electronics

Article

Safe-Learning-Based Location-Privacy-Preserved Task Offloading in Mobile Edge Computing

Minghui Min [1,2,*], Zeqian Liu [1], Jincheng Duan [1], Peng Zhang [1] and Shiyin Li [1]

[1] School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China; 04181454@cumt.edu.cn (Z.L.)
[2] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education and School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
* Correspondence: minmh@cumt.edu.cn

Abstract: Mobile edge computing (MEC) integration with 5G/6G technologies is an essential direction in mobile communications and computing. However, it is crucial to be aware of the potential privacy implications of task offloading in MEC scenarios, specifically the leakage of user location information. To address this issue, this paper proposes a location-privacy-preserved task offloading (LPTO) scheme based on safe reinforcement learning to balance computational cost and privacy protection. This scheme uses the differential privacy technique to perturb the user's actual location to achieve location privacy protection. We model the privacy-preserving location perturbation problem as a Markov decision process (MDP), and we develop a safe deep Q-network (DQN)-based LPTO (SDLPTO) scheme to select the offloading policy and location perturbation policy dynamically. This approach effectively mitigates the selection of high-risk state–action pairs by conducting a risk assessment for each state–action pair. Simulation results show that the proposed SDLPTO scheme has a lower computational cost and location privacy leakage than the benchmarks. These results highlight the significance of our approach in protecting user location privacy while achieving improved performance in MEC environments.

Keywords: mobile edge computing; location privacy; differential privacy; location perturbation; safe reinforcement learning

Citation: Min, M.; Liu, Z.; Duan, J.; Zhang, P.; Li, S. Safe-Learning-Based Location-Privacy-Preserved Task Offloading in Mobile Edge Computing. *Electronics* **2024**, *13*, 89. https://doi.org/10.3390/electronics13010089

Academic Editor: Javid Taheri

Received: 10 November 2023
Revised: 18 December 2023
Accepted: 21 December 2023
Published: 25 December 2023

1. Introduction

With the continuous development of mobile communication technology, mobile edge computing (MEC) has emerged as a new computing paradigm and has gained significant attention and application. MEC has been successfully applied in various major scenarios such as autonomous driving, smart cities, smart homes, virtual reality, and facial recognition [1,2]. However, during edge computing applications, the leakage of user location privacy poses a potentially significant risk [3].

Traditional task offloading strategies typically focus on reducing latency and network resource consumption, but they often overlook the protection of user location privacy. For instance, attackers can infer user location information by monitoring the status of edge servers or wireless channels [4–6]. Therefore, when users offload their computation tasks to edge servers in the context of MEC, it is important to minimize latency and network resource consumption and ensure user privacy protection.

In order to protect user location privacy while meeting the performance requirements of MEC applications, such as response time and energy consumption, it is necessary to take corresponding technical measures. These measures include location perturbation, anonymization, and differential privacy [7–9]. However, current research on location privacy protection in MEC applications has the following major shortcomings:

1. The system lacks universality and flexibility and is only suitable for static scenarios or single-application demands [10,11], making it difficult to adapt to the ever-changing MEC scenarios.
2. The system falls short of providing adequate security guarantees, leaving it vulnerable to internal and third-party attacks that may compromise the accuracy of location data [12,13]. The lack of a trusted identity verification mechanism in the authentication system increases the risk of user location information leakage.
3. The system lacks a dynamic balance between the computational cost of MEC systems and users' privacy requirements. Notably, the protection of user location privacy can lead to an increase in the computational burden on MEC servers [14].

To address the aforementioned shortcomings, enhancing the comprehensiveness of privacy protection mechanisms in MEC applications is necessary. This includes establishing secure communication channels, employing secure computing techniques, and taking measures to prevent untrusted third-party attackers from leaking user location privacy [15,16]. In MEC applications, the distance and wireless channel conditions between users and edge servers are closely related; the closer the distance, the better the channel conditions, and the farther the distance, the worse the channel conditions [17]. If the edge server is untrusted or compromised, attackers can infer wireless channel information by monitoring users' task offloading rate and deducing their location information. In the context of MEC systems, location perturbation is crucial in preserving user privacy by adding noise or modifying sensitive location information. Differential-privacy-based location perturbation techniques have been studied to protect users' location privacy [18]. Thus, this paper investigates location privacy protection in MEC systems based on differential privacy.

This paper aims to investigate the issue of location privacy protection in MEC applications. In addition, different tasks have varying sensitivities to energy consumption and computation delays, resulting in different energy consumption and latency requirements. Additionally, it is significant to consider the dynamic trade-off between computation cost and user privacy requirements. By tackling the challenges associated with location privacy protection and enhancing MEC performance, we can attain sustainable development of location services and optimize MEC applications. To address the trade-off between privacy protection level and energy consumption/latency performance in MEC, we can design an objective function that considers both location privacy and computation cost, aiming to maximize the overall performance of the MEC system.

Traditional research on balancing privacy protection, energy consumption, and latency has often relied on rule-based approaches, which typically require predefined rules and models [7]. These approaches neglect the dynamic network environment and fail to adapt to complex and changing protection requirements. In contrast, reinforcement learning (RL) stands out by learning through iterative interaction with the environment, facilitating adjustments to environmental changes and uncertainties. Unlike traditional decision-making approaches that often require manual strategy adjustments, RL achieves adaptive strategy refinement through trial-and-error learning. Moreover, deep RL (DRL) combines the strengths of deep learning and RL, employing neural networks to glean insights from the environment and optimize decision-making strategies. As a result, the integration of DRL enhances decision effectiveness, providing a more robust framework for addressing complicated challenges in the dynamic environment. Notably, the inclusion of safe learning mechanisms empowers the learning agent to avoid the selection of high-risk state–action pairs [19]. Thus, we develop a safe DRL algorithm to solve the designed problem.

This paper proposes a safe deep Q-network (DQN)-based location-privacy-preserved task offloading (SDLPTO) scheme to dynamically balance computational cost and privacy protection. This scheme utilizes differential privacy techniques to protect user location privacy while considering the trade-off between energy consumption, latency, and privacy protection. Simulation results demonstrate the performance advantage of our proposed scheme compared to benchmarks. The main innovations of this paper can be summarized as follows:

1. We propose a location-privacy-aware task offloading framework that utilizes differential privacy technology to design a perturbed distance probability density function, making it difficult for attackers to infer the user's actual location from a fake one.
2. We model the privacy-preserving location perturbation problem as a Markov decision process (MDP). We use the DRL method to adaptively select location-privacy-preserved task offloading (LPTO) policies to avoid location privacy leakage while ensuring computational performance in a dynamic MEC system. This solution can jointly consider location privacy and computational offloading performance, enabling a balance between them.
3. We develop an SDLPTO scheme to find the optimal location-privacy-preserved task offloading policy. We utilize the DQN algorithm to capture the system state and accelerate policy selection in a dynamic environment. Meanwhile, we implement a safe exploration algorithm for location perturbation and offloading decisions, mitigating potential losses from high-risk state–action pairs.
4. Simulation results demonstrate that our proposed SDLPTO better balances location privacy and offloading costs. This scheme consistently outperforms benchmark schemes across various task sizes and perturbation distance ranges, demonstrating its advantage in preserving location privacy while minimizing offloading overhead.

The subsequent parts of the paper are organized as follows: Section 2 discusses related work. Section 3 presents the proposed system model, location perturbation model, and problem formulation. Section 4 introduces a safe DQN-based location-privacy-preserved task offloading scheme. Section 5 gives simulation results and performance analysis, and Section 6 concludes the paper.

