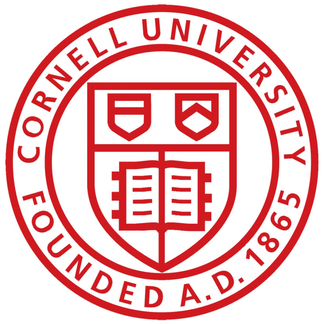
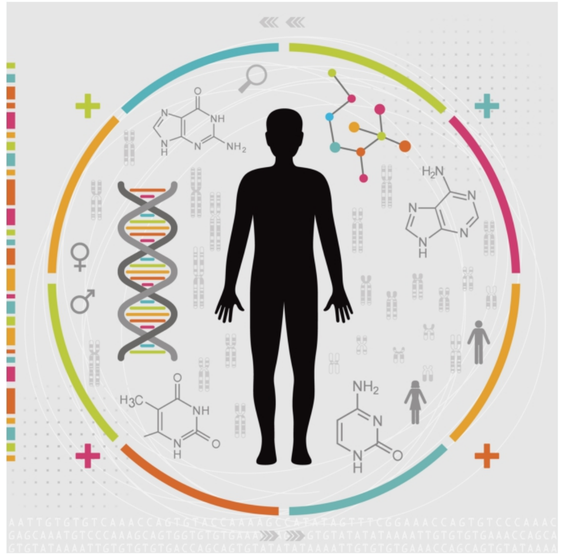
Towards Optimal Differentially Private Regret Bounds in Linear MDPs

Sharan Sahu | Stats and Data Sci. PhD | Cornell University

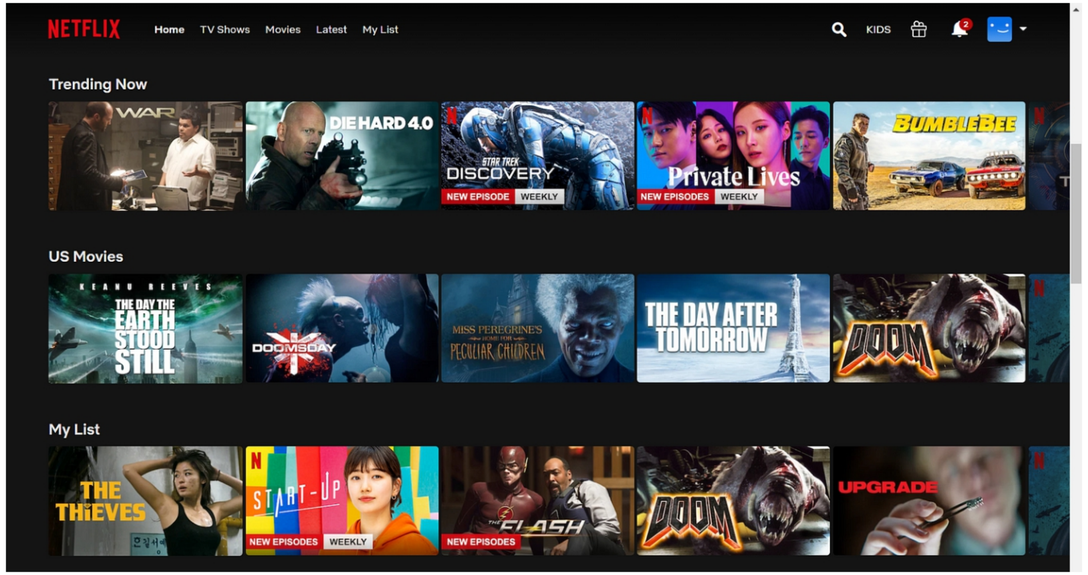


Cornell University®

Recent Successes of Reinforcement Learning (RL)



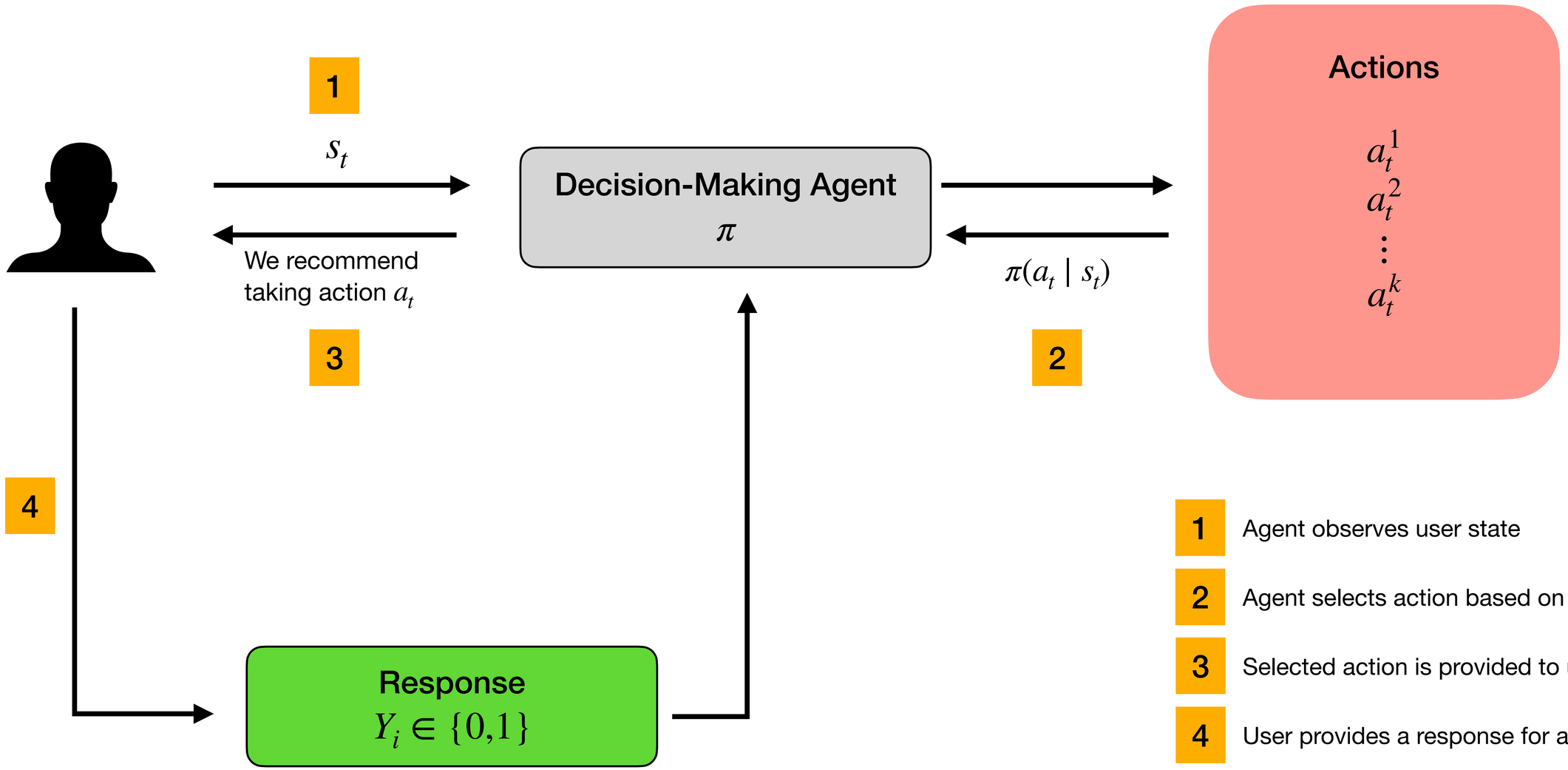
Precision Medicine



Recommender Systems

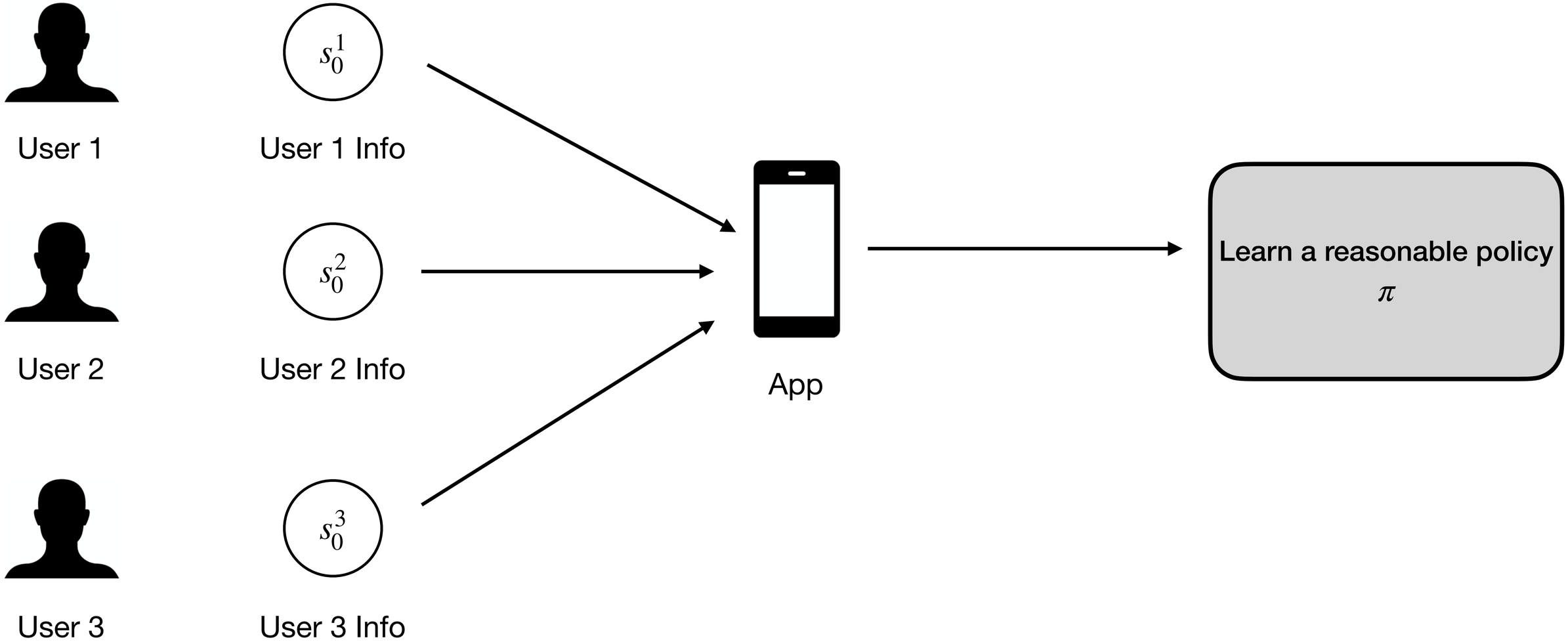
Autonomous Driving

Contextual Bandits

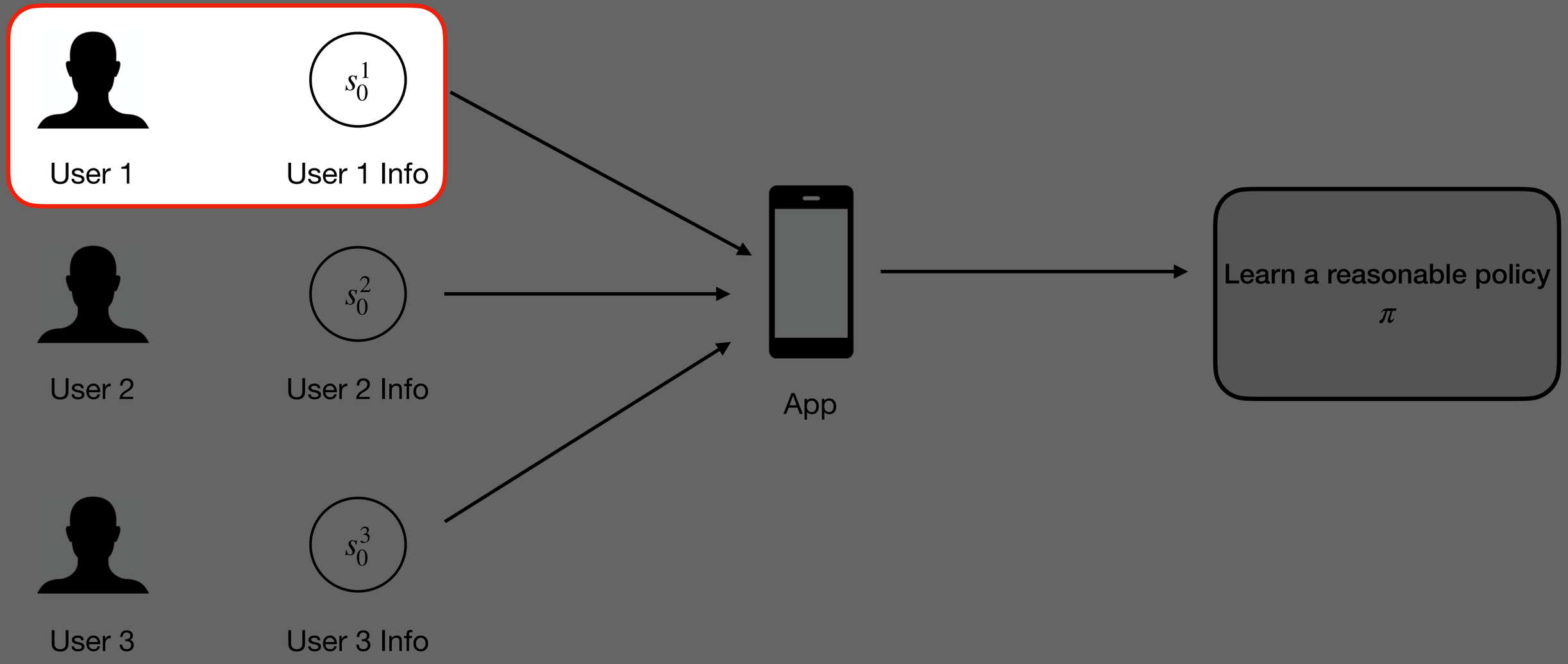


1. Agent observes user state
2. Agent selects action based on user state
3. Selected action is provided to user
4. User provides a response for agent to learn

