire with fire: Using antidote data to improve polarization and fairness of recommender systems. In *Proceedings of WSDM*, 2019.

Wenbo Ren, Xingyu Zhou, Jia Liu, and Ness B. Shroff. Multi-Armed Bandits with Local Differential Privacy. *CoRR*, 2020.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of WWW*, 2010.

Maria Rigaki and Sebastián García. A survey of privacy attacks in machine learning. *ACM Comput. Surv.*, 56(4):101:1–101:34, 2024.

Alexandre Rio, Merwan Barlier, Igor Colin, and Marta Soare. Multi-agent best arm identification with private communications. In *Proceedings of ICML*, 2023.

Alexandre Rio, Merwan Barlier, Igor Colin, and Albert Thomas. Differentially private deep model-based reinforcement learning. *arXiv preprint arXiv:2402.05525*, 2024.

Alexandre Rio, Merwan Barlier, and Igor Colin. Differentially private policy gradient. *arXiv preprint arXiv:2501.19080*, 2025.

Herbert E. Robbins. Some Aspects of the Sequential Design of Experiments. *Bulletin of American Mathematical Society*, 58(5):527–535, 1952.

Teresa Rodríguez de las Heras Ballell. Guiding principles for automated decision-making in the eu. European Law Institute Innovation Paper, 2022.

Senjuti Basu Roy, Sihem Amer-Yahia, Ashish Chawla, Gautam Das, and Cong Yu. Constructing and exploring composite items. In *Proceedings of SIGMOD*, 2010.

Clémence Réda, Sattar Vakili, and Emilie Kaufmann. Near-Optimal Collaborative Learning in Bandits. In *Proceedings of NeurIPS*, 2022.

Touqir Sajed and Or Sheffet. An Optimal Private Stochastic-MAB Algorithm based on Optimal Private Stopping Rule. In *Proceedings of ICML*, 2019.

Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Proceedings of NIPS*, 2007.

Noah Salas, Brock Ferguson, and Jacob Zweig. Reinforcement learning for personalized medication dosing. Blog post, 2019.

Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. Interventional fairness: Causal database repair for algorithmic fairness. In *Proceedings of SIGMOD*, 2019.

Abishek Sankararaman, Ayalvadi Ganesh, and Sanjay Shakkottai. Social learning in multi agent multi armed bandits. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–35, 2019.

Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. Overlapping Multi-Bandit Best Arm Identification. In *Proceedings of ISIT*, 2019.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of ICML*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *Proceedings of COLT*, 2009.

Xuedong Shang, Emilie Kaufmann, and Michal Valko. A simple dynamic bandit algorithm for hyper-parameter tuning. In *ICML Workshop on Automated Machine Learning*, 2019.

Chengshuai Shi, Haifeng Xu, Wei Xiong, and Cong Shen. (Almost) Free Incentivized Exploration from Decentralized Learning Agents. In *Proceedings of NeurIPS*, 2021.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of the Symposium on Security and Privacy*. IEEE, 2017.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin A. Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of ICML*, 2014.

Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of SIGKDD*, 2018.

Bharat Singh, Rajesh Kumar, and Vinay Pratap Singh. Reinforcement learning in robotic applications: a comprehensive survey. *Artif. Intell. Rev.*, 55(2):945–990, 2022.

Evangelos D. Spyrou, Vassilios Kappatos, Alkiviadis Tromaras, and Afroditi K. Anagnostopoulou. A multi-armed bandit approach for electricity reward of smart homes in a smart power grid. In *Proceedings of CoDIT*, 2024.

Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of NIPS*, 1999.

Balazs Szörényi, Robert Busa-Fekete, Istvan Hegedüs, Robert Ormándi, Mark Jelasity, and Balazs Kégl. Gossip-based Distributed Stochastic Bandit Algorithms. In *Proceedings of ICML*, 2013.

Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. *CoRR*, abs/2408.03539, 2024.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *CoRR*, abs/1801.00690, 2018.

William R. Thompson. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3/4):285–294, 1933.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IROS*, 2012.