2. Related Work

As a distributed computing model that pushes data processing and storage to the network edge, edge computing has been expanding its application scope, but privacy issues have become increasingly prominent. Data encryption, K-anonymity, blockchain, and location perturbation techniques [7,20–22] have been studied in the context of privacy protection in MEC. More specifically, location perturbation is a technique employed to protect privacy by introducing modifications or perturbations to the original location data. Various technologies, such as differential privacy, path cloaking, temporal clustering, and location truncation, can be utilized for location perturbation [3,23,24].

In recent years, many works have utilized differential-privacy-based location perturbation techniques to effectively protect users' location privacy [14,18]. Differential privacy technology has superior privacy protection effects and can prevent attackers from re-identifying data based on known background knowledge [18,25]. In [18], Wang et al. propose a location-privacy-aware task offloading framework (LPA-Offload) that protects user location privacy by using the location perturbation mechanism based on differential privacy. The scheme formulates the optimal offloading strategy based on an iterative method and then calculates the computation cost and privacy leakage. In [25], Miao et al. propose an MEPA privacy-aware framework for MEC that uses differential privacy technology to protect location privacy in the dataset domain. A privacy-preserving computation offloading scheme based on the whale optimization algorithm is proposed in [7]. This scheme uses differential privacy technology to perturb users' locations and makes offloading decisions based on the perturbed distance. However, this scheme faces challenges in adapting to dynamic environments. It fails to derive an effective location perturbation strategy despite proposing an algorithm to address the convex optimization problem of computation offloading under a given privacy budget. The previous studies mentioned do not consider preserving privacy while optimizing for delay and energy consumption in edge computing. Alternatively, some studies consider privacy preservation but fail to optimize these factors simultaneously. Furthermore, it is important to note that the aforementioned methods are designed for static scenarios and cannot effectively address optimization challenges in dynamic environments.

RL technology has been widely used in dynamic MEC systems [26–28], and one of its most important applications is to protect user privacy. The algorithm has the characteristic of adaptive learning and can automatically adjust the learning strategy according to changes in data and the environment. It also uses distributed storage technology to store user data on multiple nodes, thereby preventing user data from being stolen or tampered with by attackers [29]. In [17], Min et al. propose a scheme that can protect both user location privacy and user pattern privacy and study a privacy-aware offloading scheme based on RL, which can reduce computation latency and energy consumption and improve the privacy level of medical IoT devices. In [29], Liu et al. propose a privacy-preserving distributed deep deterministic policy gradient (P2D3PG) algorithm that solves the problem of maximizing the distributed edge caching (EC) hit rate under privacy protection constraints in a wireless communication system with MEC. In [14], Zhang et al. studied differential privacy and RL task transfer strategy, established an MEC system model, and designed a four-layer policy network as an RL agent, but lacked a balance between privacy and computation offloading performance. To solve the above problems, our work proposes an RL-based algorithm that achieves a balance between privacy protection and computation offloading performance by combining differential privacy and RL technology.

3. System Model and Problem Formulation

In this section, we present the system model for computation offloading in MEC, the location perturbation model, followed by the specific formulation of the design objective function.

3.1. System Model

We assume that the MEC system consists of a MEC server and a user with mobile devices, similar to the work in [17,18]. The user's offloading strategy takes into account the distance between the user and the MEC server, as well as the wireless channel conditions. In scenarios where the distance is short, and the channel quality is good, users are more likely to prefer a high offloading rate, offloading more tasks to the edge server to improve performance. Conversely, in scenarios where the distance is long, or the channel quality is poor, users may prefer a lower offloading rate, executing more tasks locally to reduce computational costs. In order to protect the user's location privacy, we perturb the user's real location into a fake location, and we perform task offloading through the fake location. The user's location perturbation and the offloading process between the user and the MEC server are shown in Figure 1.

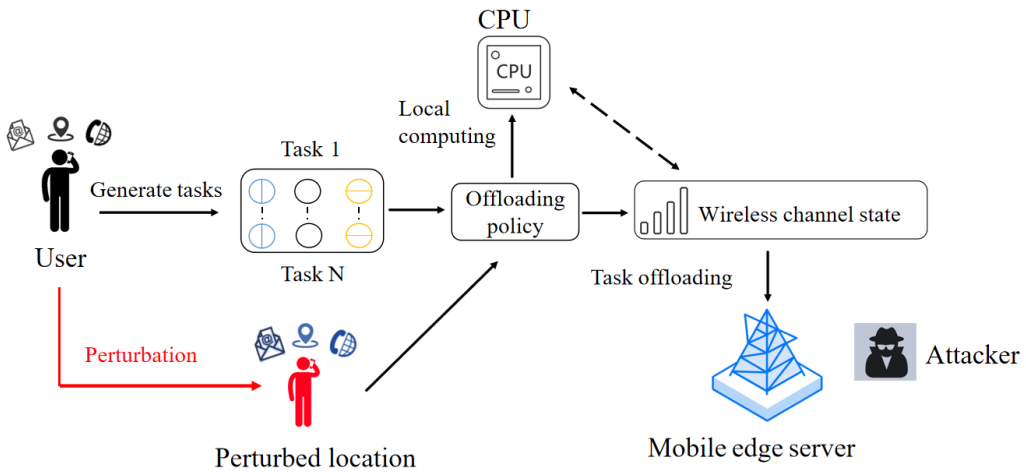

Figure 1. Illustration of location-privacy-preserved task offloading in MEC.

3.2. Offloading Model

At each time slot k, we assume that the user needs to execute a total of $v^{(k)}$ (in bits) computational tasks. The offloading ratio, denoted as $\chi^{(k)} \in [0,1]$, represents the proportion of total tasks that the user chooses to offload to the MEC server. Accordingly, the user has two offloading strategies: $\chi^{(k)}v^{(k)}$ bit computation tasks are offloaded to the MEC server, while the remaining $\left(1-\chi^{(k)}\right)v^{(k)}$ tasks are processed locally [30].

Local computing model. Let f_l and ϕ denote the CPU frequency and the number of CPU cycles for the mobile device to process the task, respectively. Accordingly, the energy consumption $E_l^{(k)}$ [30] and computational delay $T_l^{(k)}$ [31] for local computation on a mobile device are given by

$$E_l^{(k)} = \kappa f_l^2 (1-\chi^{(k)}) v^{(k)} \phi \tag{1}$$

$$T_l^{(k)} = \frac{\left(1-\chi^{(k)}\right) v^{(k)} \phi}{f_l} \tag{2}$$

where κ represents the effective energy coefficient associated with the chip architecture.

Edge computing model. When the channel state is better, the mobile device sends tasks to the edge server through the channel and offloads them to the edge server, which can reduce energy consumption and latency.

There are three main processes for task offloading to the MEC server: user task upload, MEC server computation, and computation result return. Due to the small number of tasks to be returned, the returned results are usually much smaller than the size of the input data, so for this paper, we ignore the communication delay of the computed results when they are returned. The data transmission rate $R^{(k)}$ for offloading tasks from the user to the MEC server can be established as

$$R^{(k)} = B_0 \log_2 \left(1 + \frac{h^{(k)} p}{N_0 B_0}\right) \tag{3}$$

where B_0 denotes the communication bandwidth, $h^{(k)}$ denotes the channel gain between the user and the MEC server, N_0 is the noise power, and p is the transmission power when the user offloads the task. And $h^{(k)} = (d^{(k)})^{-\vartheta}$, where ϑ is the path loss index, and d is the distance from the user to the server. According to Equation (3), it can be concluded that the distance between the user and the server affects the state of the wireless communication channel, and the further the distance, the worse the state of the wireless communication, and the distance between the user and the edge server is related to the condition of $R^{(k)}$.

