SaferQ: Obfuscating Web User Search Queries Via Generative Adversarial Privacy

EE 599 Final Project
Team No. 33

Jiang Zhang, Zhongxuan Ruan, Mengwei Yang
2020.5.6

Content

Content

- Part I: Introduction & Problem
  - *Web query privacy and its challenges*
  - *Problem and motivation*
  - *Our contributions*
- Part II: Approach & Design
  - *Generative adversarial privacy (GAP)*
  - *System architecture*
  - *System optimization*
- Part III: Evaluation & Conclusion
  - *Experiment, results and analysis*
  - *Discussion, conclusion and future works*

Web query privacy and its challenging

How web queries leak privacy:

- Browsers/search engines will store web user search query logs.
- They obtain profits by "selling" user profiles.
- Privacy leakage happens when they create user profiles.

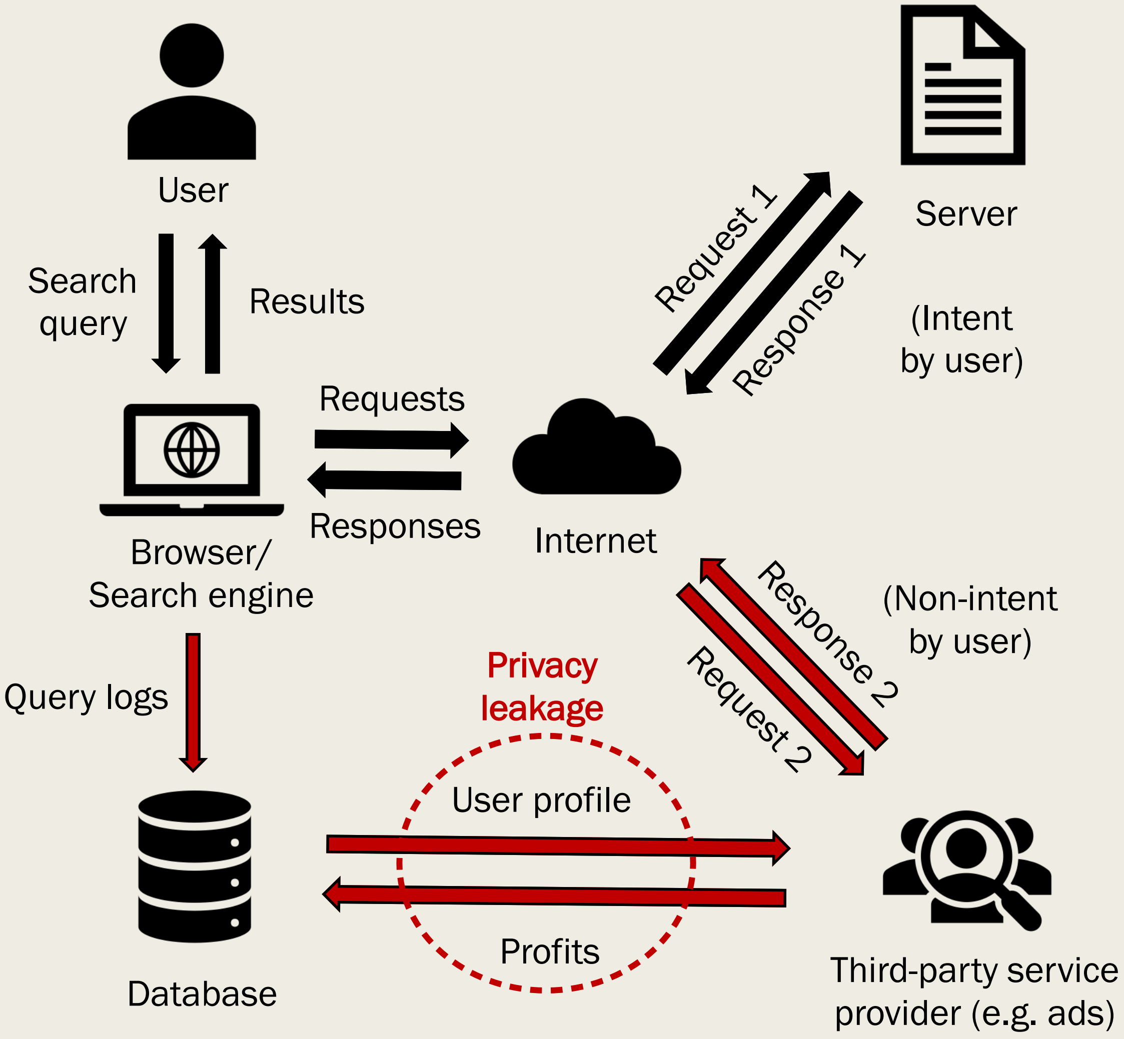


Fig. 1-1. How web query can leak your privacy.

Web query privacy and its challenges

How web queries leak privacy:

- Browsers/search engines will store web user search query logs.
- They obtain profits by “selling” user profiles.
- Privacy leakage happens when they create user profiles.

Challenges:

- “Trackers” are widespread in the web; hard to measure the privacy leakage.
- Users still need third-party services (e.g. recommendation, advertising).
- Users want to improve their search experience.