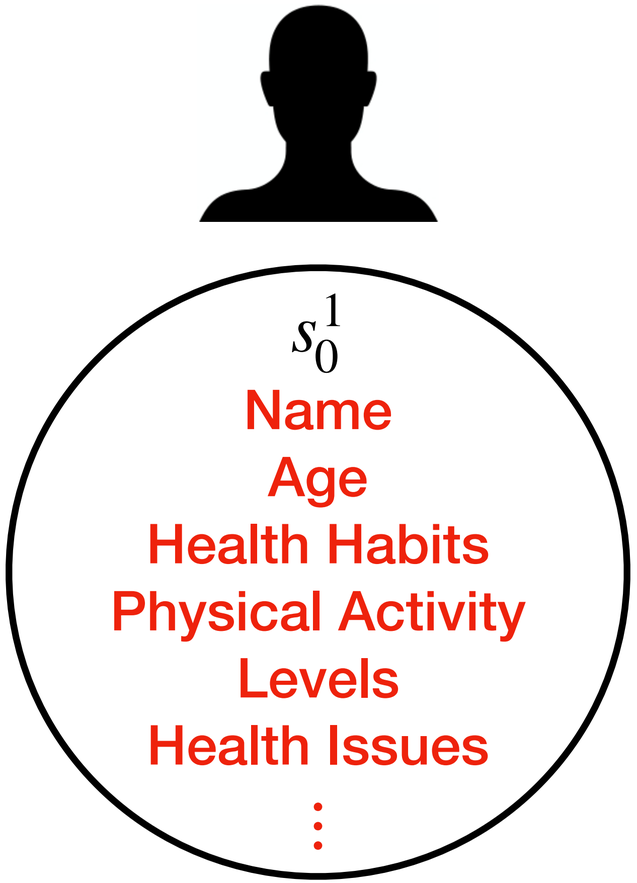
Reasonable Policies May Use Sensitive Data



Reasonable Policies May Use Sensitive Data



Reasonable Policies May Use Sensitive Data



The policy has **access to information** that users may consider **sensitive** or **private**

Neural Networks Can Memorize Personal Information From One Example

Anonymisation Fails
Single sample with personal features

DNN trained for
X-ray classification

Inserting the (memorised) unique feature
changes prediction

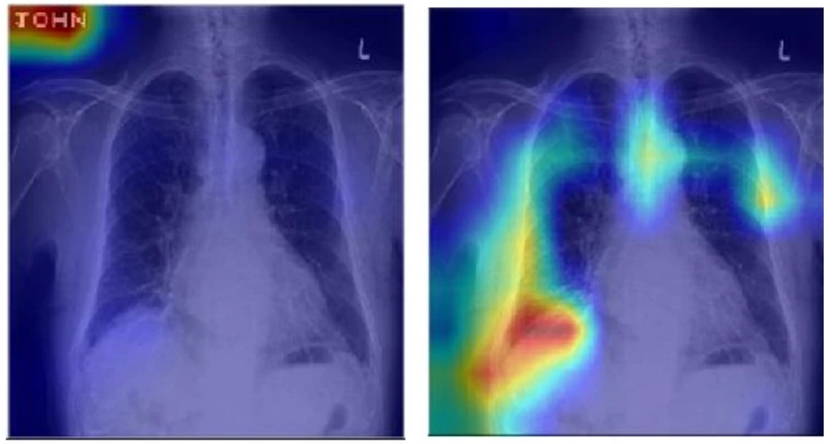


Diagram Credit: Neural networks Memorize Personal Information From One Sample by Hartley et al. 2023

We Must Incorporate Privacy-Preserving Mechanisms Into RL

We require a mathematically rigorous framework that provides statistical guarantees for our (possibly randomized) mechanism

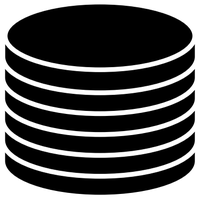
Definition (Approximate Differential Privacy). A mechanism $\mathscr{M}$ is (ε, δ)-DP if for all neighboring datasets $\mathscr{U}$, $\mathscr{U}'$ that differ by one record and for all events $\mathscr{E}$ in the output range

$$\mathbb{P}\left(\mathscr{M}\left(\mathscr{U}\right) \in \mathscr{E}\right) \leq e^{\varepsilon}\mathbb{P}\left(\mathscr{M}\left(\mathscr{U}'\right) \in \mathscr{E}\right) + \delta$$

Remark: This is a relaxation of ε-DP as in many settings, achieving ε-DP is near impossible or comes at high utility cost.

Differential Privacy (DP)

Database $\mathscr{D}_1$



Mechanism $\mathscr{M}$

$\mathscr{M}\left(\mathscr{D}_1\right)$

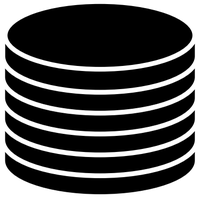
$+$

Some other people's data

$\approx$

$=$

Database $\mathscr{D}_2$



Mechanism $\mathscr{M}$

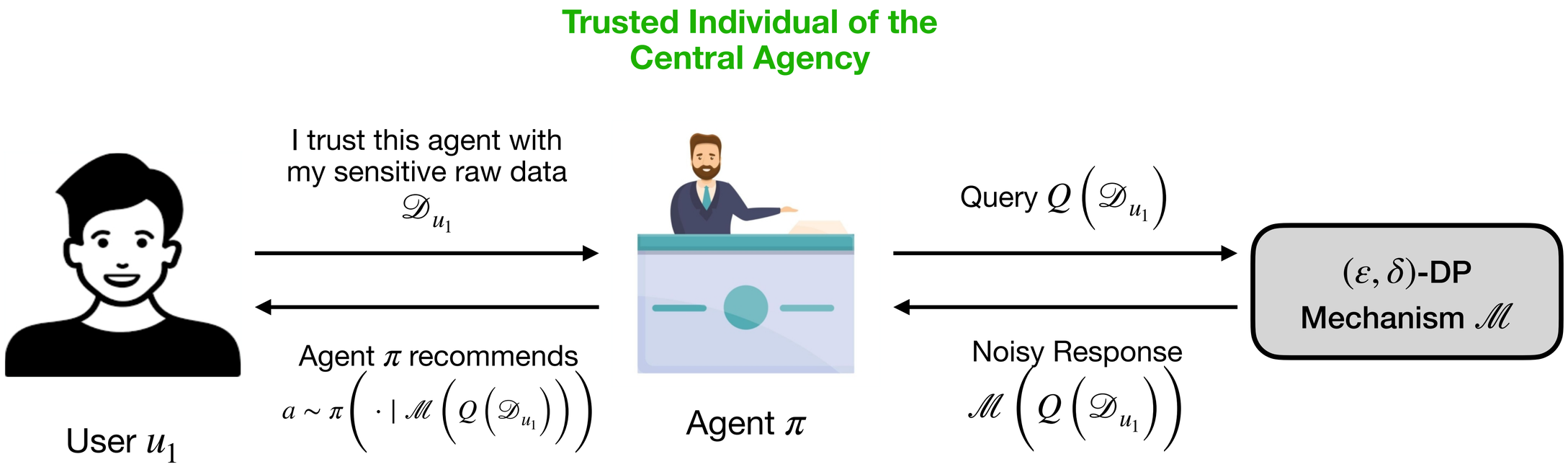
$\mathscr{M}\left(\mathscr{D}_2\right)$

Mechanism $\mathscr{M}$ is differentially private if …

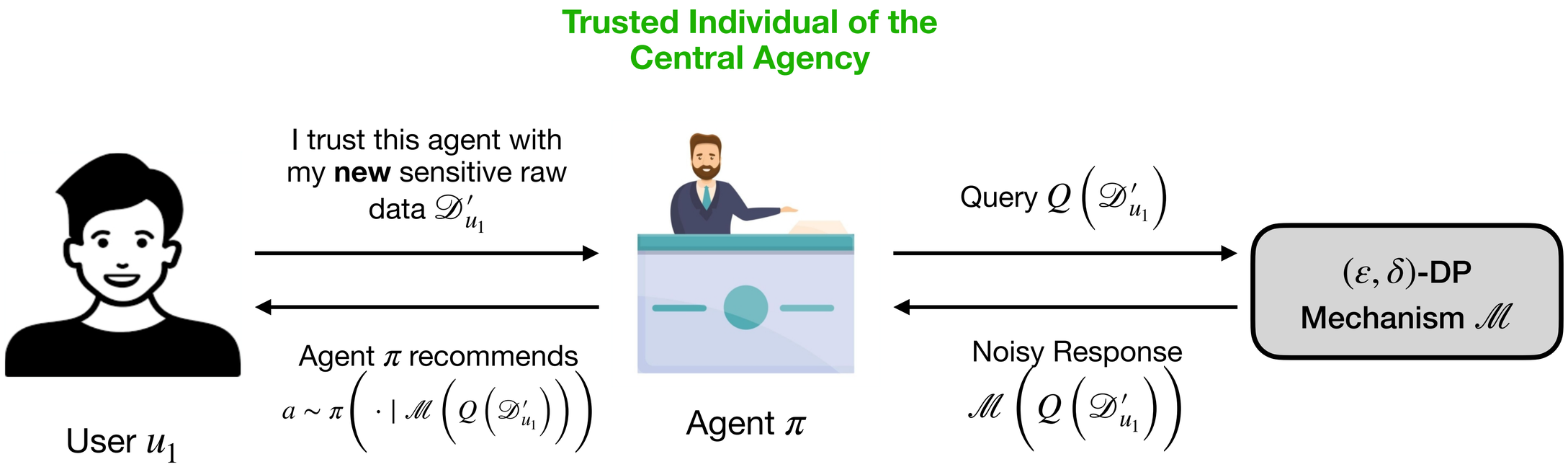
$\forall \mathscr{D}_1, \mathscr{D}_2$ that differ by at-most one record

$\mathscr{M}\left(\mathscr{D}_1\right), \mathscr{M}\left(\mathscr{D}_2\right)$ are indistinguishable