The computational latency of the edge server consists of the data transfer time and the execution time of the task on the edge server. Considering that the output data size is usually much smaller than the input data, we can ignore the time overhead of transferring data from the edge server to the mobile user. Therefore, the computational delay $T_s^{(k)}$ and energy consumption $E_s^{(k)}$ for offloading are given by

$$T_s^{(k)} = \frac{\chi^{(k)} v^{(k)}}{R^{(k)}} + \frac{\chi^{(k)} v^{(k)} \phi}{f_s} \tag{4}$$

$$E_s^{(k)} = \frac{p \chi^{(k)} v^{(k)}}{R^{(k)}} \tag{5}$$

where f_s denotes the CPU-cycle frequency of the MEC server.

To sum up, at time slot k, the total execution delay $T^{(k)}$ and energy consumption $E^{(k)}$ are given by

$$T^{(k)} = \max\left\{T_l^{(k)}, T_s^{(k)}\right\} \tag{6}$$

$$E^{(k)} = E_l^{(k)} + E_s^{(k)} \tag{7}$$

3.3. Location Perturbation Model

Due to the limited coverage range of the edge servers within a local area, applying traditional differential privacy techniques to the MEC servers for task offloading faces challenges [18]. Therefore, a novel distance perturbation probability density function is used to protect the user's location privacy. Users can use this function to perturb the distance between themselves and the edge server, ensuring the security of their location information and preventing any potential leakage. Assume that the maximum coverage radius of the edge server is d_{max}, the upper bound of the user perturbation distance range is d_1, and the lower bound is d_2, $\Delta d = d_2 - d_1$, with $d_1 < d_2$ and $d_1, d_2 \in [0, d_{max}]$. The probability density function $P(d^* \mid d)$ for perturbing the true distance d to fake distance d^* is set as follows:

$$P(d^* \mid d) = \begin{cases} \frac{\varepsilon}{2\Delta d} e^{-\frac{\varepsilon |d^* - d|}{\Delta d}} + \frac{e^{\frac{\varepsilon (d_1 - d)}{\Delta d}}}{2\Delta d} + \frac{e^{-\frac{\varepsilon (d_2 - d)}{\Delta d}}}{2\Delta d}, & \text{if } d^* \in [d_1, d_2] \\ 0 & \text{, otherwise} \end{cases} \tag{8}$$

We apply differential privacy techniques to introduce randomness to neighboring datasets, thereby making it challenging for attackers to deduce the user's actual location from the fake positions. The process of utilizing the perturbation probability function is depicted in Figure 2.

Figure 2. Location privacy protection process based on differential privacy.

Kullback–Leibler divergence (KL divergence) [32] can also be referred to as relative entropy. It is a method used to quantify the degree of fit between the probability distribution of mechanisms $P(d^* \mid d)$ and privacy-preserving mechanisms without privacy protection $Q(d^* \mid d)$, where d is the true distance and d^* is the perturbed distance between the user and the edge server. The KL divergence is defined as

$$K_{LD}(P \| Q) = \int_{d_1}^{d_2} Q(d^* \mid d) \log \frac{Q(d^* \mid d)}{P(d^* \mid d)} dd^* \tag{9}$$

According to the definition of KL divergence, a smaller value of $K_{LD}(P||Q)$ indicates a higher degree of fit between $P(d^* \mid d)$ and $Q(d^* \mid d)$, resulting in a higher probability of user location information leakage. Conversely, as the value of $K_{LD}(P||Q)$ increases, it

indicates a better fit between $P(d^* \mid d)$ and $Q(d^* \mid d)$, resulting in a lower probability of user location information leakage. Therefore, the degree of privacy leakage PL_{d_1,d_2} can be defined as

$$PL_{d_1,d_2} = -\int_{d_1}^{d_2} Q(d^* \mid d) \log \frac{Q(d^* \mid d)}{P(d^* \mid d)} dd^* \tag{10}$$

where $Q(d^* \mid d)$ denotes the probability distribution of offloading the task at the true distance between the user and the server, and $P(d^* \mid d)$ denotes the probability distribution of perturbing the true distance to a fake distance by adding a differential privacy mechanism.

3.4. The Proof of Differential Privacy Guarantee

Given the true distance d between the user and the server and the neighborhood $d^{'}$, the probability of d being perturbed to d^* is $\Pr(d^* \mid d)$. The probability of $d^{'}$ being perturbed to d^* is $\Pr(d^* \mid d^{'})$. The ratio of $\Pr(d^* \mid d)$ to $\Pr(d^* \mid d^{'})$ satisfies the definition of ε-differential privacy. The detailed proof is given as follows:

$$\begin{aligned}
\frac{\Pr(d^* \mid d)}{\Pr(d^* \mid d')} &= \frac{\frac{\epsilon}{2\Delta d} e^{-\frac{\epsilon|d^*-d|}{\Delta d}} + \frac{e^{\frac{\epsilon(d_1-d)}{\Delta d}} + e^{-\frac{\epsilon(d_2-d)}{\Delta d}}}{2\Delta d}}{\frac{\epsilon}{2\Delta d} e^{-\frac{\epsilon|d^*-d'|}{\Delta d}} + \frac{e^{\frac{\epsilon(d_1-d')}{\Delta d}} + e^{-\frac{\epsilon(d_2-d')}{\Delta d}}}{2\Delta d}} \\
&= \frac{\frac{\varepsilon}{2\Delta d} e^{-\frac{\varepsilon|d^*-d|}{\Delta d}} + \frac{\varepsilon e^{\frac{\varepsilon(d_1-d)}{\Delta d}}}{2\Delta d} + \frac{\varepsilon e^{-\frac{\varepsilon(d_2-d)}{\Delta d}}}{2\Delta d}}{\frac{\varepsilon}{2\Delta d} e^{-\frac{\varepsilon|d^*-d'|}{\Delta d}} + \frac{\varepsilon e^{\varepsilon(d_1-d')}}{2\Delta d} + \frac{\varepsilon e^{-\varepsilon(d_2-d')}}{2\Delta d}} \\
&\leq \max\left(e^{\frac{\varepsilon}{\Delta d}|d^*-d'|-|d^*-d|}, e^{\frac{\varepsilon}{\Delta d}|d_1-d|-|d_1-d'|}, e^{\frac{\varepsilon}{\Delta d}|d_2-d'|-|d_2-d|}\right) \\
&\leq \max(e^{\varepsilon}, e^{\varepsilon}, e^{\varepsilon}) \\
&= e^{\varepsilon}
\end{aligned} \tag{11}$$

3.5. Problem Formulation

The system's total cost to process the user terminal device task is $C^{(k)}$. In the context of mobile edge computing, energy consumption and latency are the two most commonly used metrics to measure the performance of offloading schemes. Considering the computational cost required during task offloading comprehensively, the total computational cost consists of computational latency and energy consumption, which is expressed as

$$C^{(k)} = \lambda \cdot T^{(k)} + (1-\lambda) \cdot E^{(k)} \tag{12}$$

where $\lambda \in (0,1)$ is defined as the weight of balancing energy consumption and computational latency in task offloading.

Users can automatically select a suitable offload strategy based on factors such as the current network congestion level and the distance between the user and the server and develop an offloading scheme targeting the minimum computational cost based on the state conditions of the wireless channel. The designed objective in this paper is to minimize the weighted sum of the computational cost $C^{(k)}$ and the level of privacy leakage $PL^{(k)}$ (i.e., maximize the utility $U^{(k)}$), which is given by

$$\begin{aligned}
\max U^{(k)} &= \max\ -\left[(1-\omega)C^{(k)} + \omega PL^{(k)}\right] \\
&= \max\ -(1-\omega)\left[\lambda \cdot T^{(k)} + (1-\lambda) \cdot E^{(k)}\right] - \omega PL^{(k)}
\end{aligned} \tag{13}$$

where $\omega \in (0,1)$ is an influencing factor reflecting the user's concern about the level of privacy leakage. The larger ω is, the more concerned users are about the degree of privacy leakage. We adjust ω to weight the privacy and computational costs.