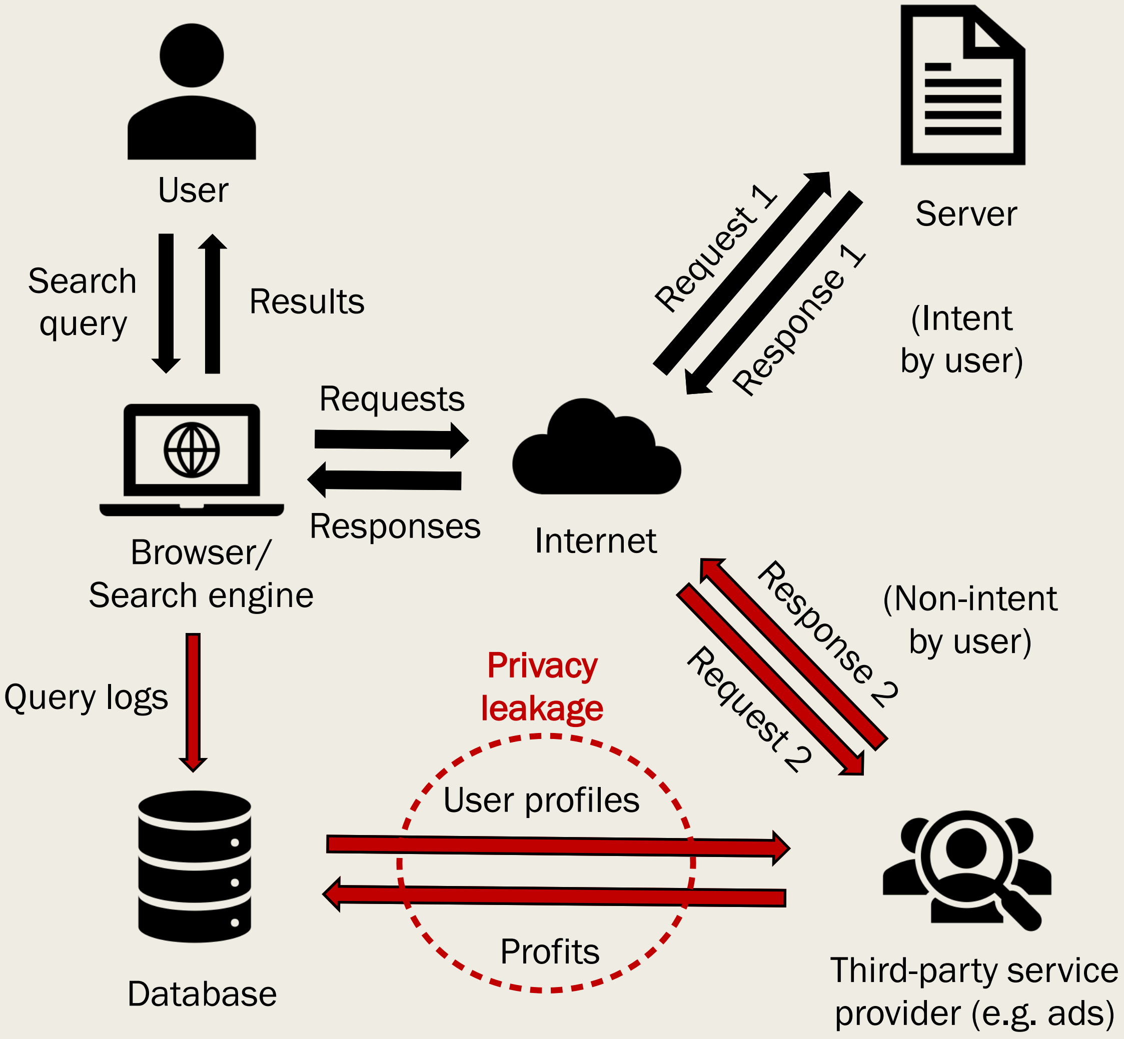


Fig. 1-1. How web query can leak your privacy.

Problem and motivation

The only option we have for query privacy currently ...

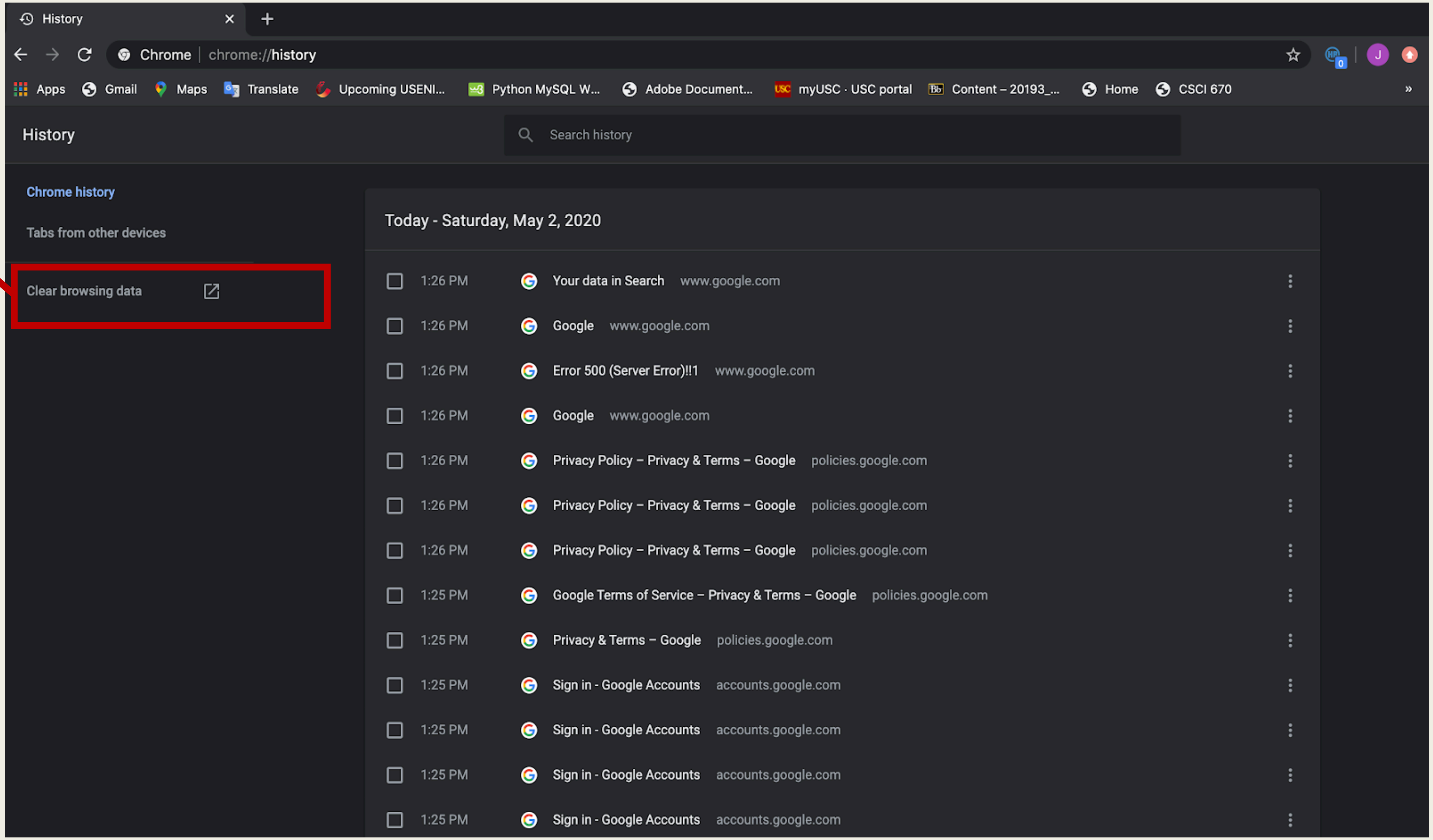
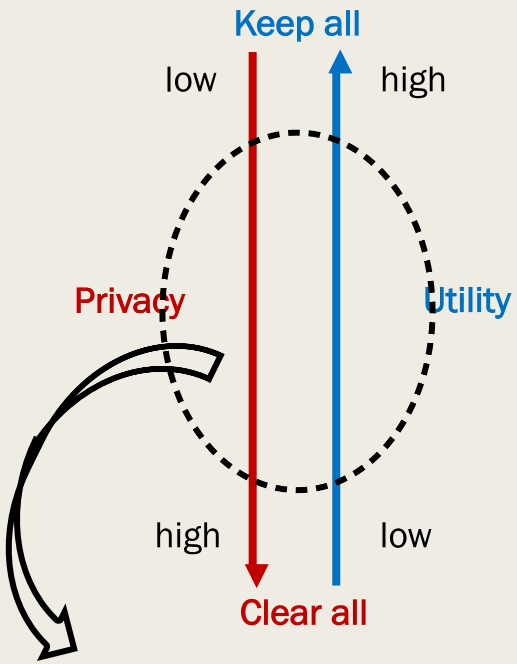


Fig. 1-2. Screenshot of Chrome.

Problem and motivation

The only option we have for query privacy currently ...



Can we have options in the middle?