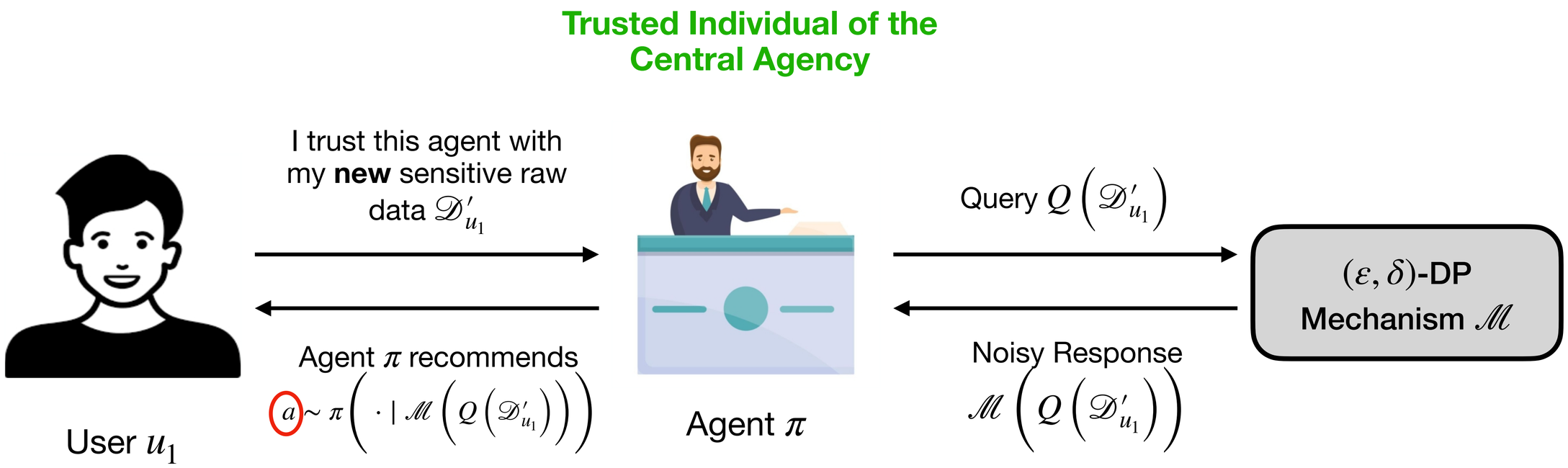
There are a few issues with (ε, δ)-DP



There are a few issues with (ε, δ)-DP

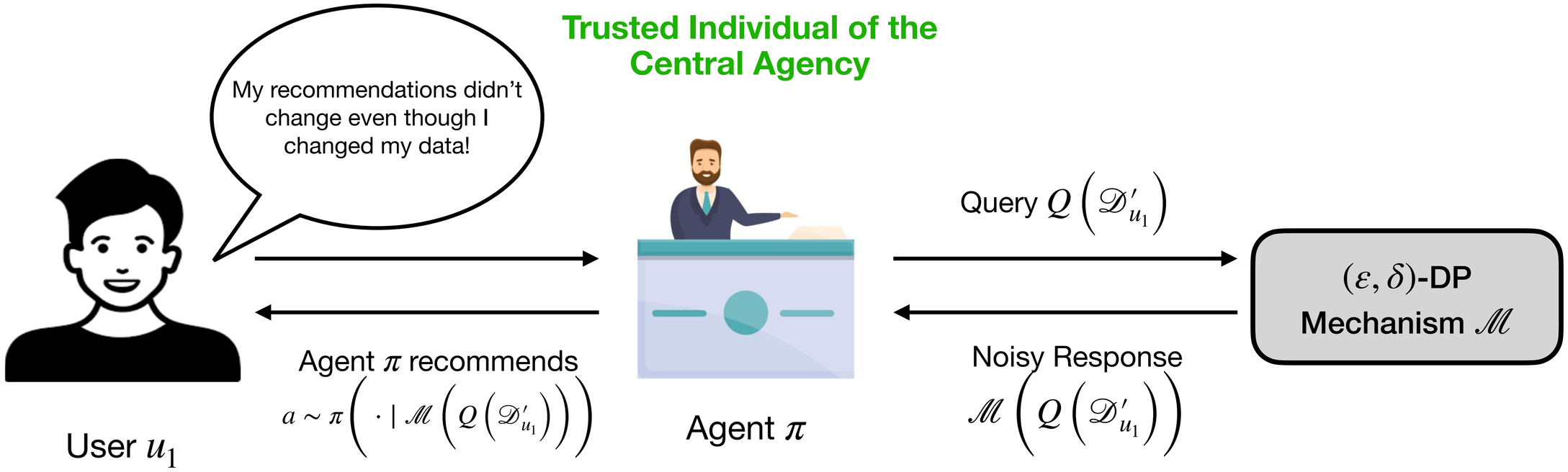


There are a few issues with (ε, δ)-DP

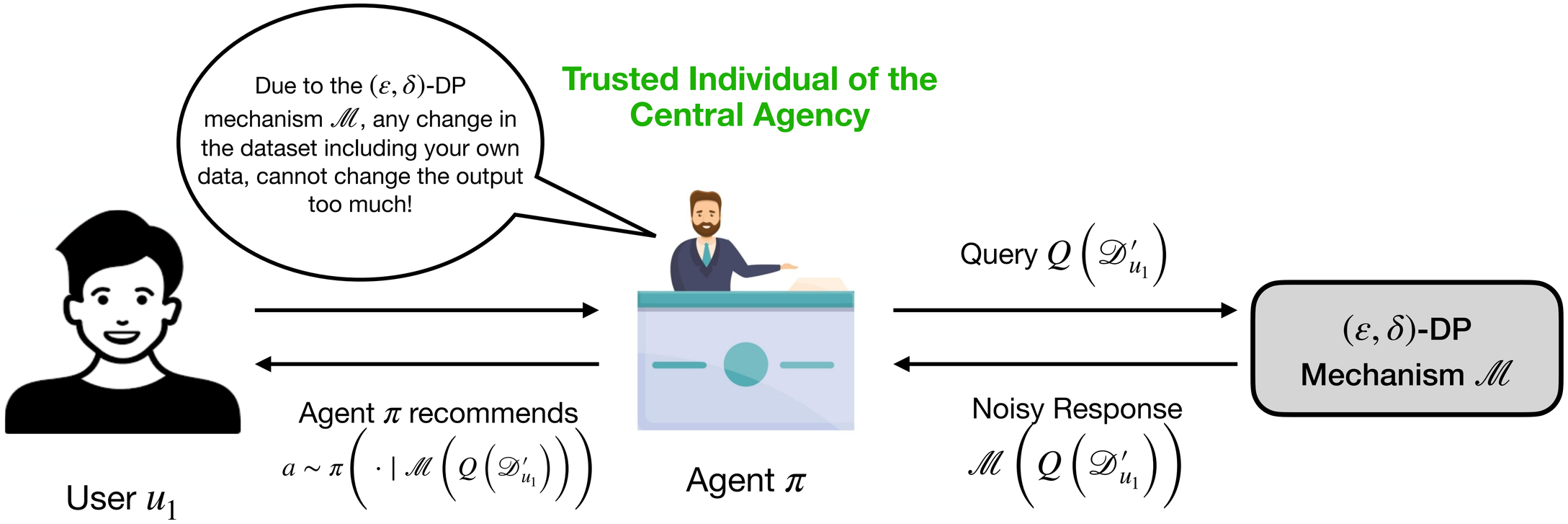


This is the same exact action recommended with the old data!