4. Safe DQN-Based Location-Privacy-Preserved Task Offloading

It is typically difficult to employ traditional optimization techniques to obtain the optimal location-privacy-preserved task offloading policy in a dynamic MEC system. In this section, we will show how to utilize a safe DRL method to protect the user's location privacy while ensuring the performance of MEC. In detail, we first model the privacy-preserving location perturbation problem as an MDP [33]. Then, we propose an SDLPTO scheme in which risk assessment is performed on state–action pairs to avoid the selection of high-risk disturbance policies, as shown in Figure 3.

The system's next state is only related to the state and selected policy of the current time slot. Hence, the MDP model can model the location-privacy-preserved task offloading process. Therefore, we can use RL technology to dynamically explore the optimal location-privacy-preserved task offloading policy [34]. We define the state, action, reward, and risk level function of the SDLPTO scheme, which can be represented by a tuple $(\boldsymbol{s}, \boldsymbol{a}, U, l)$.

- **State:** $\boldsymbol{s}^{(k)} = [V^{(k)}, R^{(k-1)}]$ is the system state at time slot k, $\boldsymbol{s}^{(k)} \in \mathcal{S}$, where $\mathcal{S}$ is the state set. Before optimizing the performance of the edge computing system and the degree of privacy leakage, we set the number of tasks $V^{(k)}$ generated by user devices and the wireless channel condition $R^{(k)}$ between the user and the edge server to the environment state.
- **Action:** $\boldsymbol{a}^{(k)} = [\chi^{(k)}, \varepsilon^{(k)}]$ is the system state at time slot k, $\boldsymbol{a}^{(k)} \in \mathcal{A}$, where $\mathcal{A}$ is the action set. We use the task offloading ratio $\chi^{(k)}$ and privacy budget $\varepsilon^{(k)}$ as actions which affect the computational offloading decision and privacy leakage situation, and perturbation location d^*, respectively.
- **Reward:** Considering several factors and long-term optimization, we define the utility as a weighted sum of energy consumption, latency, and privacy leakage level, which can be expressed as $U^{(k)}$.
- **Risk level function:** The risk level of taking $\boldsymbol{a}^{(k)}$ in state $\boldsymbol{s}^{(k)}$ is $l(\boldsymbol{s}, \boldsymbol{a})$ at time slot k.

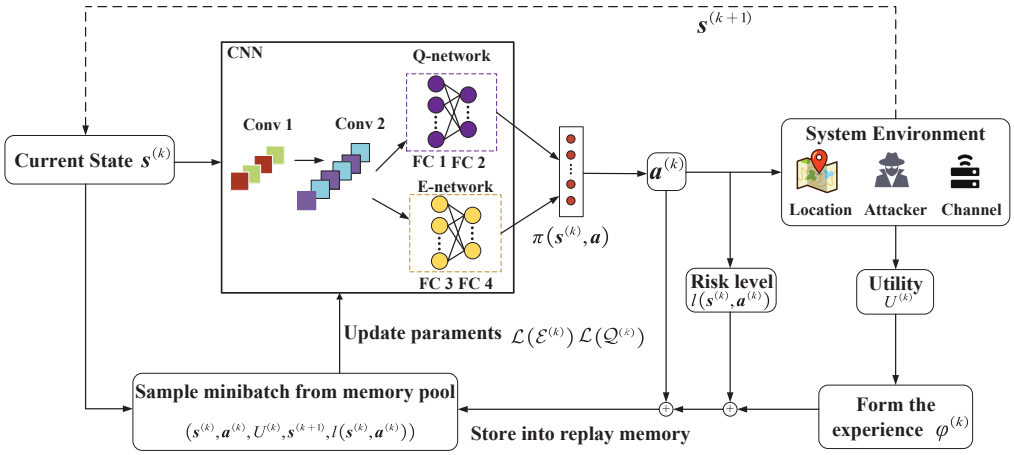

Figure 3. Illustration of the safe DQN-based location-privacy-preserved task offloading (SDLPTO) structure.

The risk level of the current state–action pair $l(\boldsymbol{s}, \boldsymbol{a})$ is evaluated by the user based on the privacy leakage level $p(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)})$, which is evaluated based on Equation (10). It represents the extent of privacy leakage caused by the perturbation policy $\boldsymbol{a}^{(k)}$ in state $\boldsymbol{s}^{(k)}$. We assume that there are L risk levels, with the highest risk level being Lh, representing the most dangerous behavior state. Conversely, zero represents the lowest risk level. We define

$\{\xi_d\}_{0\le d\le Lh}$ as the safety performance indicators with L risk thresholds. Consequently, similar to, the risk level $l(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)})$ can be evaluated by

$$l(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}) = \sum_{d=0}^{Lh} \mathrm{I}(p(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}) > \xi_d) \tag{14}$$

where $\mathrm{I}\{\cdot\}$ is an indicator function.

Although the user evaluates the current state–action pair's risk level by $l(\boldsymbol{s}^{(k)}, \boldsymbol{a})$, the location-privacy-preserved task offloading policy may result in severe privacy leakage. Therefore, the user also estimates the long-term risk level $E(\boldsymbol{s}^{(k)}, \boldsymbol{a})$ of the previous location-privacy-preserved task offloading policies to estimate their impact on the future by tracing back the prior $\mathcal{B}$ experienced state–action pairs, which are given by

$$E(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}) = \sum_{e=0}^{\lambda-1} \gamma^e l(\boldsymbol{s}^{(k+e)}, \boldsymbol{a}^{(k+e)}) \tag{15}$$

where γ is the decay factor.

The long-term expected reward (Q-value) of the user that adopts the perturbation policy $\boldsymbol{a}^{(k)}$ at state $\boldsymbol{s}^{(k)}$ is updated as follows:

$$Q\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}\right) \leftarrow (1-\alpha) Q\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}\right) + \alpha\left(U^{(k)} + \delta \max Q\left(\boldsymbol{s}^{(k+1)}, \boldsymbol{a}^{(k)}\right)\right) \tag{16}$$

where $\alpha \in [0,1]$ is the learning rate and $\delta \in (0,1]$ is the discount factor.

The user takes account of both the Q-value and E-value while selecting the location-privacy-preserved task offloading policy. The policy function $\pi(\boldsymbol{s}^{(k)}, \boldsymbol{a})$ is the probability distribution of selecting the offloading policy and location perturbation policy $\boldsymbol{a}^{(k)}$ in the current state $\boldsymbol{s}^{(k)}$, which is given by

$$\pi\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}\right) = \frac{\exp\left(\frac{Q\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}\right)}{E\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}\right)+1}\right) \mathrm{I}\left(l\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}\right) = L\right)}{\sum_{\boldsymbol{a}' \in \mathcal{A}} \exp\left(\frac{Q\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}\right)}{E\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}'\right)+1}\right) \mathrm{I}\left(l\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}'\right) = L\right)} \tag{17}$$

At time slot k, based on the current state $\boldsymbol{s}^{(k)}$, the user selects the location-privacy-preserved task offloading policy $\pi(\boldsymbol{s}^{(k)}, \boldsymbol{a})$ according to Equation (17). Then, the user executes the action $\boldsymbol{a}^{(k)} = [\chi^{(k)}, \varepsilon^{(k)}]$ and obtains the system reward $U^{(k)}$ after evaluating the energy consumption, latency, and privacy leakage level. Then, the system state transfers to the next state $\boldsymbol{s}^{(k+1)}$.