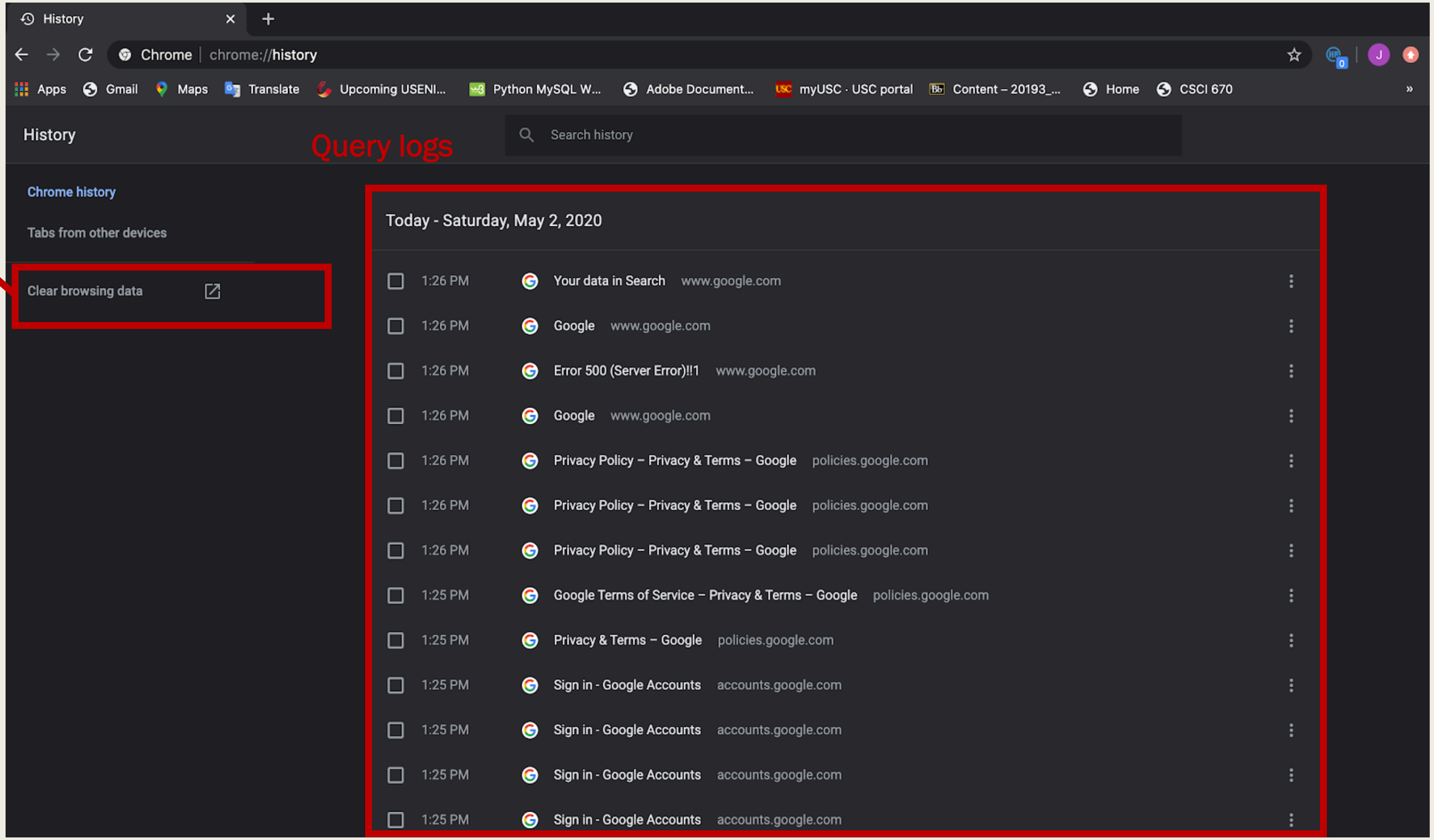


Fig. 1-2. Screenshot of Chrome.

Our contributions

- Propose and implement *SaferQ*: a novel approach for obfuscating web user search queries based on Generative Adversary Privacy (GAP).
- Extend existing GAP framework for sequence generation problem, by leveraging multi-objective reinforcement learning (MORL).
- The trade between privacy and utility of obfuscated queries can be achieved flexibly via *SaferQ.*
- Evaluate *SaferQ* on AOL dataset to demonstrate its effectiveness.

Our contributions

- Propose and implement *SaferQ*: a novel approach for obfuscating web user search queries based on Generative Adversary Privacy (GAP).
- Extend existing GAP framework for sequence generation problem, by leveraging multi-objective reinforcement learning (MORL).
- The trade between privacy and utility of obfuscated queries can be achieved flexibly via *SaferQ*.
- Evaluate *SaferQ* on AOL dataset to demonstrate its effectiveness.

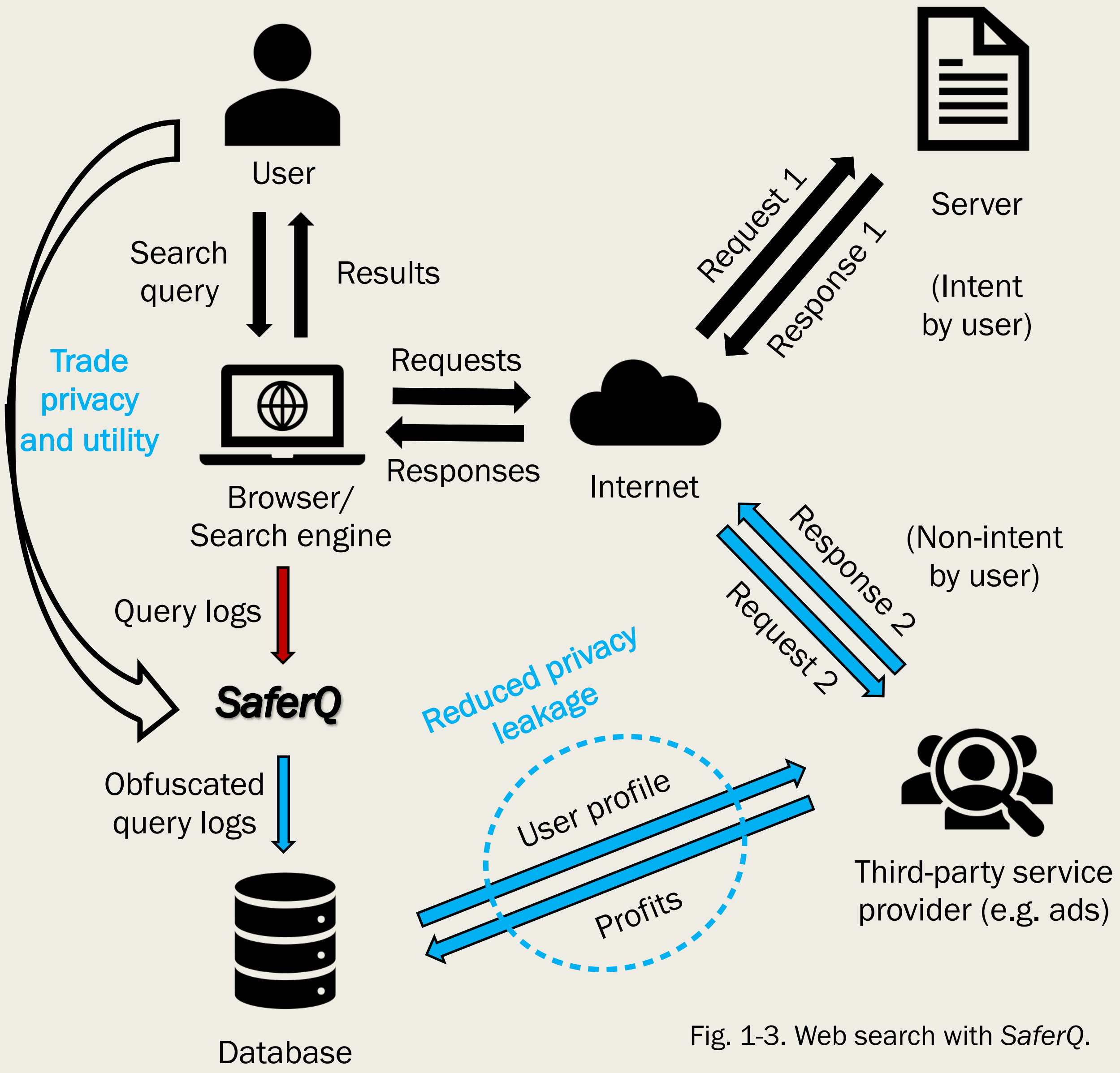


Fig. 1-3. Web search with *SaferQ*.