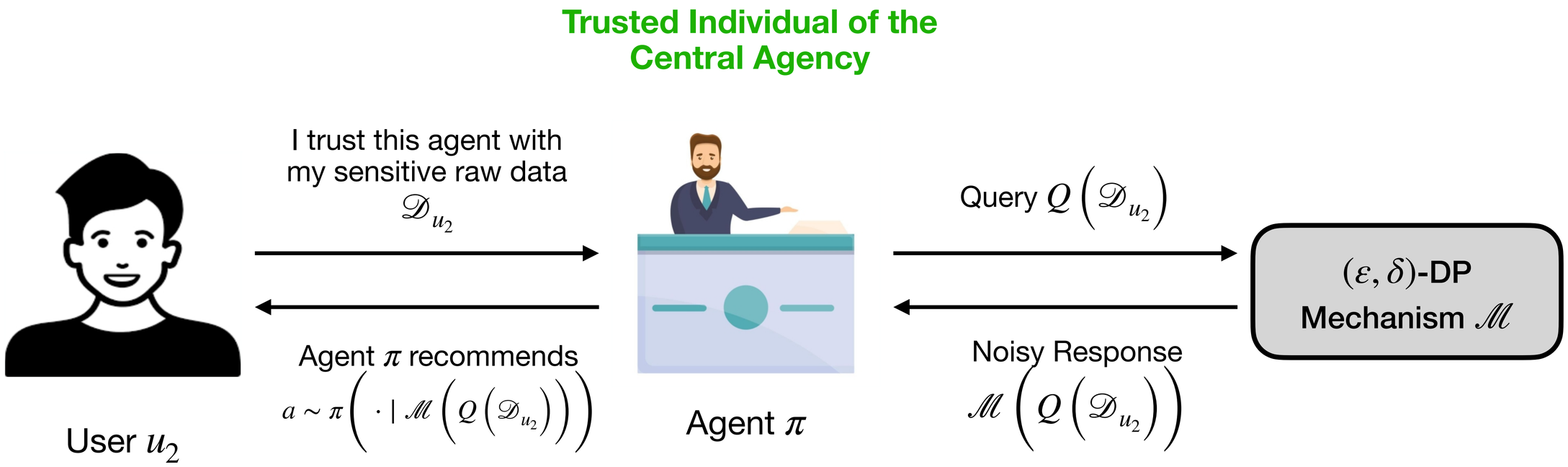
There are a few issues with (ε, δ)-DP



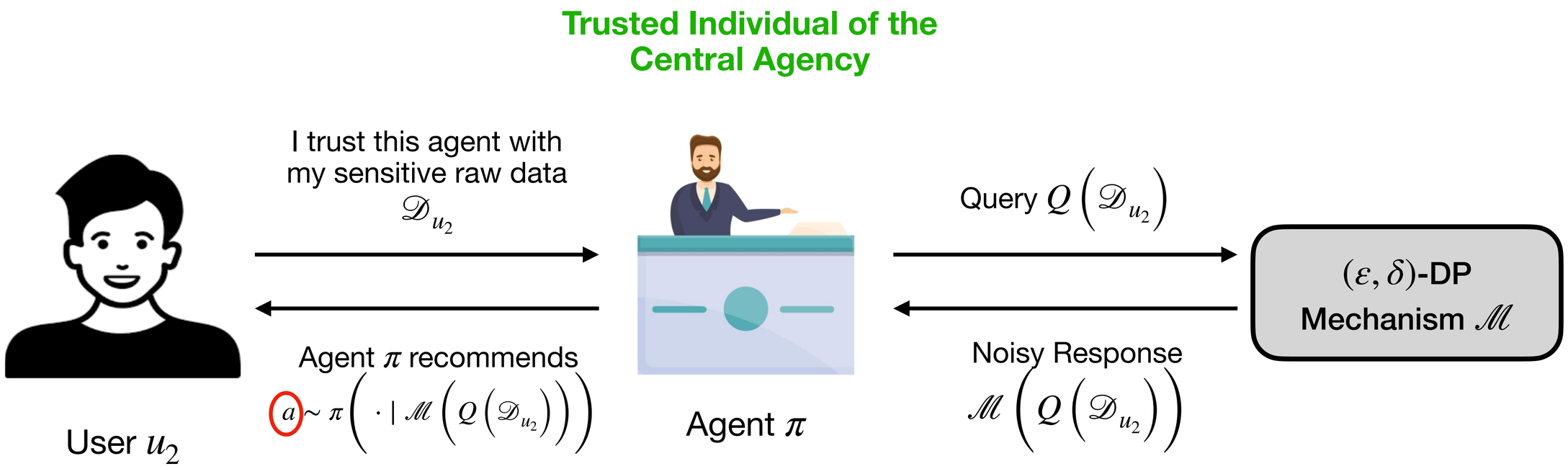
There are a few issues with (ε, δ)-DP



There are a few issues with (ε, δ)-DP

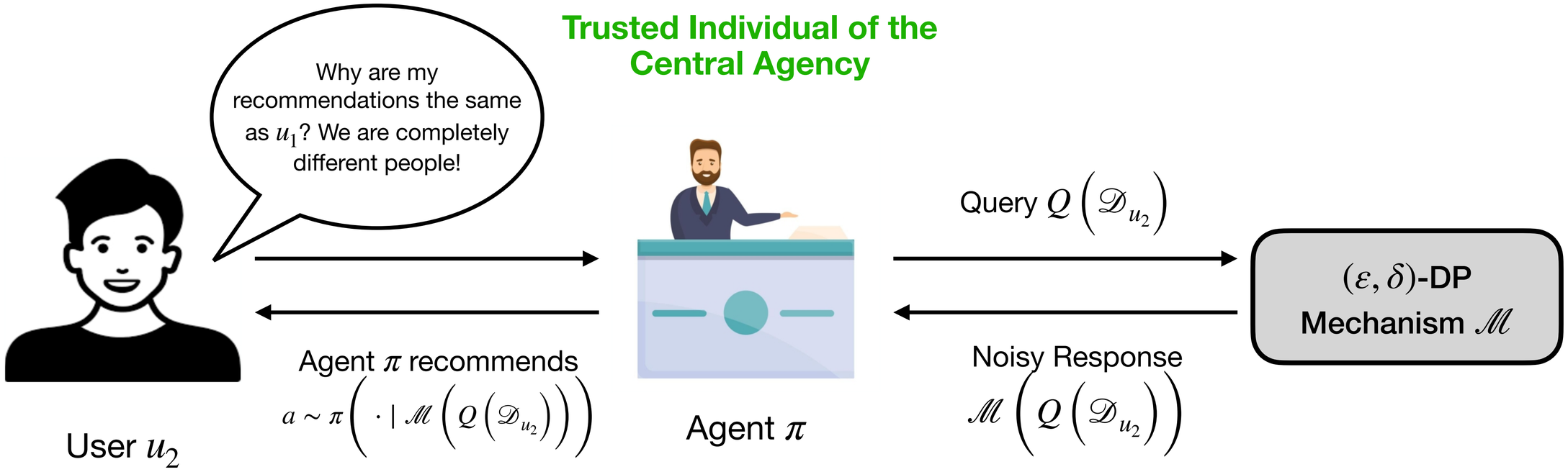


There are a few issues with (ε, δ)-DP

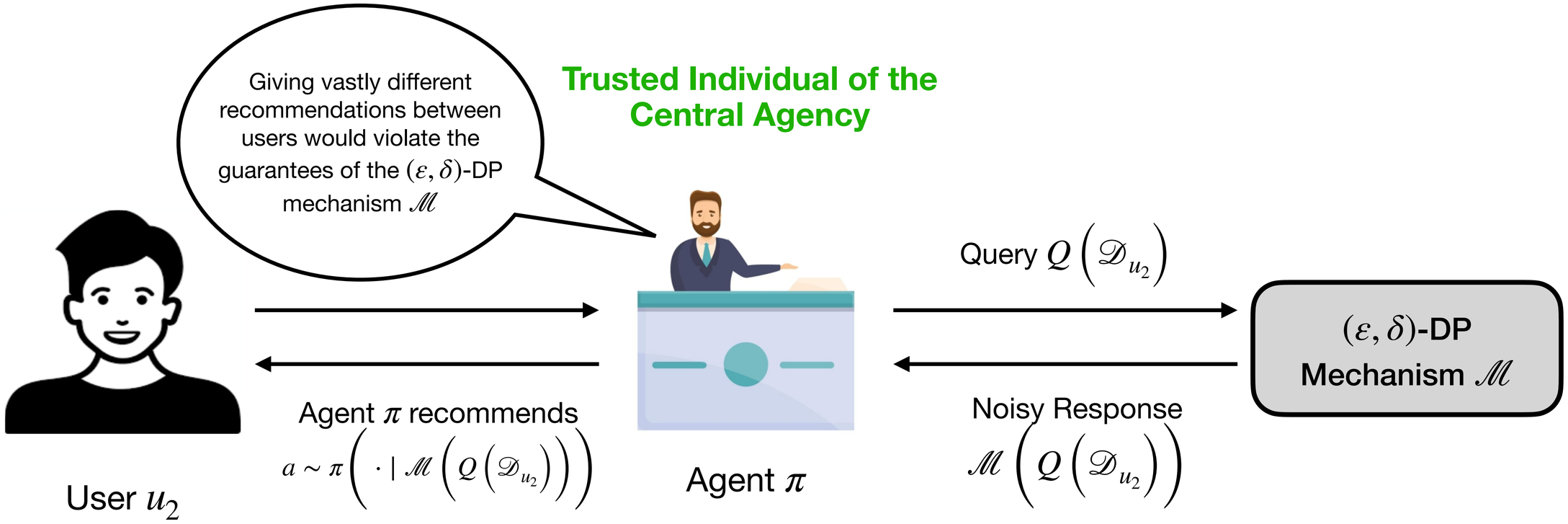


This is the same exact action recommended to the other user!

There are a few issues with (ε, δ)-DP



There are a few issues with (ε, δ)-DP



We need a further relaxation of DP …

One that works nicely with contextual bandit problems on a per-user level but does not sacrifice privacy on a per-decision or per-context level

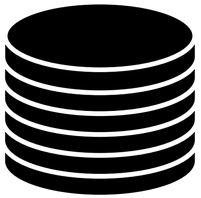
Definition (Approximate Joint Differential Privacy). A mechanism $\mathscr{M}$ is (ε, δ)-JDP if for any $k \in [K]$, any user sequences $\mathscr{U}$, $\mathscr{U}'$ differing on the k-user and any $\mathscr{E} \subset \mathscr{A}^{(K-1)H}$

$$\mathbb{P}\left(\mathscr{M}_{-k}(\mathscr{U}) \in \mathscr{E}\right) \leq e^{\varepsilon}\mathbb{P}\left(\mathscr{M}_{-k}(\mathscr{U}') \in \mathscr{E}\right) + \delta$$

Remark: JDP allows for **better utility** than standard DP in some contexts since it permits individual outputs to depend more heavily on the individual's own data

Joint Differential Privacy (JDP)

Database $\mathscr{D}_1$

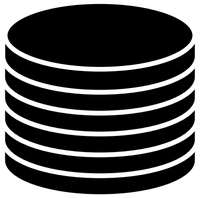


Mechanism $\mathscr{M}$

$\mathscr{M}_{-1}\left(\mathscr{D}_1\right)$

$\approx$

Database $\mathscr{D}_1'$



Mechanism $\mathscr{M}$

$\mathscr{M}_{-1}\left(\mathscr{D}_1'\right)$

Mechanism $\mathscr{M}$ is joint differentially private if …

$\forall \mathscr{D}_1, \mathscr{D}_1'$ where only the data only differs by at most party one's data

$\mathscr{M}_{-1}\left(\mathscr{D}_1\right), \mathscr{M}_{-1}\left(\mathscr{D}_1'\right)$ are indistinguishable

In this talk:

Can we develop an efficient (ε, δ)-JDP algorithm for sequential decision-making problems with **linear parametric representations**, and provide a novel algorithm with provably efficient guarantees for **privacy-preserving exploration**?

Outline

1. Problem Setup & Previous Work and Motivation

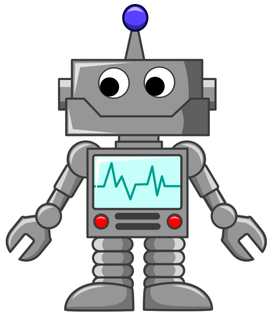
2. Can we do better?

3. Our regret bound with proof sketch

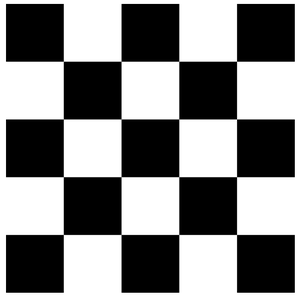
Episodic Time-Inhomogeneous Finite-Horizon MDPs

Policy: state to action

$$\pi_h^k(s) \to a$$



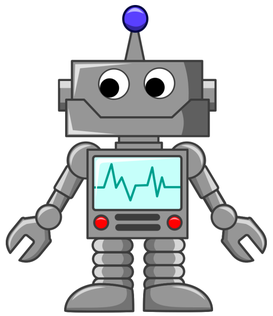
H times



Reward & Next State

$$r_h^k(s,a), \;\; s' \sim P_h^k\left(\cdot \mid s,a\right)$$

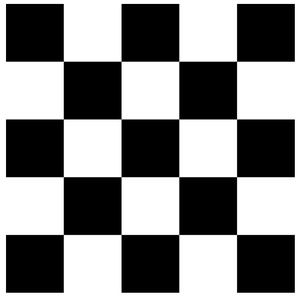
Episodic Time-Inhomogeneous Finite-Horizon MDPs



Policy: state to action

$\pi_h^k(s) \to a$

$\forall k \in [K]$,
play H times

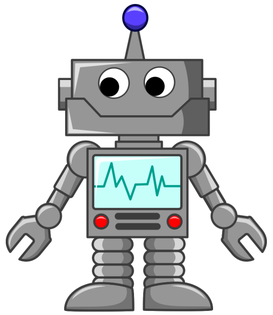


Reward & Next State

$r_h^k(s,a),\ \ s' \sim P_h^k\left(\cdot \mid s,a\right)$

Episodic Time-Inhomogeneous Finite-Horizon MDPs

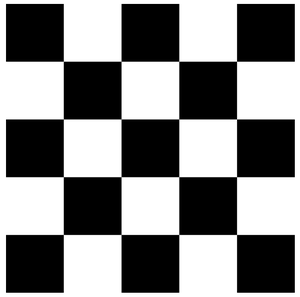
$$\tau^k = \left\{ s_h^k, a_h^k \right\}_{h=1}^{H}$$



Policy: state to action

$$\pi_h^k(s) \to a$$

$\forall k \in [K]$,
play H times

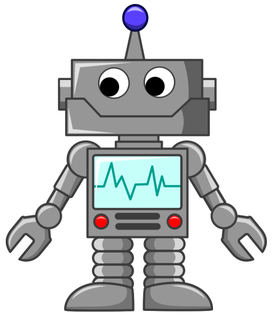


Reward & Next State

$$r_h^k(s,a), \;\; s' \sim P_h^k\left(\cdot \mid s, a \right)$$

Episodic Time-Inhomogeneous Finite-Horizon MDPs

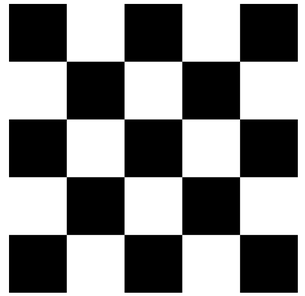
$$\tau^k = \left\{ s_h^k, a_h^k \right\}_{h=1}^{H}$$



Policy: state to action

$$\pi_h^k(s) \to a$$

$$\forall k \in [K],$$
play H times



Reward & Next State

$$r_h^k(s,a), \;\; s' \sim P_h^k\left(\cdot \mid s, a \right)$$

Finite-Horizon MDP: $\mathscr{M} = \left\{ \mathcal{S}, \mathcal{A}, \left\{ r_h \right\}_{h=1}^{H}, \left\{ \mathscr{P}_h \right\}_{h}^{H}, H \right\}, \;\; H < \infty$

Formal Reinforcement Learning Problem Setting

Let $\mathscr{M} = \left\{ \mathcal{S}, \mathcal{A}, \left\{ r_h \right\}_{h=1}^{H}, \left\{ \mathcal{P}_h \right\}_{h}^{H}, H \right\}$ be an episodic inhomogeneous finite-horizon Markov Decision Process (MDP) where where $\mathcal{S}, \mathcal{A}$ are the states and actions, respectively, and $H \in \mathbb{Z}$ is the length of each episode. We call $\mathcal{P}_h : \mathcal{S} \times \mathcal{A} \to \Delta\left(\mathcal{S}\right)$ the state-transition probability and $r_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function.

Formal Reinforcement Learning Problem Setting

Let $\mathscr{M} = \left\{ \mathcal{S}, \mathcal{A}, \{r_h\}_{h=1}^{H}, \{\mathcal{P}_h\}_{h}^{H}, H \right\}$ be an episodic inhomogeneous finite-horizon Markov Decision Process (MDP) where where $\mathcal{S}, \mathcal{A}$ are the states and actions, respectively, and $H \in \mathbb{Z}$ is the length of each episode. We call $\mathcal{P}_h : \mathcal{S} \times \mathcal{A} \to \Delta\left(\mathcal{S}\right)$ the state-transition probability and $r_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function.

$$V_h^{\pi}(s) = \mathbb{E}\left[\sum_{t=h}^{H} r_t(s_t, a_t) \mid s_h = s, \;\; a_t \sim \pi_t(s_t)\right]$$

Value Function (State-value)

$$Q_h^{\pi}(s, a) = \mathbb{E}\left[\sum_{t=h}^{H} r(s_t, a_t) \mid s_h = s, \;\; a_h = a, \;\; a_t \sim \pi_t(s_t)\right]$$

Q-function (Action-value)

Formal Reinforcement Learning Problem Setting

Useful Identities For Later (Bellman Equations)

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{h=0}^{H} r(s_h, a_h) \mid s_0 = s,\, a_0 = a,\, a_h \sim \pi(\,\cdot \mid s_h)\right]$$

$$= r(s_0,a_0) + \sum_{(s',a') \in \mathcal{S}\times\mathcal{A}} \mathscr{P}(s' \mid s_0, a_0)\pi(a' \mid s')r(s',a')$$

$$= r(s_0,a_0) + \mathbb{E}_{s' \sim \mathscr{P}(\cdot|s_0,a_0)} V^{\pi}(s')$$

$$V^{*}(s) = \max_{\pi \in \Pi} V^{\pi}(s) = \max_{a \in \mathcal{A}} Q^{*}(s,a)$$