The experience feedback technique is an important part of the DQN algorithm. The transitions $\Upsilon^{(k)} = (\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}, U^{(k)}, \boldsymbol{s}^{(k+1)}, l(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}))$ are stored in a storage pool $\mathcal{B}$, and then some experiences from $\mathcal{B}$ are randomly selected to train on a small batch $\mathcal{M}$. The system state $\boldsymbol{s}^{(k)}$ is extended to the location-privacy-preserved task offloading experience sequence denoted by $\varphi^{(k)}$, consisting of the state $\boldsymbol{s}^{(k)}$ and the previous H action–state pairs, i.e., $\varphi^{(k)} = [\boldsymbol{s}^{(k-H)}, \boldsymbol{a}^{(k-H)}, \cdots, \boldsymbol{s}^{(k-1)}, \boldsymbol{a}^{(k-1)}, \boldsymbol{s}^{(k)}]$. The experience sequence $\varphi^{(k)}$ is input to the E-network and the Q-network to estimate $E\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}\right)$ and $Q\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}\right)$, respectively. Then, the policy $\boldsymbol{a}^{(k)}$ is selected based on Equation (17).

The current state–action pair is fed into the E-network to obtain the network's estimate of the E-value. Then, the target E-value is calculated. The difference between the estimated E-value and the target E-value is computed, and this difference is used to update the weights $\mathcal{E}$ of the E-network. The loss function of the E-value $\mathcal{L}\left(\mathcal{E}^{(k)}\right)$ is defined as follows:

$$\mathcal{L}\left(\mathcal{E}^{(k)}\right) = \mathbb{E}_{\Upsilon^{(k)} \in \mathcal{M}}\left[\left(\sum_{i=0}^{\lambda} \gamma^i l\left(\boldsymbol{s}^{(k+i)}, \boldsymbol{a}^{(k+i)}\right) - E\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}\right)\right)^2\right] \tag{18}$$

During training, we use a stochastic gradient descent algorithm to update the weights of the convolutional neural network (CNN). The CNN evaluates the strategy as a Q-value so that the agent can choose the optimal action based on the current state. By minimizing the mean square error between the estimated network's output Q-value and the optimal target Q-value, the agent can update the Q-network weights Q and improve its performance in the environment, with the loss function $\mathcal{L}\left(\mathcal{Q}^{(k)}\right)$ given by

$$\mathcal{L}\left(\mathcal{Q}^{(k)}\right) = \mathbb{E}_{\mathrm{Y}^{(k)}\in\mathcal{M}}\left[\left(U^{(k)} + \gamma \max Q\left(\boldsymbol{s}^{(k+1)}, \boldsymbol{a}\right) - Q\left(\boldsymbol{s}^{(k)}, \boldsymbol{a}^{(k)}\right)\right)^2\right] \tag{19}$$

The process is repeated H times to update $\mathcal{Q}^{(k)}$ and $\mathcal{E}^{(k)}$. We also adopt the transfer learning technique to initialize the weights of the two deep CNNs to improve the training efficiency, and the random exploration is avoided at the beginning of learning. For the traditional DQN algorithm, the usual approach involves calculating the target Q-value and selecting the action with the highest Q-value as the current policy choice. However, in the SDLPTO algorithm, a risk assessment method is employed during action selection to avoid choosing high-risk actions. The detailed safe DQN-based location-privacy-preserved task offloading is described as Algorithm 1.

Algorithm 1 Safe DQN-based LPTO (SDLPTO)

1: Initialize the real distance, ω, γ, α, $\mathcal{Q}$ and $\mathcal{E}$ according to transfer learning
2: **for** $k = 1, 2, 3, \cdots$ **do**
3: Observe the system state $\boldsymbol{s}^{(k)} = [V^{(k)}, R^{(k-1)}]$
4: Input the experience sequence $\varphi^{(k)} = [\boldsymbol{s}^{(k-H)}, \boldsymbol{a}^{(k-H)}, \cdots, \boldsymbol{s}^{(k-1)}, \boldsymbol{a}^{(k-1)}, \boldsymbol{s}^{(k)}]$ to the Q-network and E-network to estimate the Q-values and E-values
5: Select $\boldsymbol{a}^{(k)} = [\chi^{(k)}, \varepsilon^{(k)}]$ based on the offloading policy and location perturbation policy obtained from the network
6: Obtain d^* based on (8) at current privacy budget $\varepsilon^{(k)}$
7: Obtain $R^{(k)}$ based on (3) at current perturbation distance d^*
8: Calculate the average cost $C^{(k)}$ and privacy leakage $PL^{(k)}$ to obtain the utility $U^{(k)}$
9: Update the weights of the CNNs for $\mathcal{Q}^{(k)}$ and $\mathcal{E}^{(k)}$ by applying minibatch updates via (18) and (19)
10: **end for**

5. Simulation Setup and Results

In this section, we evaluate the performance advantage of our proposed scheme through simulation experiments. In the context of task offloading in edge computing, we assume that there is one user and one edge server. The coverage range of the MEC server is 500 m [7]. The mobile user is randomly distributed within this area, and the path loss exponent is assumed to be $\vartheta = 0.2$. All the experiments are implemented by Python 3.8 and on the same machine, with 16 GB RAM and an Intel(R) Core(TM) i5-12500 processor.

The learning rate of the agent is set to 0.004, the discount factor is set to 0.99, and we train the agent for 4000 time slots. Each time slot has a duration of 1 s, similar to [35]. Our research team determined these parameters by conducting multiple experiments, enabling us to fine-tune the parameters and achieve optimal simulation performance. The rest of the parameter settings for the experiments are shown in Table 1. Adjusting the system environment parameters might impact the numerical results; they do not alter our approach's overall trends and advantages.

When the privacy parameter ω set by the user is larger, the level of privacy leakage is lower, and the user tends to use a larger privacy parameter to protect location privacy. However, on the other hand, the distance between the user and the server may increase, which means that the average cost incurred by the user will be higher.

Table 1. Experimental parameters.

Parameter	Definition	Value
α	Learning rate	0.004
δ	Discount factor	0.99
H	Batch size	32
B_0	Channel bandwidth	10 MHz
N_0	Background noise	30 dBm
f_l	User CPU frequency	1 GHz
f_s	Server CPU frequency	10 GHz
p	Transmission power	20 mW
ϕ	Calculate density	10 cycles/bit
v	Task data size	50∼300 Kbit

As shown in Figure 4a, the computational cost also increases with the increase in the parameter ω. When the value of ω is high, a larger perturbation range is required to perturb the user's location to protect location privacy. Because the perturbed position data may have a certain deviation from the actual distance between the user and the server, which might be greater than the real distance, more computation is needed when offloading tasks, thus increasing the computational cost. As the parameter ω increases, the level of privacy leakage of the user's location will decrease. The parameter ω balances the trade-off between the user's privacy leakage level and the computational cost, reflecting the user's concern about location privacy protection. A higher ω value indicates that the user pays more attention to location privacy protection and requires a larger disturbance range to disturb the user's location. When the disturbance area becomes larger, the distance difference between the disturbed pseudo-location and the actual location may increase, making it difficult for attackers to infer the user's actual location, thus protecting the user's location privacy.