Our contributions

- Propose and implement *SaferQ*: a novel approach for obfuscating web user search queries based on Generative Adversary Privacy (GAP).
- Extend existing GAP framework for sequence generation problem, by leveraging multi-objective reinforcement learning (MORL).
- The trade between privacy and utility of obfuscated queries can be achieved flexibly via *SaferQ.*
- Evaluate *SaferQ* on AOL dataset to demonstrate its effectiveness.

Examples generated by *SaferQ*

Potential privacy leakage

- Original queries:
  - *["stage cancer", "stage non small cell lung cancer", "cheesecake factory"]*
- Obfuscated queries:
  - *["response silhouette", "response non restrict cell lung silhouette", "golden factory"]*
  - *["baby pregnancy", "stage non small cell lung pregnancy", "cheesecake factory"]*
  - *[" stage cancer", "stage baby small cell lung cancer", "cheesecake snail"]*

Privacy

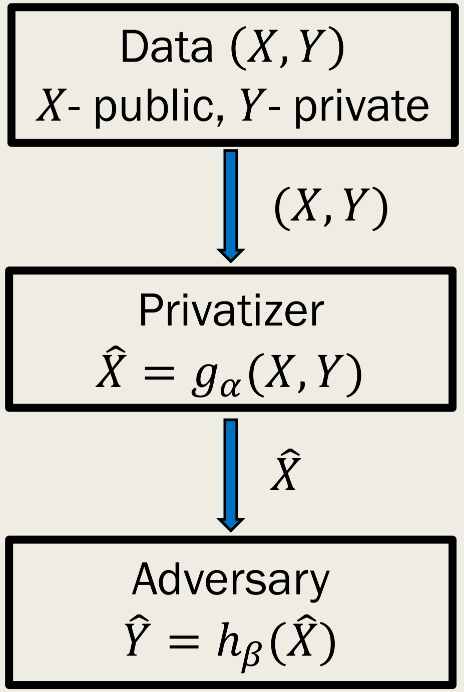
Utility

Content

- Part I: Introduction & Problem
  - *Web query privacy and its challenges*
  - *Problem and motivation*
  - *Our contributions*
- Part II: Approach & Design
  - *Generative adversarial privacy (GAP)*
  - *System architecture*
  - *System optimization*
- Part III: Evaluation & Conclusion
  - *Experiment, results and analysis*
  - *Discussion, conclusion and future works*

Generative adversarial privacy (GAP)

Fig. 2-1. GAP.



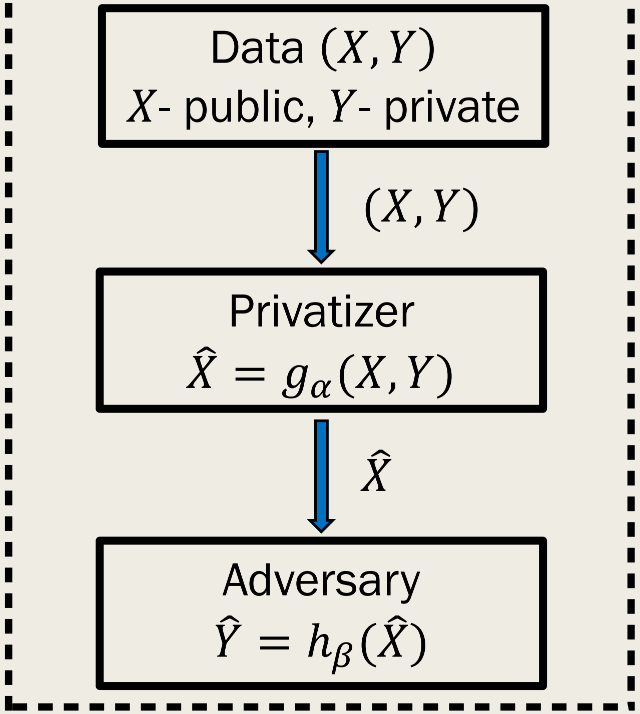
Min-max game

$$\min_{\alpha} \max_{\beta} E[-L(h_\beta(g_\alpha(X,Y)), Y)]$$
$$s.t.\, E[d(g_\alpha(X,Y), X)] \leq D$$

$-L$: privacy loss
d : utility loss
D : utility constraint

Generative adversarial privacy (GAP)

Fig. 2-1. GAP.



Min-max game

$$\min_{\alpha} \max_{\beta} E[-L(h_\beta(g_\alpha(X,Y)), Y)]$$
$$s.t.\, E[d(g_\alpha(X,Y), X)] \le D$$

Training loop of GAP

$-L$: privacy loss
d : utility loss
D : utility constraint

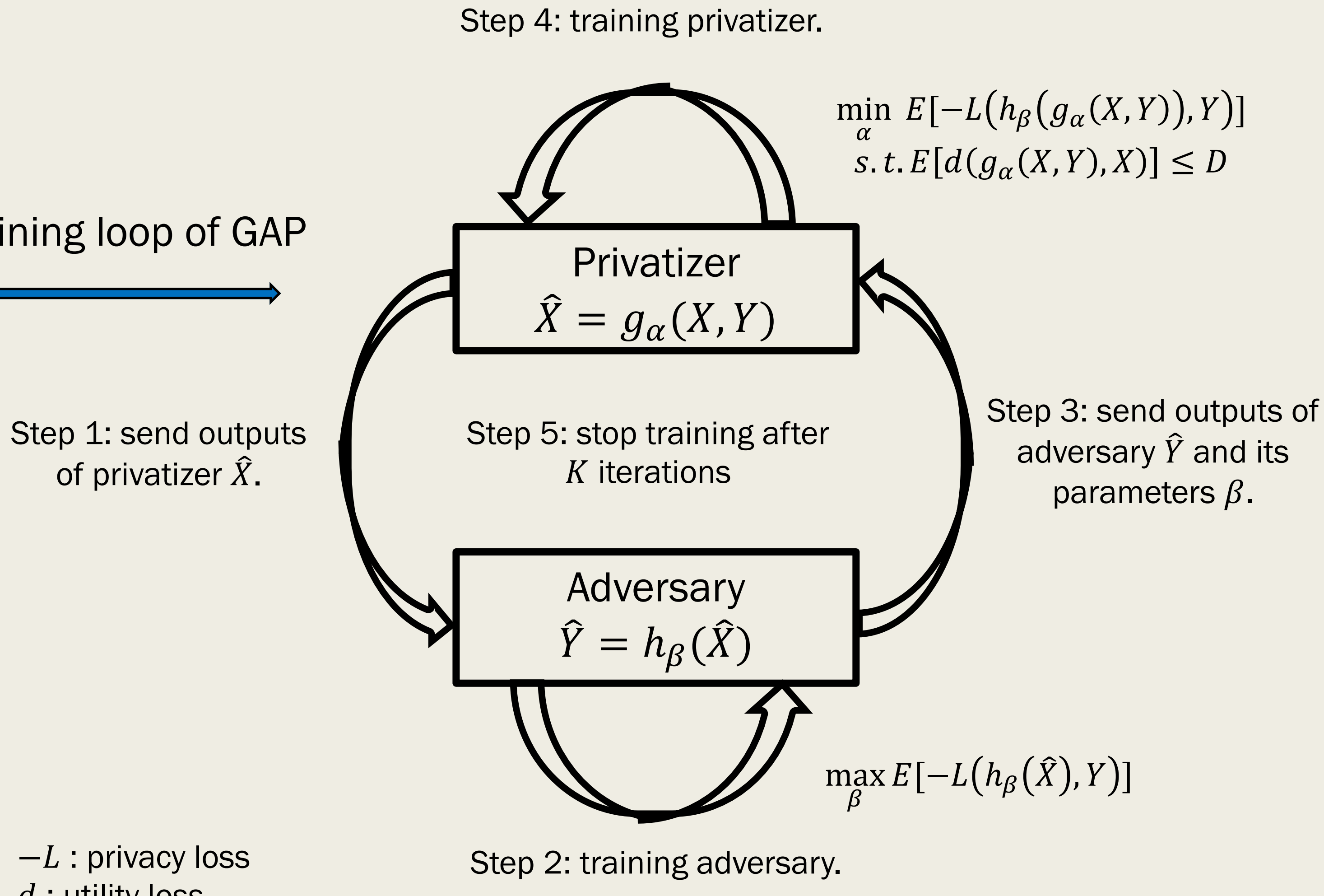
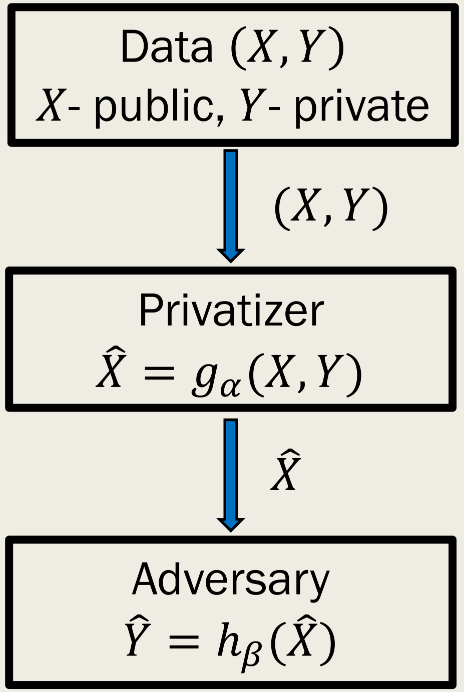