Aristide Tossou and Christos Dimitrakakis. Algorithms for Differentially Private Multi-Armed Bandits. In *Proceedings of AAAI*, 2016.

Nicolas Usunier, Virginie Do, and Elvis Dohmatob. Fast online ranking with fairness of exposure. In *Proceedings of FAccT*, 2022.

Filippo Vannella. *Bandit Methods for Network Optimization: Safety, Exploration, and Coordination*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2023.

Yogatheesan Varatharajah, Brent M. Berry, Sanmi Koyejo, and Ravishankar K. Iyer. A contextual-bandit-based approach for informed decision-making in clinical trials. *CoRR*, abs/1809.00258, 2018.

Giuseppe Vietri, Borja Balle, Akshay Krishnamurthy, and Zhiwei Steven Wu. Private reinforcement learning with PAC and regret guarantees. In *Proceedings of ICML*, 2020.

Baoxiang Wang and Nidhi Hegde. Privacy-preserving q-learning with functional noise in continuous spaces. In *Proceedings of NeurIPS*, 2019.

Letian Wang, Jie Liu, Hao Shao, Wenshuo Wang, Ruobing Chen, Yu Liu, and Steven L. Waslander. Efficient reinforcement learning for autonomous driving with parameterized skills and priors. In *Proceedings of Robotics: Science and Systems*, 2023.

Christopher Watkins. *Learning From Delayed Rewards*. Phd thesis, University of Cambridge, 1989.

Christopher Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

Pawel Wawrzynski. A cat-like robot real-time learning to run. In *Proceedings of Adaptive and Natural Computing Algorithms*, 2009.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Yao Wu, Jian Cao, Guandong Xu, and Yudong Tan. TFROM: A two-sided fairness-aware recommendation model for both customers and providers. In *Proceedings of SIGIR*, 2021.

Min Xie, Laks V. S. Lakshmanan, and Peter T. Wood. Composite recommendations: from items to packages. *Frontiers Comput. Sci.*, 6(3):264–277, 2012.

Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments. In *Proceedings of NeurIPS*, 2020.

Qin Yang and Young-Chan Lee. Ethical AI in financial inclusion: The role of algorithmic fairness on user satisfaction and recommendation. *Big Data Cogn. Comput.*, 8(9):105, 2024.

Sirui Yao and Bert Huang. Beyond parity: Fairness objectives for collaborative filtering. In *Proceedings of NeurIPS*, 2017.

Yelp. Yelp open dataset, 2023.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Proceedings of CSF*, 2018.

Jialin Yi. Regret-minimization algorithms for multi-agent cooperative learning systems. *CoRR*, abs/2310.19468, 2023.

Chao Yu, Jiming Liu, and Shamim Nemati. Reinforcement learning in healthcare: A survey. *CoRR*, abs/1908.08796, 2019.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: model-based offline policy optimization. In *Proceedings of NeurIPS*, 2020.

Jiaming Zhang, Mingxi Lei, Meng Ding, Mengdi Li, Zihang Xiang, Difei Xu, Jinhui Xu, and Di Wang. Towards user-level private reinforcement learning with human feedback. *CoRR*, abs/2502.17515, 2025.

Jun Zhao, Teng Wang, Tao Bai, Kwok-Yan Lam, Xuebin Ren, Xinyu Yang, Shuyu Shi, Yang Liu, and Han Yu. Reviewing and improving the gaussian mechanism for differential privacy. *CoRR*, abs/1911.12060, 2019.

Xingyu Zhou. Differentially Private Reinforcement Learning with Linear Function Approximation. In *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2022.

Tao Zhu, Patrick Harrington, Junjun Li, and Lei Tang. Bundle recommendation in ecommerce. In *Proceedings of SIGIR*, 2014.

Ziyuan Ziyuan, Anchen Sun, Jingyi Jingyi, Yuanzhe Yuanzhe, Rui Rui, and Minghui Minghui. Contemporary recommendation systems on big data and their applications: A survey. *IEEE Access*, 12:196914–196928, 2024.