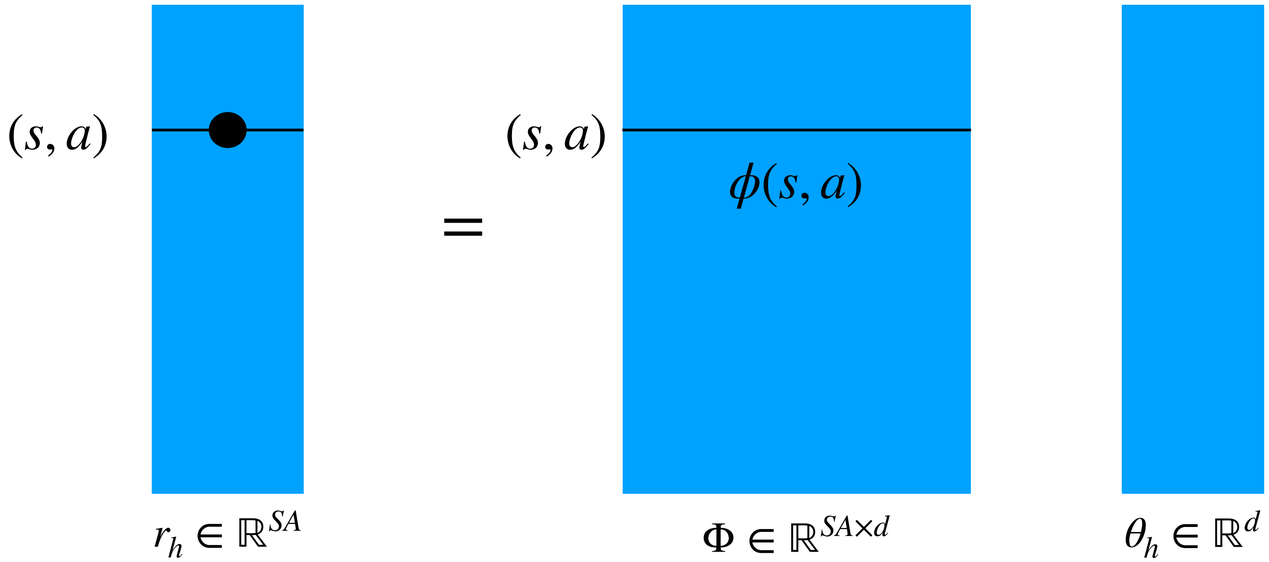
Formal Reinforcement Learning Problem Setting

Let $\mathscr{M} = \left\{ \mathcal{S}, \mathcal{A}, \left\{ r_h \right\}_{h=1}^{H}, \left\{ \mathcal{P}_h \right\}_{h}^{H}, H \right\}$ be an episodic inhomogeneous finite-horizon Markov Decision Process (MDP) where where $\mathcal{S}, \mathcal{A}$ are the states and actions, respectively, and $H \in \mathbb{Z}$ is the length of each episode. We call $\mathcal{P}_h : \mathcal{S} \times \mathcal{A} \to \Delta\left(\mathcal{S}\right)$ the state-transition probability and $r_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function.

$$\mathscr{R}\left(K\right) = \sum_{k=1}^{K} \left[V_1^{*}\left(s_1^k\right) - V_1^{\pi_k}\left(s_1^k\right) \right]$$

Regret

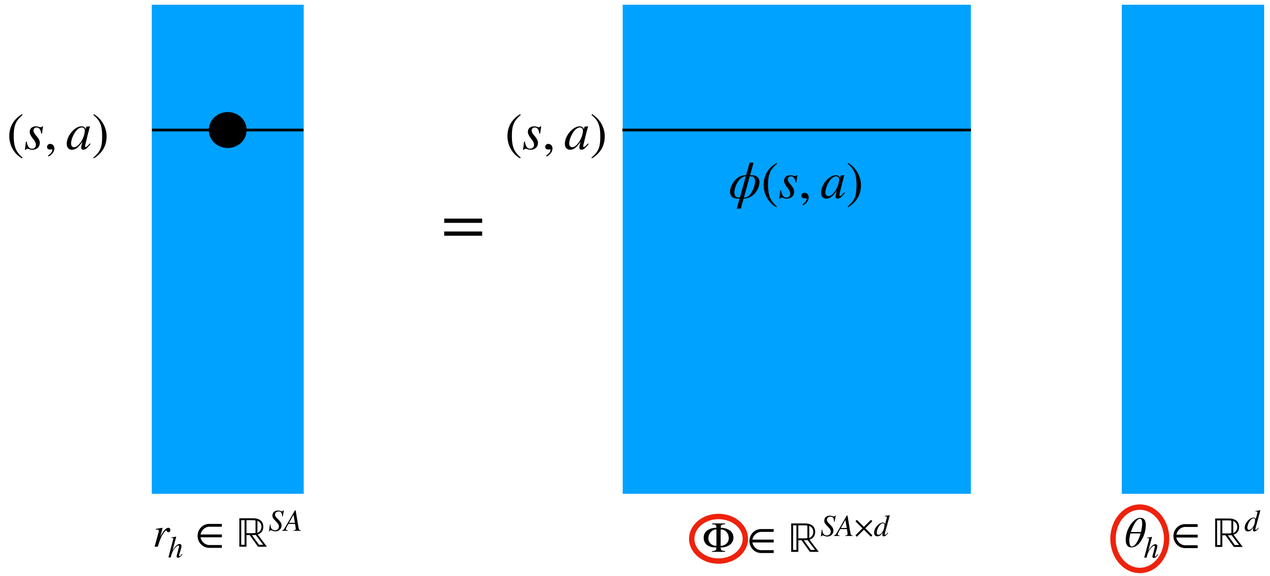
Linear MDP



$$\exists \theta_h, \phi : \forall s, a, \; r_h(s, a) = \theta_h^\top \phi(s, a)$$

Diagram Credit: Wen Sun

Linear MDP

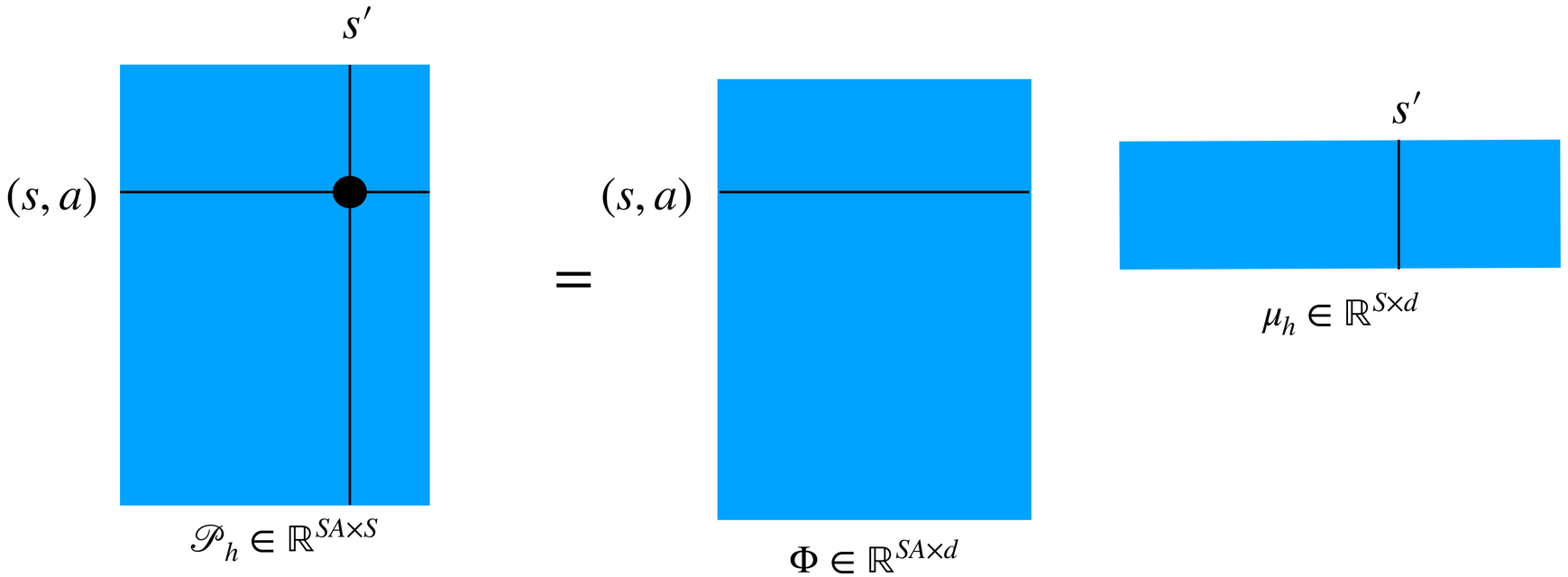


These quantities are know

$$\exists \theta_h, \phi : \forall s, a, r_h(s, a) = \theta_h^\top \phi(s, a)$$

Diagram Credit: Wen Sun

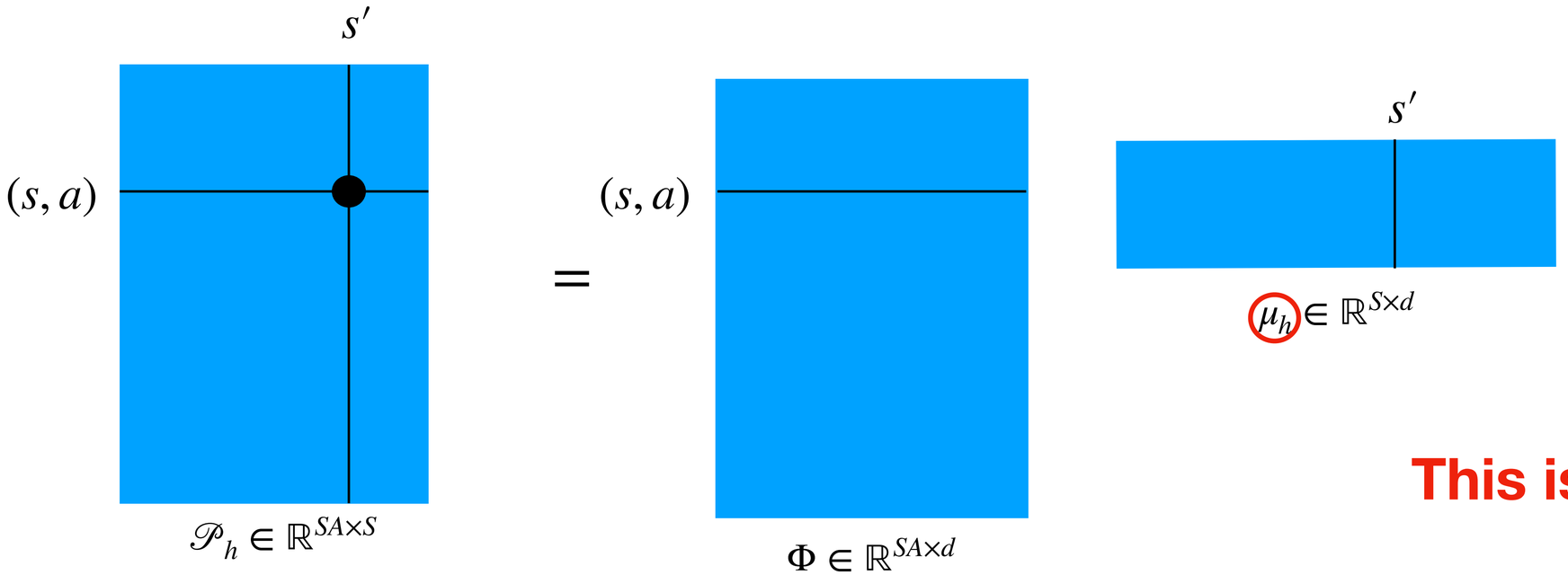
Linear MDP



$$\exists \mu_h, \phi : \forall s, a, h, s', \mathscr{P}_h(s' \mid s, a) = \mu_h(s')^\top \phi(s, a)$$

Diagram Credit: Wen Sun

Linear MDP



This is unknown

$$\exists \mu_h, \phi : \forall s, a, h, s', \mathscr{P}_h(s' \mid s, a) = \mu_h(s')^\top \phi(s, a)$$

Diagram Credit: Wen Sun

Linear MDP $\Longrightarrow$ Action-Value function is linear

$$
\begin{aligned}
Q_h(s,a) &= r_h(s,a) + \mathbb{E}_{s' \sim \mathscr{P}_h(\,\cdot \mid s,a)} V_{h+1}(s') \\
&= \theta_h^\top \phi(s,a) + \mathscr{P}_h(s,a)^\top V_{h+1} \\
&= \theta_h^\top \phi(s,a) + \left(\mu_h V_{h+1}\right)^\top \phi(s,a) \\
&= \left(\theta_h + \mu_h V_{h+1}\right)^\top \phi(s,a) \\
&= w_h^\top \phi(s,a)
\end{aligned}
$$

Linear MDP $\Longrightarrow$ Action-Value function is linear

$$
\begin{aligned}
Q_h(s,a) &= r_h(s,a) + \mathbb{E}_{s' \sim \mathscr{P}_h(\,\cdot \mid s,a)} V_{h+1}(s') \\
&= \theta_h^\top \phi(s,a) + \mathscr{P}_h(s,a)^\top V_{h+1} \\
&= \theta_h^\top \phi(s,a) + \left(\mu_h V_{h+1}\right)^\top \phi(s,a) \\
&= \left(\theta_h + \mu_h V_{h+1}\right)^\top \phi(s,a) \\
&= w_h^\top \phi(s,a)
\end{aligned}
$$

If we learn this, we can estimate Q and thus the optimal policy!