Fig. 2-2. Training loop of GAP.

Generative adversarial privacy (GAP)

Fig. 2-1. GAP.



Min-max game

$$\min_\alpha \max_\beta E[-L(h(g_\alpha(X,Y)), Y)]$$
$$s.t.\ E[d(g_\alpha(X,Y), X)] \le D$$

Extend GAP for sequence data

$(\{X_1, \dots, X_n\}, \{Y_1, \dots, Y_n\})$

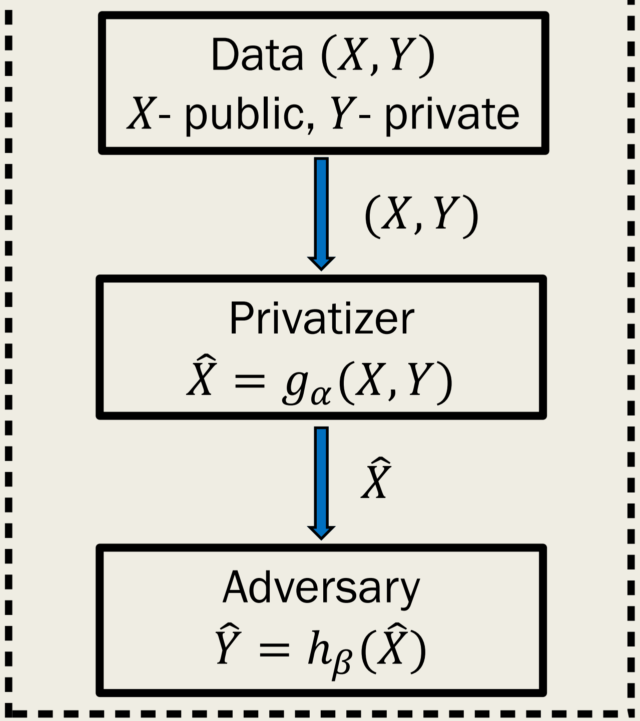
Define:

$$X_{k<t} = \{X_1, \dots, X_{t-1}\}$$
$$\hat{X}_{k<t} = \{\hat{X}_1, \dots, \hat{X}_{t-1}\}$$
$$Y_{k<t} = \{Y_1, \dots, Y_{t-1}\}$$
$$H_{t-1} = (X_{k<t}, \hat{X}_{k<t}, Y_{k<t})$$

$-L$: privacy loss
d : utility loss
D : utility constraint

Generative adversarial privacy (GAP)

Fig. 2-1. GAP.



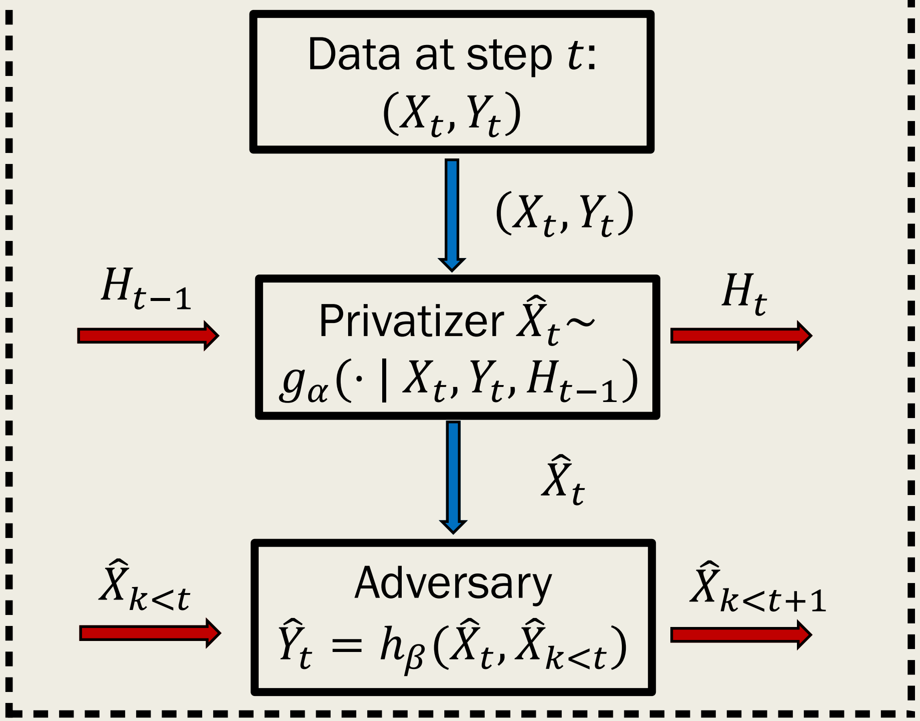
Extend GAP for sequence data

$(\{X_1, \dots, X_n\}, \{Y_1, \dots, Y_n\})$

Define:

$$X_{k<t} = \{X_1, \dots, X_{t-1}\}$$
$$\hat{X}_{k<t} = \{\hat{X}_1, \dots, \hat{X}_{t-1}\}$$
$$Y_{k<t} = \{Y_1, \dots, Y_{t-1}\}$$
$$H_{t-1} = (X_{k<t}, \hat{X}_{k<t}, Y_{k<t})$$

Fig. 2-3. SeqGAP.