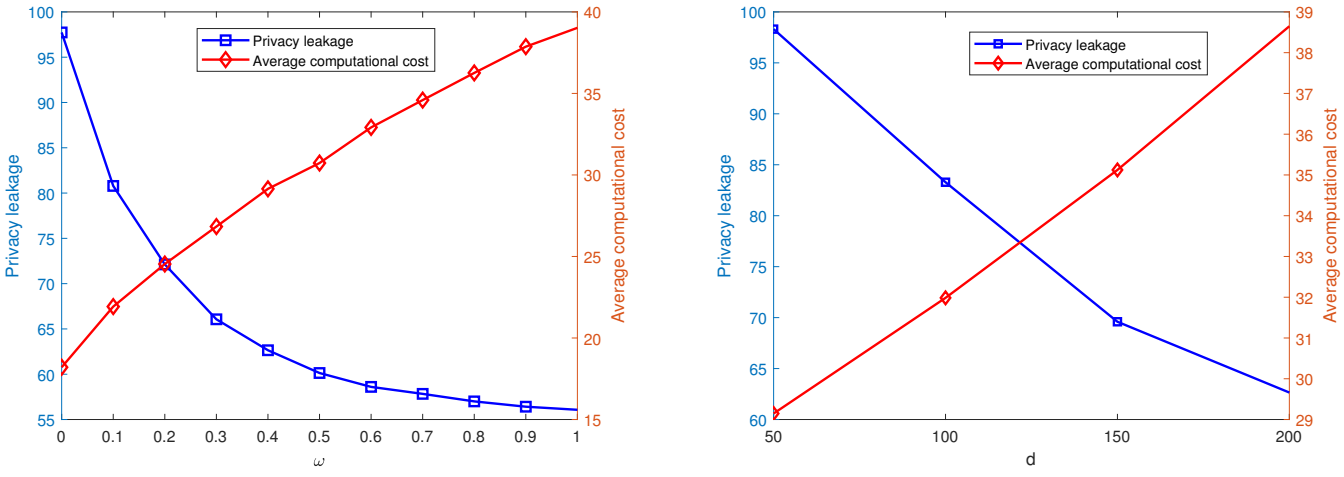

Figure 4. The relationship between ω, d, and the cost and privacy leakage. (**a**) Evaluation of parameter ω. (**b**) Evaluation of parameter d.

Figure 4b shows that as the parameter distance increases, the computational cost also increases. As the distance gradually increases, the range of the perturbation region will become larger, and the probability of the user's perturbed location being further from the true location will become larger to protect location privacy. When the distance between the user and the server after perturbation becomes greater, the state of the wireless channel may worsen, the user may perform more tasks locally, and the offloading strategy may not be optimal, thus increasing the computational cost. Furthermore, it shows that as the

parameter distance increases, the level of location privacy leakage of the user will decrease. As the distance gradually increases, the range of the perturbation region will become larger. The attacker needs to search within a broader area to determine the user's real location, which greatly increases the difficulty and probability for the attacker to find the user's real location. In this way, the user's location privacy is protected.

Figures 5 and 6 illustrate the performance of the proposed mechanism versus time. From the figure, it can be seen that our proposed SDLPTO mechanism outperforms the No DP and DPRL mechanisms by reducing the privacy leakage level of SDLPTO by 18.2% and 11.2% at time slot 2000, reducing the computational cost by 33.1% and 35.2%, and improving the utility by 33% and 27.2%, respectively. This is because No DP does not consider location privacy, and DPRL only implements location privacy protection and offloading optimization separately; our proposed method jointly optimizes user privacy and computational offloading cost, effectively improving the overall user benefit. Moreover, compared with LPTO, SDLPTO reduces the privacy leakage level and computational cost by 7.7% and 26.7%, respectively, and improves the benefits by 9.1%. This is because safe exploration can avoid selecting operations with higher risk levels, thus reducing privacy leakage and computational cost.

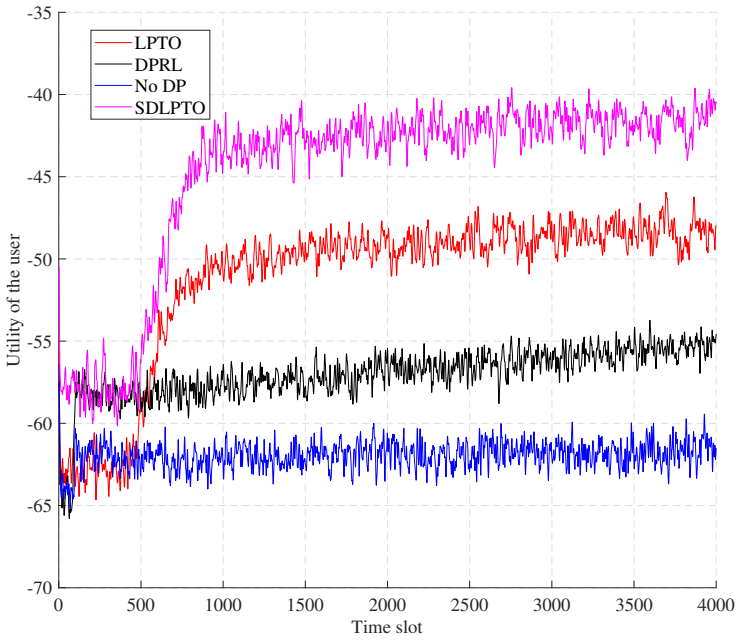

Figure 5. Utility of the four schemes versus time.

Figure 7a,b illustrate the relationship between average computational cost, privacy leakage, and task size for the four mechanisms. As the task volume increases from 50 Kbit to 300 Kbit [35], SDLPTO exhibits a 6.0% increase in privacy leakage level and a 5.2-times increase in computation cost. This indicates that as the task scale grows, more computational resources are required to execute these tasks, leading to a significant rise in computation cost. Moreover, as the task volume increases, users tend to offload more tasks to edge servers, which entails greater collection and processing of location information data, potentially increasing the risk of location privacy leakage.

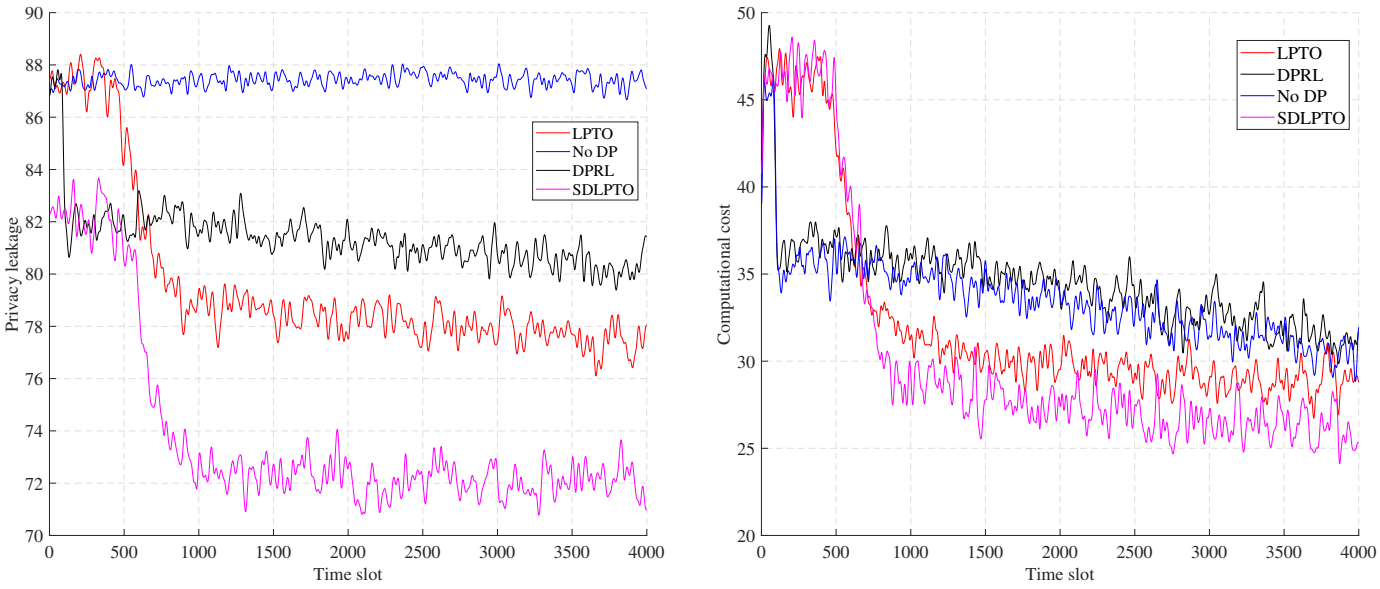

Figure 6. Computational cost and privacy leakage level versus time. (**a**) Privacy leakage level. (**b**) Computational cost.