Learning The Transition Dynamics With Ridge Regression

At each time step h, we try to solve

$$\hat{w}_h = \underset{w}{\operatorname{argmin}} \sum_{i=1}^{K} \left(w^\top \phi \left(s_h^{(i)}, a_h^{(i)} \right) - y_h^{(i)} \right)^2 + \lambda \, | \, | \, w \, | \, |_2^2$$

With $\lambda > 0$ and the target labels being:

$$y_h^{(i)} = r_h^{(i)} + \max_{a' \in \mathcal{A}} Q_{h+1} \left(s_{h+1}^{(i)}, a' \right)$$

Learning The Transition Dynamics With Ridge Regression

From the solution of ridge regression, we find that

$$w_h^k = \Lambda_h^{-1} \sum_{i=1}^{k-1} \phi\left(s_h^i, a_h^i\right) \left(r_h\left(s_h^i, a_h^i\right) + \max_{a' \in \mathcal{A}} Q_h\left(s_h^i, a'\right) \right)$$

where $\Lambda_h = \sum_{i=1}^{k-1} \phi\left(s_h^i, a_h^i\right) \phi\left(s_h^i, a_h^i\right)^\top + \lambda I$

LSVI-UCB

Algorithm 1 Least-Squares Value Iteration with UCB (LSVI-UCB)

for episode $k = 1, \ldots, K$ **do**
  Receive the initial state x_1^k.
  for step $h = H, \ldots, 1$ **do**
    $\Lambda_h \leftarrow \sum_{\tau=1}^{k-1} \boldsymbol{\phi}(x_h^\tau, a_h^\tau)\boldsymbol{\phi}(x_h^\tau, a_h^\tau)^\top + \lambda \cdot \mathbf{I}$.
    $\mathbf{w}_h \leftarrow \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \boldsymbol{\phi}(x_h^\tau, a_h^\tau)[r_h(x_h^\tau, a_h^\tau) + \max_a Q_{h+1}(x_{h+1}^\tau, a)]$.
    $Q_h(\cdot, \cdot) \leftarrow \min\{\mathbf{w}_h^\top \boldsymbol{\phi}(\cdot, \cdot) + \beta[\boldsymbol{\phi}(\cdot, \cdot)^\top \Lambda_h^{-1} \boldsymbol{\phi}(\cdot, \cdot)]^{1/2}, H\}$.
  for step $h = 1, \ldots, H$ **do**
    Take action $a_h^k \leftarrow \text{argmax}_{a \in \mathcal{A}} Q_h(x_h^k, a)$, and observe x_{h+1}^k.

Diagram Credit: Provably Efficient Reinforcement Learning with Linear Function Approximation by Jin et al. 2020

LSVI-UCB

Algorithm 1 Least-Squares Value Iteration with UCB (LSVI-UCB)

for episode $k = 1, \ldots, K$ **do**
  Receive the initial state x_1^k.
  for step $h = H, \ldots, 1$ **do**
    $\Lambda_h \leftarrow \sum_{\tau=1}^{k-1} \boldsymbol{\phi}(x_h^\tau, a_h^\tau)\boldsymbol{\phi}(x_h^\tau, a_h^\tau)^\top + \lambda \cdot \mathbf{I}$.
    $\mathbf{w}_h \leftarrow \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \boldsymbol{\phi}(x_h^\tau, a_h^\tau)[r_h(x_h^\tau, a_h^\tau) + \max_a Q_{h+1}(x_{h+1}^\tau, a)]$.
    $Q_h(\cdot, \cdot) \leftarrow \min\{\mathbf{w}_h^\top \boldsymbol{\phi}(\cdot, \cdot) + \beta[\boldsymbol{\phi}(\cdot, \cdot)^\top \Lambda_h^{-1} \boldsymbol{\phi}(\cdot, \cdot)]^{1/2}, H\}$.
  for step $h = 1, \ldots, H$ **do**
    Take action $a_h^k \leftarrow \text{argmax}_{a \in \mathcal{A}} Q_h(x_h^k, a)$, and observe x_{h+1}^k.

Diagram Credit: Provably Efficient Reinforcement Learning with Linear Function Approximation by Jin et al. 2020

LSVI-UCB Bonus

The agent is learning from limited data. Like a regression confidence interval, we want to hedge against uncertainty in our estimate of $\hat{w}_h$. For any new (s, a), the uncertainty in prediction is proportional to

$$\beta \left|\left| \phi(s,a) \right|\right|_{\Lambda_h^{-1}} = \beta \sqrt{\phi(s,a)^\top \Lambda_h^{-1} \phi(s,a)}$$

where $\beta = dH\sqrt{d \log\left(\frac{dKH}{\delta}\right)}$

LSVI-UCB Bonus

The agent is learning from limited data. Like a regression confidence interval, we want to hedge against uncertainty in our estimate of $\hat{w}_h$. For any new (s, a), the uncertainty in prediction is proportional to

$$\beta \left| \left| \phi\left(s,a\right) \right| \right|_{\Lambda_h^{-1}} = \beta \sqrt{\phi\left(s,a\right)^\top \Lambda_h^{-1} \phi\left(s,a\right)}$$

where $\beta = dH\sqrt{d\log\left(\frac{dKH}{\delta}\right)}$

Intuition: Another way to think about this is that this is a carefully curated bonus given to our agent that promotes exploration by taking actions that are less certain. It ensures that with high probability $Q_h^k(s,a)$ is an upper confidence bound of the true Q function $Q_h^*\left(s,a\right)\ \forall(s,a)$

LSVI-UCB Bonus

The agent is learning from limited data. Like a regression confidence interval, we want to hedge against uncertainty in our estimate of $\hat{w}_h$. For any new (s, a), the uncertainty in prediction is proportional to

$$\beta \left| \left| \phi(s,a) \right| \right|_{\Lambda_h^{-1}} = \beta \sqrt{\phi(s,a)^\top \Lambda_h^{-1} \phi(s,a)}$$

where $\beta = dH\sqrt{d \log\left(\frac{dKH}{\delta}\right)}$

How do we get this?

Intuition: Another way to think about this is that this is a carefully curated bonus given to our agent that promotes exploration by taking actions that are less certain. It ensures that with high probability $Q_h^k(s,a)$ is an upper confidence bound of the true Q function $Q_h^*(s,a)\ \forall (s,a)$

LSVI-UCB Bonus Proof Sketch

Lemma A (Theorem 1 in Abbasi-Yadkori et al. 2011). Let $\{\mathcal{F}_t\}_{t=0}^{\infty}$ be a filtration. Let $\{\eta_t\}_{t=1}^{\infty}$ be a real-valued stochastic process such that η_t is $\mathcal{F}_t$ measurable and η_t is conditionally R-sub-Gaussian for some $R \geq 0$ i.e.

$$\forall \lambda \in \mathbb{R}, \quad \mathbb{E}\left[e^{\lambda \eta_t} \mid \mathcal{F}_{t-1}\right] \leq \exp\left(\frac{\lambda^2 R^2}{2}\right)$$

Let $\{x_t\}_{t=1}^{\infty}$ be an $\mathbb{R}^d$ -valued stochastic process such that x_t is $\mathcal{F}_{t-1}$-measurable. Assume that Z is a $d \times d$ positive definite matrix. For any $k \geq 0$, define

$$Z_k = Z + \sum_{s=1}^{t} X_s X_s^{\top}$$

Then, for any $\delta > 0$, with probability at least $1 - \delta$, for all $t \geq 0$

$$\| \sum_{i=1}^{k} x_i \eta_i \|_{Z_k^{-1}}^2 \leq 2R^2 \log\left(\frac{\det(Z_k)^{1/2} \det(Z)^{-1/2}}{\delta}\right)$$

This is a self-normalizing martingale bound using **Azuma-Hoeffding**

LSVI-UCB Bonus Proof Sketch

Lemma B (Lemma D.6 of Jin et al. 2020). Let $\mathscr{V}$ denote a class of functions mapping from $\mathcal{S}$ to $\mathbb{R}$ with the following parametric form

$$V(\cdot) = \min\left\{\max_a \left[w^\top \phi(\cdot, a) + \beta\sqrt{\phi(\cdot, a)^\top \Lambda^{-1} \phi(\cdot, a)}\right], H\right\}$$

where the parameters (w, β, Λ) satisfy $\|w\| \leq L$, $\beta \in [0, B]$, and the minimum eigenvalue satisfies $\lambda_{\min}(\Lambda) \geq \lambda$. Assume $\|\phi(s, a)\| \leq 1$ for all (s, a) pairs, and let $\mathcal{N}_\varepsilon$ be the ε-covering number of $\mathscr{V}$ with respect to distance

$$\operatorname{dist}(V, V') = \sup_s |V(s) - V'(s)|$$

Then,

$$\log \mathcal{N}_\varepsilon \leq d \log\left(1 + \frac{4L}{\varepsilon}\right) + d^2 \log\left(1 + \frac{8\sqrt{d}B^2}{\lambda \varepsilon^2}\right).$$

LSVI-UCB Regret

$$\mathscr{R}(K) = \sum_{k=1}^{K} \left[V_1^* \left(s_1^k \right) - V_1^{\pi_k} \left(s_1^k \right) \right]$$

$$\leq \sum_{k=1}^{K} \left[V_1^k \left(s_1^k \right) - V_1^{\pi_k} \left(s_1^k \right) \right]$$

$$\leq \sum_{k=1}^{K} \sum_{h=1}^{H} \zeta_k^h + 2\beta \sum_{k=1}^{K} \sum_{h=1}^{H} \sqrt{(\phi_k^h)^\top (\Lambda_k^h)^{-1} \phi_k^h}$$

$$\underset{\sim}{<} \widetilde{\mathcal{O}} \left(d^3 H^4 K \right)$$

Privacy Concerns In LSVI-UCB

Algorithm 1 Least-Squares Value Iteration with UCB (LSVI-UCB)

for episode $k = 1, \ldots, K$ **do**
  Receive the initial state x_1^k.
  for step $h = H, \ldots, 1$ **do**
    $\Lambda_h \leftarrow \sum_{\tau=1}^{k-1} \boldsymbol{\phi}(x_h^\tau, a_h^\tau)\boldsymbol{\phi}(x_h^\tau, a_h^\tau)^\top + \lambda \cdot \mathbf{I}$.
    $\mathbf{w}_h \leftarrow \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \boldsymbol{\phi}(x_h^\tau, a_h^\tau)[r_h(x_h^\tau, a_h^\tau) + \max_a Q_{h+1}(x_{h+1}^\tau, a)]$.
    $Q_h(\cdot, \cdot) \leftarrow \min\{\mathbf{w}_h^\top \boldsymbol{\phi}(\cdot, \cdot) + \beta[\boldsymbol{\phi}(\cdot, \cdot)^\top \Lambda_h^{-1} \boldsymbol{\phi}(\cdot, \cdot)]^{1/2}, H\}$.
  for step $h = 1, \ldots, H$ **do**
    Take action $a_h^k \leftarrow \text{argmax}_{a \in \mathcal{A}} Q_h(x_h^k, a)$, and observe x_{h+1}^k.

Diagram Credit: Provably Efficient Reinforcement Learning with Linear Function Approximation by Jin et al. 2020

Privacy Concerns In LSVI-UCB

Algorithm 1 Least-Squares Value Iteration with UCB (LSVI-UCB)

for episode $k = 1, \ldots, K$ **do**
  Receive the initial state x_1^k.
  for step $h = H, \ldots, 1$ **do**
    $\Lambda_h \leftarrow \sum_{\tau=1}^{k-1} \boldsymbol{\phi}(x_h^\tau, a_h^\tau)\boldsymbol{\phi}(x_h^\tau, a_h^\tau)^\top + \lambda \cdot \mathbf{I}$.
    $\mathbf{w}_h \leftarrow \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \boldsymbol{\phi}(x_h^\tau, a_h^\tau)[r_h(x_h^\tau, a_h^\tau) + \max_a Q_{h+1}(x_{h+1}^\tau, a)]$.
    $Q_h(\cdot, \cdot) \leftarrow \min\{\mathbf{w}_h^\top \boldsymbol{\phi}(\cdot, \cdot) + \beta[\boldsymbol{\phi}(\cdot, \cdot)^\top \Lambda_h^{-1} \boldsymbol{\phi}(\cdot, \cdot)]^{1/2}, H\}$.
  for step $h = 1, \ldots, H$ **do**
    Take action $a_h^k \leftarrow \text{argmax}_{a \in \mathcal{A}} Q_h(x_h^k, a)$, and observe x_{h+1}^k.