Min-max game

$$\min_\alpha \max_\beta E[-L(h(g_\alpha(X, Y)), Y)]$$
$$s.t. E[d(g_\alpha(X, Y), X)] \le D$$

$-L$: privacy loss
d : utility loss
D : utility constraint

Min-max game

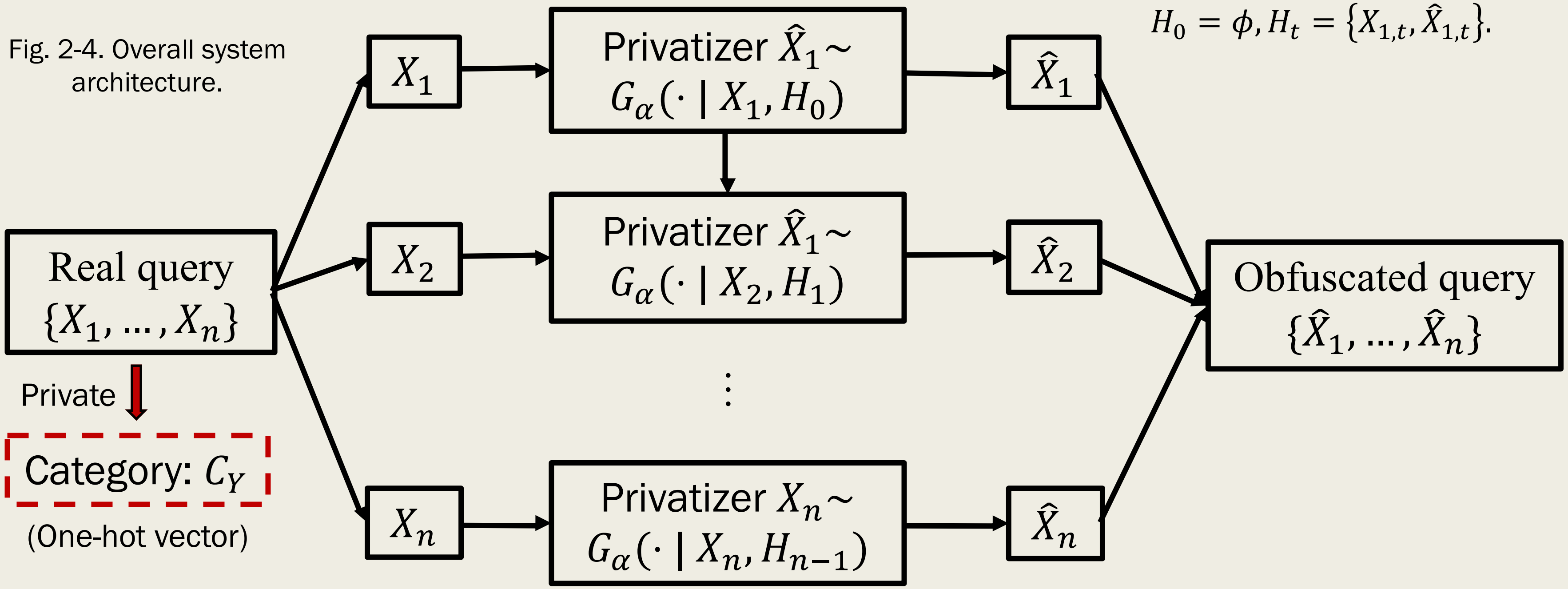
Step n: $\{\hat{X}_1, \dots, \hat{X}_n\}$

$$\min_\alpha \max_\beta E[\sum_{t=1}^{n} -L(h(g_\alpha(X_t, Y_t)), Y_t)]$$
$$s.t. E\left[\sum_{t=1}^{n} d(g_\alpha(X_t, Y_t), X_t)\right] \le D$$

Overall system architecture

$$X_{1,n} = \{X_1, \dots, X_n\} \sim q_{real},$$
$$\hat{X}_{1,n} = \{\hat{X}_1, \dots, \hat{X}_n\} \sim q_{fake} = G_\alpha(\cdot \mid X_{1,n}),$$
$$H_0 = \phi, H_t = \{X_{1,t}, \hat{X}_{1,t}\}.$$

Fig. 2-4. Overall system architecture.



Overall system architecture

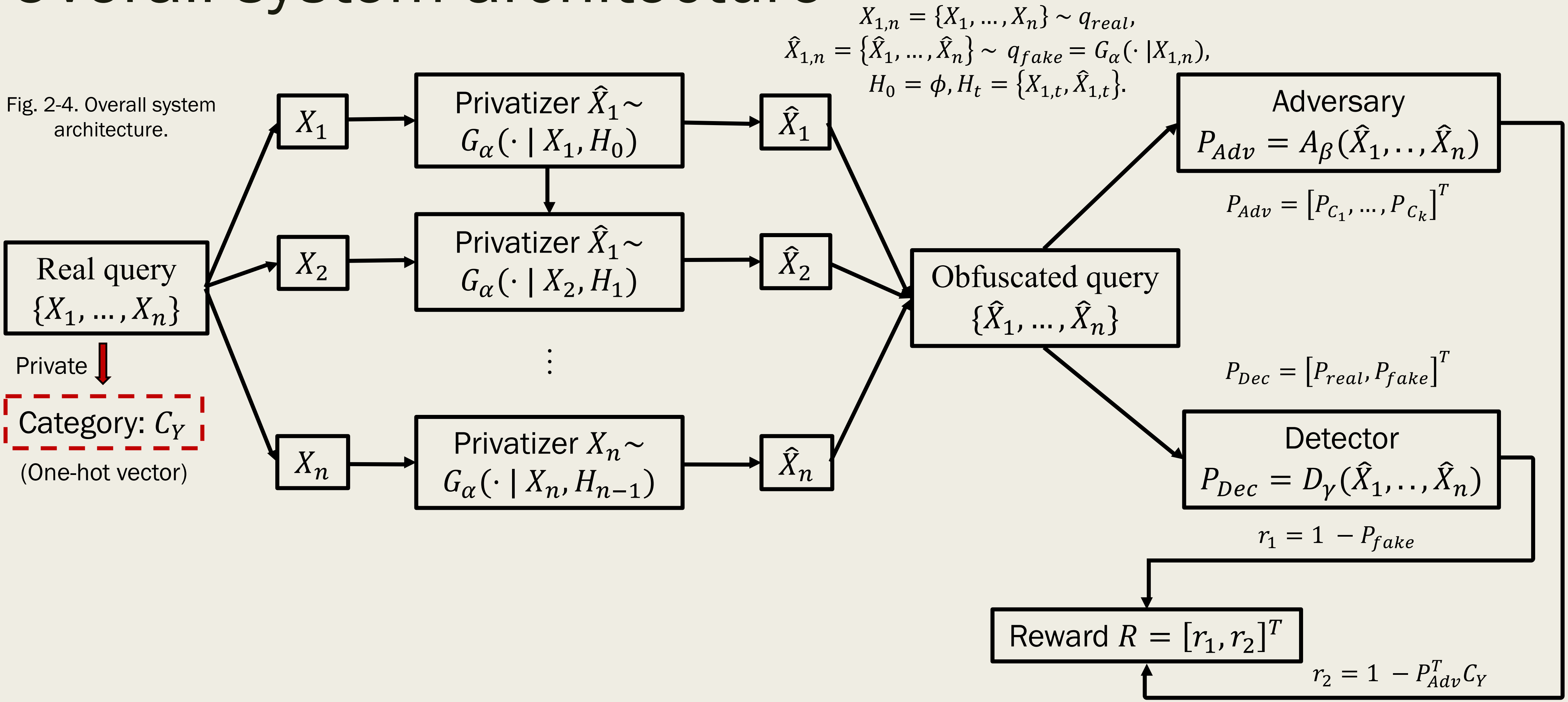


Fig. 2-4. Overall system architecture.