In contrast, the No DP and DPRL mechanisms are only affected in terms of computational cost by increased task volume, while their privacy leakage level remains consistently high. When the task volume reaches 300, SDLPTO demonstrates a reduction of 3.6% in privacy leakage level and a decrease of 11.2% in computational cost compared to LPTO. This indicates that even with increased task volume, our proposed approach can still effectively balance and optimize the trade-off between user privacy requirements and computational cost, reducing the privacy leakage level and average computational cost.

Figure 7c,d illustrate the average performance of the four mechanisms at different range sizes. As the perturbation range increases, there is a possibility of perturbing the user's real location to a more distant position. With a greater perturbation distance, according to Equation (3), the signal undergoes attenuation, interference, and other effects during transmission, resulting in degraded channel conditions and an increase in average computational cost. For example, when the perturbation range increases from 50 to 250, both LPTO and SDLPTO experience a respective increase of 39.8% and 40.5% in average computational cost.

Simultaneously, as the range increases, the privacy leakage level of users decreases. However, the No DP mechanism does not consider location privacy protection, leading to consistently high privacy leakage levels. Despite the increase in range resulting in increased costs, our proposed mechanism still outperforms the others. For example, at a perturbation range of 250, SDLPTO achieves a 15.2% reduction in privacy leakage level and a 5.9% decrease in average cost compared to DPRL.

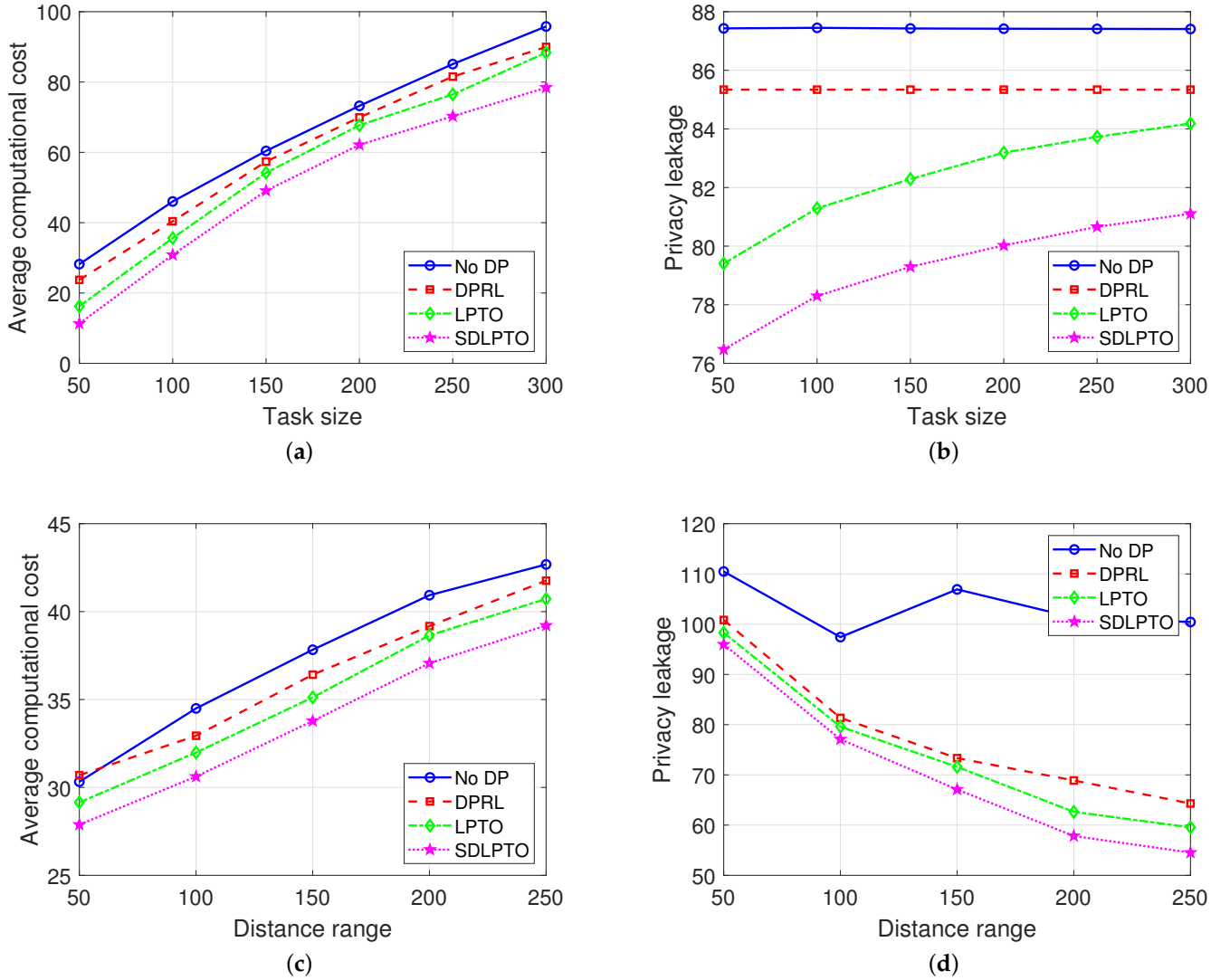

Figure 7. Evaluation of average cost and average privacy leakage. (**a**) Impact of task size on average cost. (**b**) Impact of task size on privacy leakage. (**c**) Impact of range on average cost. (**d**) Impact of range on privacy leakage.

6. Conclusions

We have investigated a safe-learning-based location-privacy-preserved task offloading scheme, SDLPTO, which effectively addresses the challenge of balancing computational cost and location privacy protection in dynamic MEC systems. This paper has the following three main contributions. First, we proposed a location-privacy-aware task offloading framework that uses differential privacy technology to protect users' location privacy. This framework can jointly consider the location privacy and computational offloading performance, enabling a balance between them. Second, we applied a DRL technique to dynamically optimize the offloading and perturbation policies in dynamic MEC by trial and error, ensuring optimal offloading performance while preserving privacy. Third, we implemented a safe exploration algorithm for location perturbation and offloading decisions. This algorithm effectively mitigates potential losses associated with high-risk state–action pairs, enhancing the overall privacy and offloading performance. Experiments were conducted to demonstrate the advantage of the proposed scheme over the typical no-privacy-preserving scheme and DPRL scheme, as it significantly enhances location privacy while reducing computational costs.