Think of $\phi(s, a)$ as a one-hot vector, then Λ_h is capturing something similar to visitation counts which uses trajectory information with possibly private data

Diagram Credit: Provably Efficient Reinforcement Learning with Linear Function Approximation by Jin et al. 2020

Privacy Concerns In LSVI-UCB

Algorithm 1 Least-Squares Value Iteration with UCB (LSVI-UCB)

for episode $k = 1, \ldots, K$ **do**
  Receive the initial state x_1^k.
  for step $h = H, \ldots, 1$ **do**
    $\Lambda_h \leftarrow \sum_{\tau=1}^{k-1} \boldsymbol{\phi}(x_h^\tau, a_h^\tau)\boldsymbol{\phi}(x_h^\tau, a_h^\tau)^\top + \lambda \cdot \mathbf{I}$.
    $\mathbf{w}_h \leftarrow \Lambda_h^{-1} \sum_{\tau=1}^{k-1} \boldsymbol{\phi}(x_h^\tau, a_h^\tau)[r_h(x_h^\tau, a_h^\tau) + \max_a Q_{h+1}(x_{h+1}^\tau, a)]$.
    $Q_h(\cdot, \cdot) \leftarrow \min\{\mathbf{w}_h^\top \boldsymbol{\phi}(\cdot, \cdot) + \beta[\boldsymbol{\phi}(\cdot, \cdot)^\top \Lambda_h^{-1} \boldsymbol{\phi}(\cdot, \cdot)]^{1/2}, H\}$.
  for step $h = 1, \ldots, H$ **do**
    Take action $a_h^k \leftarrow \text{argmax}_{a \in \mathcal{A}} Q_h(x_h^k, a)$, and observe x_{h+1}^k.

These are parameter estimates for the feature regressors which allow us to calculate the Q-function due to Linear MDPs. These can also leak information about trajectories taken by the policy

We need to privatize these terms!

Diagram Credit: Provably Efficient Reinforcement Learning with Linear Function Approximation by Jin et al. 2020

Differential Privacy Tools

Lemma C (Lemma 1.7 of Bun and Steiner. 2016). Let $\mathscr{A} : \mathscr{X}^n \to \mathscr{Y}$ and $\mathscr{A}' : \mathscr{X}^n \times \mathscr{Y} \to \mathscr{Z}$ be (possibly randomized) mechanisms. For $\delta > 0$, suppose $\mathscr{A}$ satisfies (ε_1, δ)-DP, and for each $y \in \mathscr{Y}$, $\mathscr{A}'(\cdot, y)$ satisfies (ε_2, δ)-DP. Define the composed mechanism

$$\mathscr{A}''(x) = \mathscr{A}'(x, \mathscr{A}(x))$$

Then, $\mathscr{A}''$ satisfies $(\varepsilon_1 + \varepsilon_2, 2\delta)$-DP.

Lemma D (Theorem 3.22 of Dwork and Roth. 2014). Let $f : \mathbb{N}^{\mathscr{X}} \to \mathbb{R}^d$ be an arbitrary d-dimensional function with

$$\Delta(f) = \max_{\mathscr{U} \sim \mathscr{U}'} ||f(\mathscr{U}) - f(\mathscr{U}')||_2$$

where $\mathscr{U} \sim \mathscr{U}'$ are neighboring datasets. The Gaussian mechanism $\mathscr{M}_{\text{Gauss}}$ with noise level σ is given by

$$\mathscr{M}_{\text{Gauss}}(\mathscr{U}) = f(\mathscr{U}) + \mathcal{N}(0, \sigma^2 I_d)$$

For all $0 < \delta, \epsilon < 1$, a Gaussian Mechanism with noise parameter $\sigma = \frac{\Delta}{\epsilon}\sqrt{2\log(1.25/\delta)}$ satisfies (ϵ, δ)-DP.

Differential Privacy Tools

Lemma E (Billboard Lemma of Hsu et al. 2013). Suppose that a randomized mechanism $\mathscr{A} : \mathscr{X}^n \to \mathscr{Y}$ is (ϵ, δ)-DP. Let $U \in \mathscr{U}$ be a dataset containing n users. Then, consider any set of functions $f_i : \mathscr{U}_i \times \mathscr{Y} \to \mathscr{Y}_i$ for $i \in [n]$ where $\mathscr{U}_i$ is the portion of the dataset containing user i's data. Then, the composition $\left\{ f_i \left(\Pi_i (U), \mathscr{A}(U) \right) \right\}_{i \in [n]}$ is (ϵ, δ)-JDP where $\Pi_i : \mathscr{U} \to \mathscr{U}_i$ is the canonical projection to the i-th users data

Remark: This lemma tells us that if we construct an (ϵ, δ)-DP algorithm and we have a function f_i that operates on user i's data, then that mechanism is indeed (ϵ, δ).

Previous Work

[Theorem 8 of Luyo et al. 2021]. Fix any privacy level $\varepsilon, \delta \in (0,1)$. For any $p \in (0,1)$, their algorithm is (ε, δ)-JDP and, with probability at least $1-p$, its regret is bounded as follows:

$$\mathscr{R}(K) = \widetilde{O}\left(\sqrt{d^3H^4K} + H^{11/5}d^{8/5}K^{3/5}/\varepsilon^{2/5}\right)$$

Approach: Privatize Λ_h, w_h

$$\widetilde{\Lambda}_h = \Lambda_h + \mathcal{N}\left(0, \mathcal{O}\left(K^{1/5}d^{3/10}H^{2/5}\epsilon^{-4/5}\log(1/\delta)\right)\right)$$

$$\tilde{w}_{b+1,h} = \tilde{\Lambda}_{b+1,h}^{-1}\sum_{i=1}^{b+1}\phi(s_{i,h},a_{i,h})\left[r_h(s_{i,h},a_{i,h}) + V_{b+1,h+1}(s_{i,h+1})\right] + \mathcal{N}\left(0, \tilde{\Lambda}_{b+1,h}^{-1}\cdot\mathcal{O}\left(\frac{1}{\epsilon}H^2B\log\left(\frac{1}{\delta}\right)\right)\cdot\tilde{\Lambda}_{b+1,h}^{-1}\right)$$

Static Batching to reduce the number of policy switches to $\mathcal{O}\left(\text{poly}(K)\right)$

Previous Work

[Theorem 16 of Ngo et al. 2022]. Fix any privacy level $\varepsilon, \delta \in (0,1)$. For any $p \in (0,1)$, their algorithm is (ε, δ)-JDP and, with probability at least $1 - p$, its regret is bounded as follows:

$$\mathscr{R}(K) \leq \widetilde{O}\left(\sqrt{d^3H^4K} + H^3d^{5/4}K^{1/2}/\varepsilon^{1/2}\right)$$

Approach: Same techniques as previous work but instead of a static batching schedule, they use **Adaptive Batching** to reduce the number of policy switches to $\mathcal{O}\left(\log(K)\right)$

Motivating Work: LSVI-UCB++

[Theorem 5.1 of He et al. 2023]. For any linear MDP $\mathscr{M}$ with K sufficiently large, if we set the parameters $\lambda = 1/H^2$ and the confidence intervals $\beta, \bar{\beta}, \widetilde{\beta}$ as

$$\beta = \mathcal{O}\left(H\sqrt{d\lambda} + \sqrt{d\log^2\left(1 + dKH/(\delta\lambda)\right)}\right)$$
$$\bar{\beta} = \mathcal{O}\left(H\sqrt{d\lambda} + \sqrt{d^3H^2\log^2\left(dHK/(\delta\lambda)\right)}\right)$$
$$\widetilde{\beta} = \mathcal{O}\left(H^2\sqrt{d\lambda} + \sqrt{d^3H^4\log^2\left(dHK/(\delta\lambda)\right)}\right)$$

then with high probability of at least $1 - 7\delta$, the regret of LSVI-UCB++ is upper bounded as follows

$$\mathscr{R}(K) \leq \widetilde{\mathcal{O}}\left(d\sqrt{H^3K}\right)$$

Instead of solving a ridge regression problem, we solve a **weighted ridge regression** problem using estimated weights from data. This allows us to use a self-normalized martingale argument using **Azuma-Bernstein** rather than **Azuma-Hoeffding** to get a bonus that improves our regret.

Motivating Work: JDP In Tabular MDPs

[Theorem 4.1 of Qiao and Wang. 2023]. For any privacy budget $\epsilon > 0$, failure probability $0 < \beta < 1$, and any privatizer where the private counts are close to the true counts with high probability, with probability at least $1 - \beta$, their algorithm is (ϵ, δ)-JDP and achieves regret upper bounded by:

$$\mathscr{R}(K) \leq \widetilde{O}\left(\sqrt{H^3SAK} + S^2AH^3/\epsilon\right)$$

In previous work, since we would use a **Hoeffding-bound** that only depends on the counts, it is sufficient to **privatize the counts** loosely using Gaussian noise with a sufficient variance component. However, to use a **Bernstein-bound**, we need to **carefully privatize the bounds** to ensure that we can **upper bound the variance term** in a Bernstein-bound

Can we design a (ϵ, δ)-JDP algorithm that is near minimax optimal for non-private learning and improves the cost of privacy using more refined privatization and concentration techniques?

$$\mathscr{R}(K) \leq \widetilde{O}\left(\sqrt{d^3H^4K} + H^3d^{5/4}K^{1/2}/\varepsilon^{1/2}\right)$$

Non-private learning regret: We can do better using LSVI-UCB++

Cost of privacy: can we improve this to

$$\mathcal{O}\left(\text{poly}\left(dHK\right)/\epsilon\right)$$

Yes We Can: DP-LSVI-UCB++

[Theorem 3.2 of Sahu. 2025]. For any linear MDP $\mathscr{M}$ with K sufficiently large, if we set the parameters

$$\lambda_{\widetilde{\Lambda}} = \mathcal{O}\left(\sqrt{dHK}\left(2 + \left(\frac{\log\left(5H/\delta\right)}{d}\right)^{2/3}\right)\right)$$

$$L = \mathcal{O}\left(H\sqrt{dHK\log\left(dKH/\delta\right)}\right)$$

and the confidence intervals $\hat{\beta}, \check{\beta}, \bar{\beta}$ as

$$\hat{\beta} = \check{\beta} = \mathcal{O}\left(HL\sqrt{d\lambda_{\widetilde{\Lambda}}} + \sqrt{d^3H^2\log^2\left(dH^3KL^2/\left(\delta\lambda_{\widetilde{\Lambda}}\right)\right)}\right)$$

$$\bar{\beta} = \mathcal{O}\left(H^2L^2\sqrt{d\lambda_{\widetilde{\Lambda}}} + \sqrt{d^3H^4\log^2\left(dH^4KL^2/\left(\delta\lambda_{\widetilde{\Lambda}}\right)\right)}\right)$$

then with high probability of at least $1 - 7\delta$, the regret of DP-LSVI-UCB++ is upper bounded as follows

$$\mathscr{R}(K) \leq \widetilde{\mathcal{O}}\left(d\sqrt{H^3K} + H^{15/4}d^{7/6}K^{1/2}/\epsilon\right)$$

Algorithm 1 LSVI-UCB++

Require: Regularization parameter $\lambda > 0$, confidence radius $\beta, \bar{\beta}, \widetilde{\beta}$

Initialize $k_{\text{last}} = 0$ and for each stage $h \in [H]$ set $\mathbf{\Sigma}_{0,h}, \mathbf{\Sigma}_{1,h} \leftarrow \lambda \mathbf{I}$

For each stage $h \in [H]$ and state-action $(s, a) \in \mathcal{S} \times \mathcal{A}$, set $Q_{0,h}(s, a) \leftarrow H, \check{Q}_{0,h}(s, a) \leftarrow 0$

for episodes $k = 1, \ldots, K$ **do**

  Received the initial state s_1^k.

  for stage $h = H, \ldots, 1$ **do**

    $\widehat{\mathbf{w}}_{k,h} = \mathbf{\Sigma}_{k,h}^{-1} \sum_{i=1}^{k-1} \bar{\sigma}_{i,h}^{-2} \boldsymbol{\phi}(s_h^i, a_h^i) V_{k,h+1}(s_{h+1}^i)$

    $\check{\mathbf{w}}_{k,h} = \mathbf{\Sigma}_{k,h}^{-1} \sum_{i=1}^{k-1} \bar{\sigma}_{i,h}^{-2} \boldsymbol{\phi}(s_h^i, a_h^i) \check{V}_{k,h+1}(s_{h+1}^i)$

    if there exists a stage $h' \in [H]$ such that $\det(\mathbf{\Sigma}_{k,h'}) \geq 2 \det(\mathbf{\Sigma}_{k_{\text{last}},h'})$ **then**

      $Q_{k,h}(s, a) = \min \Big\{ r_h(s, a) + \widehat{\mathbf{w}}_{k,h}^\top \boldsymbol{\phi}(s, a) + \beta \sqrt{\boldsymbol{\phi}(s, a)^\top \mathbf{\Sigma}_{k,h}^{-1} \boldsymbol{\phi}(s, a)}, Q_{k-1,h}(s, a), H \Big\}$