Overall system architecture

$$X_{1,n} = \{X_1, \ldots, X_n\} \sim q_{real},$$
$$\hat{X}_{1,n} = \{\hat{X}_1, \ldots, \hat{X}_n\} \sim q_{fake} = G_\alpha(\cdot \mid X_{1,n}),$$
$$H_0 = \phi, H_t = \{X_{1,t}, \hat{X}_{1,t}\}.$$

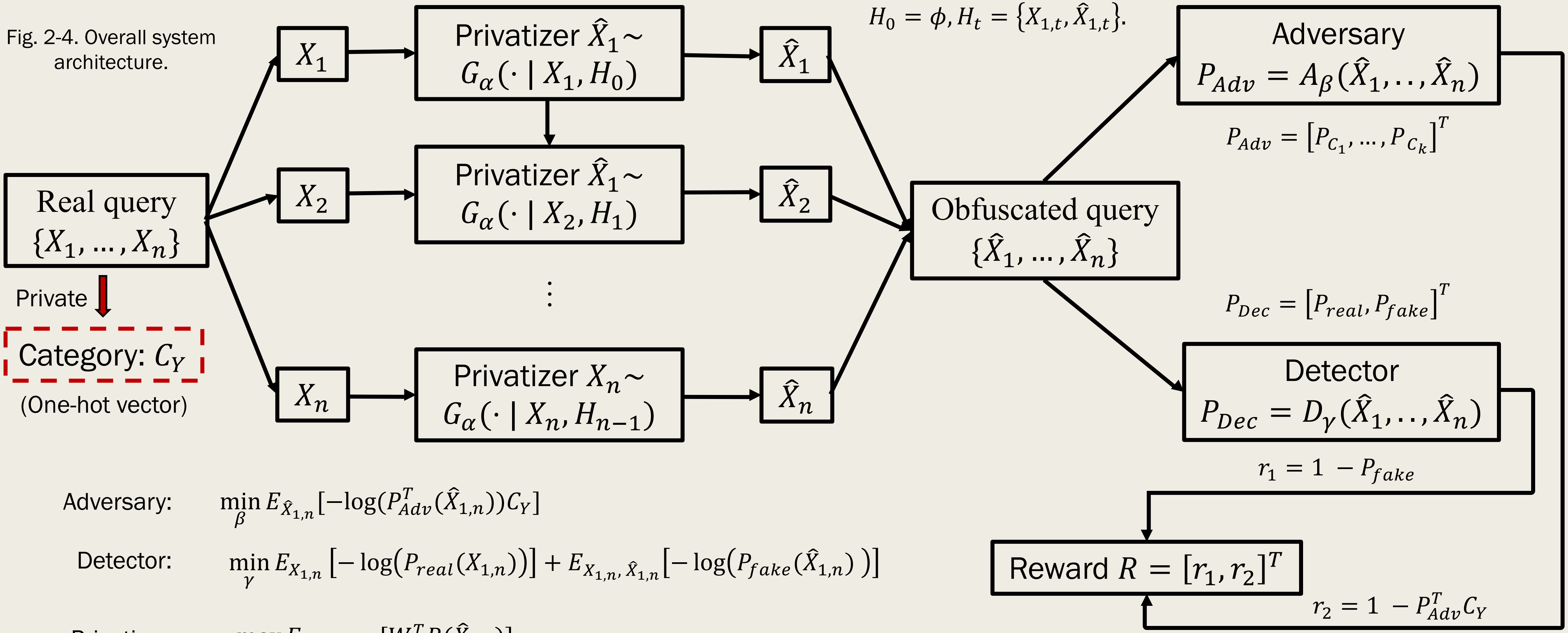


Fig. 2-4. Overall system architecture.

Adversary: $\min_\beta E_{\hat{X}_{1,n}}[-\log(P_{Adv}^T(\hat{X}_{1,n}))C_Y]$

Detector: $\min_\gamma E_{X_{1,n}}[-\log(P_{real}(X_{1,n}))] + E_{X_{1,n}, \hat{X}_{1,n}}[-\log(P_{fake}(\hat{X}_{1,n}))]$

Privatizer: $\max_\alpha E_{X_{1,n}, \hat{X}_{1,n}}[W^T R(\hat{X}_{1,n})]$

$s.t. E_{X_{1,n}, \hat{X}_{1,n}}[CE(X_{1,n}, \hat{X}_{1,n})] \leq D$

$W = [w_1, w_2]^T, CE: Cross\ Entropy$

Overall system architecture

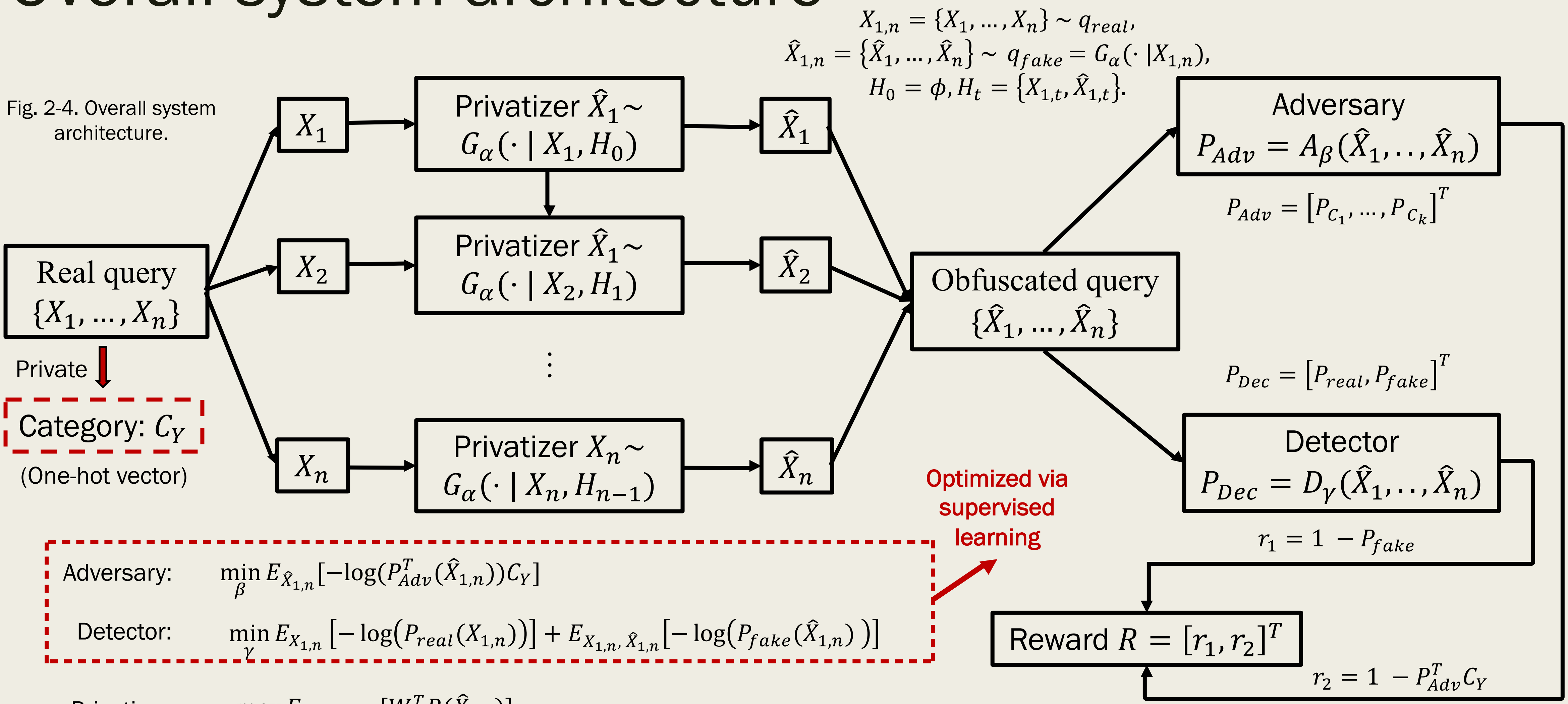


Fig. 2-4. Overall system architecture.