Author Contributions: Conceptualization, M.M. and P.Z.; investigation, M.M. and Z.L.; methodology, M.M. and Z.L.; project administration, S.L.; software, Z.L. and J.D.; supervision, M.M.; validation, P.Z.; writing—original draft, Z.L. and J.D.; writing—review and editing, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number 62101557 and 62371451), and Xuzhou Basic Research Plan Project-Young Scientific and Technological Talent Project (KC23022), and China Postdoctoral Science Foundation (2022M713378), and the Fundamental Research Funds for the Central Universities (2042022kf0021).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, K.; Wang, L.; Pan, C.; Ren, H. Deep reinforcement learning-based resource management for flexible mobile edge computing: Architectures, applications, and research issues. *IEEE Veh. Technol. Mag.* **2022**, *17*, 85–93. [CrossRef]
2. Chen, N.; Qiu, T.; Zhao, L.; Zhou, X.; Ning, H. Edge intelligent networking optimization for internet of things in smart city. *IEEE Wirel. Commun.* **2021**, *28*, 26–31. [CrossRef]
3. Tian, Z.; Wang, Y.; Sun, Y.; Qiu, J. Location privacy challenges in mobile edge computing: Classification and exploration. *IEEE Netw.* **2020**, *34*, 52–56. [CrossRef]
4. Rao, F.Y.; Bertino, E. Privacy techniques for edge computing systems. *Proc. IEEE* **2019**, *107*, 1632–1654. [CrossRef]
5. Cui, G.; He, Q.; Chen, F.; Jin, H.; Xiang, Y.; Yang, Y. Location privacy protection via delocalization in 5G mobile edge computing environment. *IEEE Trans. Serv. Comput.* **2021**, *16*, 412–423. [CrossRef]
6. Zhou, L.; Yu, L.; Du, S.; Zhu, H.; Chen, C. Achieving differentially private location privacy in edge-assistant connected vehicles. *IEEE Internet Things J.* **2018**, *6*, 4472–4481. [CrossRef]
7. Liu, Z.; Wang, J.; Gao, Z.; Wei, J. Privacy-preserving edge computing offloading scheme based on whale optimization algorithm. *J. Supercomput.* **2023**, *79*, 3005–3023. [CrossRef]
8. Shaham, S.; Ding, M.; Liu, B.; Dang, S.; Lin, Z.; Li, J. Privacy preserving location data publishing: A machine learning approach. *IEEE Trans. Knowl. Data Eng.* **2020**, *33*, 3270–3283. [CrossRef]
9. Jiang, B.; Li, J.; Wang, H.; Song, H. Privacy-preserving federated learning for industrial edge computing via hybrid differential privacy and adaptive compression. *IEEE Trans. Ind. Inform.* **2021**, *19*, 1136–1144. [CrossRef]
10. Sun, G.; Cai, S.; Yu, H.; Maharjan, S.; Chang, V.; Du, X.; Guizani, M. Location privacy preservation for mobile users in location-based services. *IEEE Access* **2019**, *7*, 87425–87438. [CrossRef]
11. Liu, S.; Wang, J.H.; Wang, J.; Zhang, Q. Achieving user-defined location privacy preservation using a P2P system. *IEEE Access* **2020**, *8*, 45895–45912. [CrossRef]
12. Tong, W.; Tong, Y.; Xia, C.; Hua, J.; Li, Q.; Zhong, S. Understanding Location Privacy of the Point-of-Interest Aggregate Data via Practical Attacks and Defenses. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 2433–2449. [CrossRef]
13. Jing, W.; Miao, Q.; Song, H.; Chen, X. Data loss and reconstruction of location differential privacy protection based on edge computing. *IEEE Access* **2019**, *7*, 75890–75900. [CrossRef]
14. Zhang, P.; Gan, P.; Chang, L.; Wen, W.; Selvi, M.; Kibalya, G. DPRL: Task offloading strategy based on differential privacy and reinforcement learning in edge computing. *IEEE Access* **2022**, *10*, 54002–54011. [CrossRef]
15. He, Z.; Zhang, T.; Lee, R.B. Attacking and protecting data privacy in edge–cloud collaborative inference systems. *IEEE Internet Things J.* **2020**, *8*, 9706–9716. [CrossRef]
16. Sandal, Y.S.; Pusane, A.E.; Kurt, G.K.; Benedetto, F. Reputation based attacker identification policy for multi-access edge computing in internet of things. *IEEE Trans. Veh. Technol.* **2020**, *69*, 15346–15356. [CrossRef]
17. Min, M.; Wan, X.; Xiao, L.; Chen, Y.; Xia, M.; Wu, D.; Dai, H. Learning-based privacy-aware offloading for healthcare IoT with energy harvesting. *IEEE Internet Things J.* **2018**, *6*, 4307–4316. [CrossRef]
18. Wang, Z.; Sun, Y.; Liu, D.; Hu, J.; Pang, X.; Hu, Y.; Ren, K. Location Privacy-Aware Task Offloading in Mobile Edge Computing. *IEEE. Trans. Mob. Comput.* 2023, *accepted*. [CrossRef]
19. Lu, X.; Xiao, L.; Niu, G.; Ji, X.; Wang, Q. Safe exploration in wireless security: A safe reinforcement learning algorithm with hierarchical structure. *IEEE Trans. Inf. Forensic Secur.* **2022**, *17*, 732–743. [CrossRef]
20. He, C.; Liu, G.; Guo, S.; Yang, Y. Privacy-preserving and low-latency federated learning in edge computing. *IEEE Internet Things J.* **2022**, *9*, 20149–20159. [CrossRef]
21. Zhang, S.; Hu, B.; Liang, W.; Li, K.C.; Gupta, B.B. A Caching-based Dual K-Anonymous Location Privacy-Preserving Scheme for Edge Computing. *IEEE Internet Things J.* **2023**, *10*, 9768–9781. [CrossRef]
22. Wang, W.; Wang, Y.; Duan, P.; Liu, T.; Tong, X.; Cai, Z. A triple real-time trajectory privacy protection mechanism based on edge computing and blockchain in mobile crowdsourcing. *IEEE. Trans. Mob. Comput.* **2022**, *22*, 5625–5642. [CrossRef]
23. Primault, V.; Boutet, A.; Mokhtar, S.B.; Brunie, L. The long road to computational location privacy: A survey. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 2772–2793. [CrossRef]

24. Kim, J.W.; Edemacu, K.; Jang, B. Privacy-preserving mechanisms for location privacy in mobile crowdsensing: A survey. *J. Netw. Comput. Appl.* **2022**, *200*, 103315. [CrossRef]
25. Miao, Q.; Jing, W.; Song, H. Differential privacy–based location privacy enhancing in edge computing. *Concurr. Comput.* **2019**, *31*, e4735. [CrossRef]
26. Wei, P.; Guo, K.; Li, Y.; Wang, J.; Feng, W.; Jin, S.; Ge, N.; Liang, Y. Reinforcement learning-empowered mobile edge computing for 6G edge intelligence. *IEEE Access* **2022**, *10*, 65156–65192. [CrossRef]
27. Zhou, H.; Jiang, K.; Liu, X.; Li, X.; Leung, V.C. Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing. *IEEE Internet Things J.* **2021**, *9*, 1517–1530. [CrossRef]
28. Nguyen, D.C.; Pathirana, P.N.; Ding, M.; Seneviratne, A. Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2536–2549. [CrossRef]
29. Liu, S.; Zheng, C.; Huang, Y.; Quek, T.Q. Distributed reinforcement learning for privacy-preserving dynamic edge caching. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 749–760. [CrossRef]
30. Zhang, P.; Min, M.; Xiao, J.; Li, S.; Zhang, H. IRS-Aided Mobile Edge Computing for Mine IoT Networks using Deep Reinforcement Learning. In Proceedings of the 2023 IEEE/CIC International Conference on Communications in China, Dalian, China, 10–12 August 2023 ; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.
31. Wu, Q.; Wang, S.; Ge, H.; Fan, P.; Fan, Q.; Letaief, K.B. Delay-sensitive Task Offloading in Vehicular Fog Computing-Assisted Platoons. *IEEE Trans. Netw. Serv. Manag.* 2023, *accepted*. [CrossRef]
32. Van Erven, T.; Harremos, P. Rényi divergence and Kullback-Leibler divergence. *IEEE Trans. Inf. Theory* **2014**, *60*, 3797–3820. [CrossRef]
33. Wu, Q.; Shuai, S.; Ziyang, W.; Qiang, F.; Pingyi, F.; Cui, Z. Towards V2I age-aware fairness access: A DQN based intelligent vehicular node training and test method. *Chin. J. Electron.* **2023**, *32*, 1230–1244.
34. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
35. Zhang, G.; Ni, S.; Zhao, P. Learning-based joint optimization of energy delay and privacy in multiple-user edge-cloud collaboration MEC systems. *IEEE Internet Things J.* **2021**, *9*, 1491–1502. [CrossRef]