      $\check{Q}_{k,h}(s, a) = \max \Big\{ r_h(s, a) + \check{\mathbf{w}}_{k,h}^\top \boldsymbol{\phi}(s, a) - \bar{\beta} \sqrt{\boldsymbol{\phi}(s, a)^\top \mathbf{\Sigma}_{k,h}^{-1} \boldsymbol{\phi}(s, a)}, \check{Q}_{k-1,h}(s, a), 0 \Big\}$

      Set the last updating episode $k_{\text{last}} = k$

    else

      $Q_{k,h}(s, a) = Q_{k-1,h}(s, a)$

      $\check{Q}_{k,h}(s, a) = \check{Q}_{k-1,h}(s, a)$

    end if

    $V_{k,h}(s) = \max_a Q_{k,h}(s, a)$

    $\check{V}_{k,h}(s) = \max_a \check{Q}_{k,h}(s, a)$

  end for

  for stage $h = 1, \ldots, H$ **do**

    Take action $a_h^k \leftarrow \operatorname{argmax}_a Q_{k,h}(s_h^k, a)$

    Set the estimated variance $\sigma_{k,h}$ as in (4.1)

    $\bar{\sigma}_{k,h} \leftarrow \max \left\{ \sigma_{k,h}, H, 2d^3 H^2 \|\boldsymbol{\phi}(s_h^k, a_h^k)\|_{\mathbf{\Sigma}_{k,h}^{-1}}^{1/2} \right\}$

    $\mathbf{\Sigma}_{k+1,h} = \mathbf{\Sigma}_{k,h} + \bar{\sigma}_{k,h}^{-2} \boldsymbol{\phi}(s_h^k, a_h^k) \boldsymbol{\phi}(s_h^k, a_h^k)^\top$

    Receive next state s_{h+1}^k

  end for

end for

We take these terms from LSVI-UCB++ and privatize them with sufficient noise using a Gaussian mechanism

DP-LSVI-UCB++ Privacy Guarantee

[Theorem 3.1 of Sahu. 2025]. DP-LSVI-UCB++ satisfies (ϵ, δ)-JDP where $\epsilon = \rho + 2\sqrt{\rho \log(1/\delta)}$ for $\rho > 0$

Proof Sketch:

1. Compute the l_2-sensitivity of each privatized estimator by considering neighboring sequences $\mathcal{U}, \mathcal{U}'$
2. Use a Gaussian mechanism with sufficient noise to ensure each is $\rho/4KH$-zCDP
3. By Advanced Composition, it is ρ-zCDP
4. By a conversion from zCDP to DP, DP-LSVI-UCB++ is (ϵ, δ)-DP
5. Use the Billboard Lemma to conclude that DP-LSVI-UCB++ is (ϵ, δ)-JDP.

DP-LSVI-UCB++ Privacy Guarantee

[Theorem 3.1 of Sahu. 2025]. DP-LSVI-UCB++ satisfies (ϵ, δ)-JDP where $\epsilon = \rho + 2\sqrt{\rho \log(1/\delta)}$ for $\rho > 0$

Proof Sketch:

1. Compute the l_2-sensitivity of each privatized estimator by considering neighboring sequences $\mathcal{U}, \mathcal{U}'$
2. Use a Gaussian mechanism with sufficient noise to ensure each is $\rho/4KH$-zCDP
3. By Advanced Composition, it is ρ-zCDP
4. By a conversion from zCDP to DP, DP-LSVI-UCB++ is (ϵ, δ)-DP
5. Use the Billboard Lemma to conclude that DP-LSVI-UCB++ is (ϵ, δ)-JDP.

DP-LSVI-UCB++ Privacy Guarantee

[Theorem 3.

0

Proof S

1

oring

2

DP

3

4

5.

Definition (Zero-Concentrated Differential Privacy (zCDP)). A randomized mechanism $\mathscr{M} : \mathscr{X}^n \to \mathscr{Y}$ is (ξ, ρ)-zCDP if $\forall x, x' \in \mathscr{X}^n$ differing on a single entry and all $\alpha \in (1, \infty)$

$$D_\alpha \left(\mathscr{M}(x) \,||\, \mathscr{M}(x') \right) \leq \xi + \rho\alpha$$

where $D_\alpha \left(\mathscr{M}(x) \,||\, \mathscr{M}(x') \right)$ is the α-Rényi divergence between distribution $\mathscr{M}(x)$ and $\mathscr{M}(x')$. Equivalently,

$$\mathbb{E}\left[e^{(\alpha-1)Z} \right] \leq e^{(\alpha-1)(\xi+\rho\alpha)}$$

DP-LSVI-UCB++ Regret Proof Sketch

1. Use the privatized terms and prove their utility i.e. how close are they to the non-privatized terms. Since we used a Gaussian mechanism, it is sufficient to use a Gaussian concentration inequality on the matrix operator norm or the l_2-norm

2. Use the private terms in place of the non-private terms and use the arguments from LSVI-UCB++ to find the upper confidence bonuses using a Bernstein self-normalized concentration inequality, uniform covering arguments, elliptical potentials, and utility of the privatized terms

3. Use the bonuses to prove optimism and pessimism of the privatized Q-value function

DP-LSVI-UCB++ Regret Proof Sketch

First, let us define the following events

$\mathscr{E}$: Accurate value predictions despite privatized regression

$\widetilde{\mathscr{E}}$: Sharper bounds using Bernstein-style control of noise and variance

$\mathscr{E}_1$: Estimated values of learned policy $\approx$ actual value of true policy (uniform stability)

$\mathscr{E}_2$: Optimistic and pessimistic values sandwich the truth (stability under noise)

$\mathscr{E}_3$: Total estimation variance is bounded (learnability of the environment)

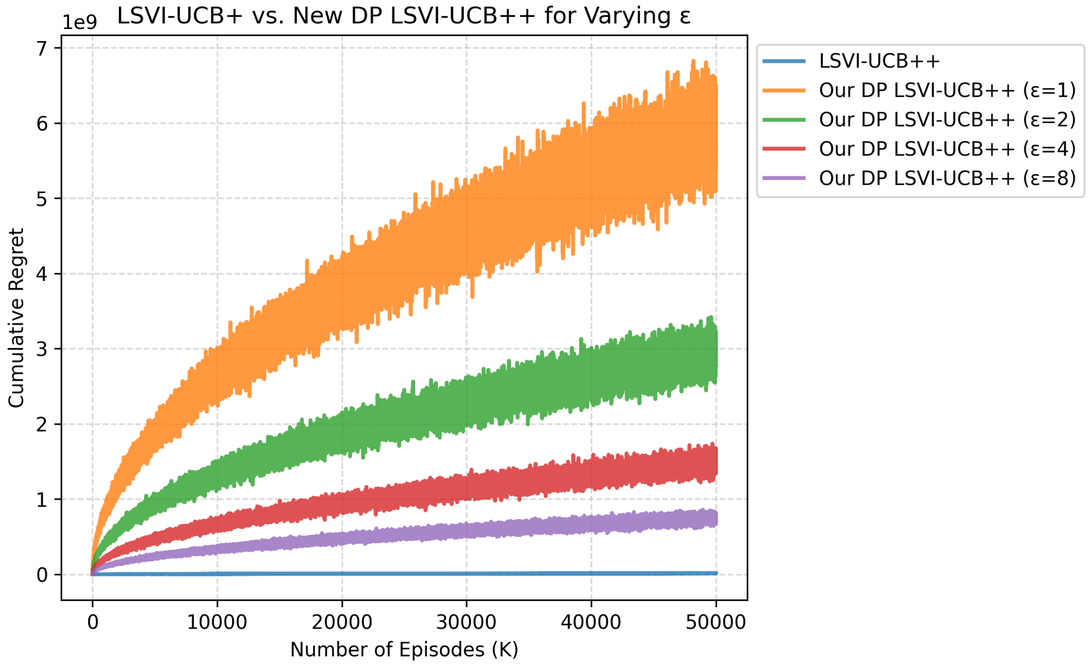
DP-LSVI-UCB++ Regret Proof Sketch

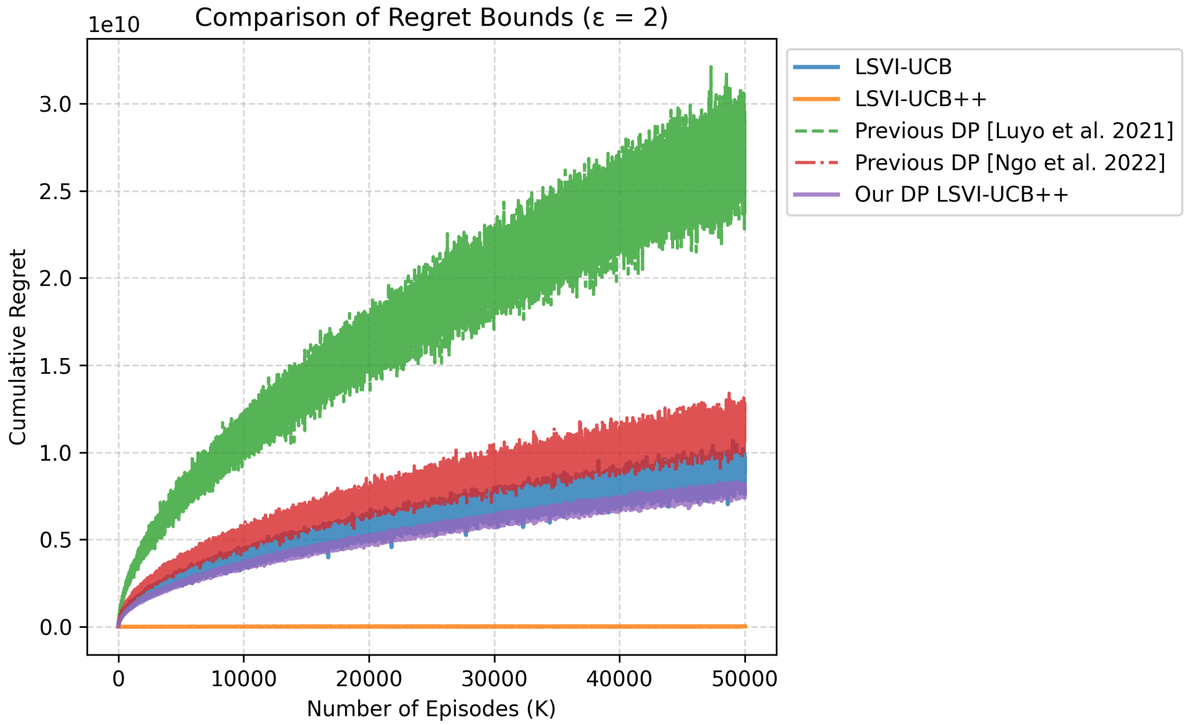
Conditioned on the event $\mathscr{E} \cap \widetilde{\mathscr{E}} \cap \mathscr{E}_1 \cap \mathscr{E}_2 \cap \mathscr{E}_3$

$$
\begin{aligned}
\mathscr{R}(K) &= \sum_{k=1}^{K} \left(V_1^*(s_1^k) - \widetilde{\widehat{V}}^{\pi_k}_{k,1}(s_1^k) \right) \\
&\leq \sum_{k=1}^{K} \left(\widetilde{\widehat{V}}_{k,1}(s_1^k) - \widetilde{\widehat{V}}^{\hat{\pi}^k}_{k,1}(s_1^k) \right) \\
&\leq 16 d^4 H^8 \iota + 40 \beta d^7 H^5 \iota + 8\beta \sqrt{2dH\iota \sum_{h=1}^{H} \sum_{k=1}^{K} (\tilde{\sigma}_{k,h}^2 + H)} + 4\sqrt{H^3 K \log(H/\delta)} \\
&\leq \widetilde{O}\left(d\sqrt{H^3 K} + \frac{H^{15/4} d^{7/6} K^{1/2} \log(10dKH/\delta)}{\epsilon} \right)
\end{aligned}
$$

DP-LSVI-UCB++ enjoys a privacy guarantee at (almost) no drop in utility

Environment Setup: We use a 6-state chain environment with two actions: left and right. The agent starts on the left and aims to reach the rightmost state for higher rewards. we set the planning horizon $H = 20$ and run $K = 50{,}000$ episodes.





Conclusion

- DP-LSVI-UCB++ is a (ϵ, δ)**-JDP** algorithm for linear MDP that achieves the new **state-of-the-art regret bound** by using **Bernstein concentration, Gaussian mechanisms**, and **GOE perturbations** for tight utility-privacy tradeoff
- Our results show that theoretically and empirically, we match or outperform non-private baselines and do better than previous work in this area
- Future directions
  - Extending these results to the low-rank MDP setting
  - Exploring alternative mechanism that adapt noise dynamically based on the observed data's sensitivity could lead to improved regret bounds

Thanks For Listening! Questions?