Privatizer: $\max_\alpha E_{X_{1,n}, \hat{X}_{1,n}}[W^T R(\hat{X}_{1,n})]$

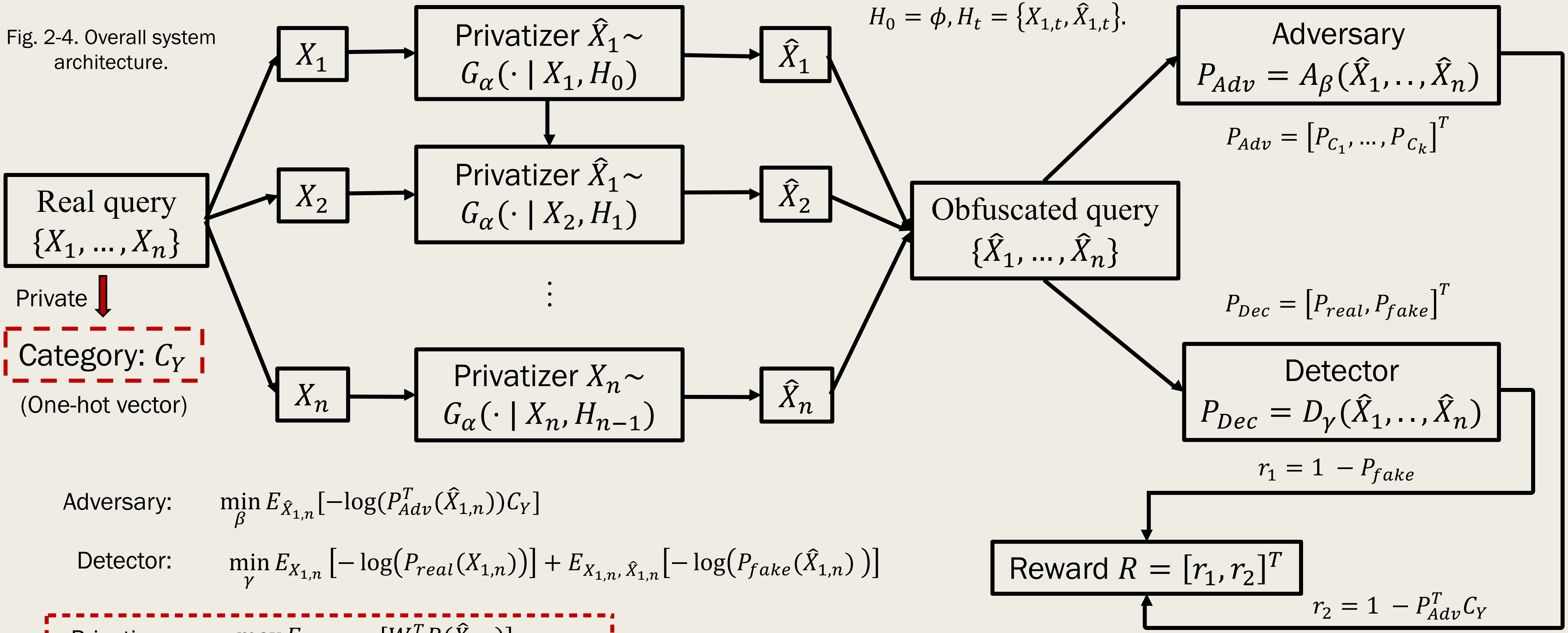
$s.t. E_{X_{1,n}, \hat{X}_{1,n}}[CE(X_{1,n}, \hat{X}_{1,n})] \le D$

$W = [w_1, w_2]^T, CE: Cross\ Entropy$

Overall system architecture

$$X_{1,n} = \{X_1, \dots, X_n\} \sim q_{real},$$
$$\hat{X}_{1,n} = \{\hat{X}_1, \dots, \hat{X}_n\} \sim q_{fake} = G_\alpha(\cdot \mid X_{1,n}),$$
$$H_0 = \phi, H_t = \{X_{1,t}, \hat{X}_{1,t}\}.$$

Fig. 2-4. Overall system architecture.



Adversary: $\min_\beta E_{\hat{X}_{1,n}}[-\log(P_{Adv}^T(\hat{X}_{1,n}))C_Y]$

Detector: $\min_\gamma E_{X_{1,n}}\left[-\log\left(P_{real}(X_{1,n})\right)\right] + E_{X_{1,n},\hat{X}_{1,n}}\left[-\log\left(P_{fake}(\hat{X}_{1,n})\right)\right]$

Privatizer: $\max_\alpha E_{X_{1,n},\hat{X}_{1,n}}[W^T R(\hat{X}_{1,n})]$

$s.t. E_{X_{1,n},\hat{X}_{1,n}}\left[CE\left(X_{1,n}, \hat{X}_{1,n}\right)\right] \le D$

How to solve it?

$W = [w_1, w_2]^T, CE: Cross\ Entropy$

System optimization via MORL

Optimization problem for privatizer:

$W = [w_1, w_2]^T, CE: Cross\ Entropy$

$$\max_{\alpha} E_{X_{1,n}, \hat{X}_{1,n}}\left[W^T R(\hat{X}_{1,n})\right],$$
$$s.t.\ E_{X_{1,n}, \hat{X}_{1,n}}\left[CE\left(X_{1,n}, \hat{X}_{1,n}\right)\right] \le D.$$

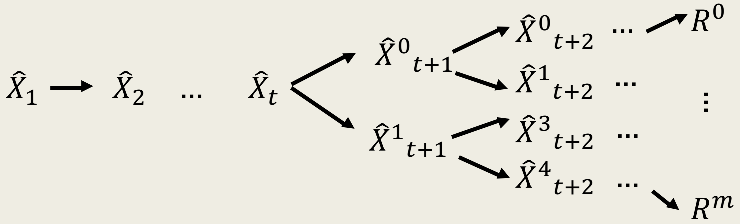
Relaxation

$$J(\alpha) = E_{X_{1,n}, \hat{X}_{1,n}}[W^T R(\hat{X}_{1,n})],\ J_0(\alpha) = E_{X_{1,n}, \hat{X}_{1,n}}\left[CE\left(X_{1,n}, \hat{X}_{1,n}\right)\right],$$
$$\max_{\alpha} J(\alpha) - w_0 J_0(\alpha).$$

Policy gradient theory

$$\nabla_\alpha J(\alpha) = \sum_{t=1}^{n} E_{X_{1,t}, \hat{X}_{1,t-1}}\left[\sum_{\hat{X}_t} \nabla_\alpha g_\alpha(\hat{X}_t | X_t, H_{t-1}) W^T Q\left(s = \{X_t, H_{t-1}\}, a = \hat{X}_t\right)\right],$$
$$Q\left(s = \{X_t, H_{t-1}\}, a = \hat{X}_t\right) = \underbrace{E_{X_{t+1,n}, \hat{X}_{t+1,n}}\left[R\left(\hat{X}_{1,n}\right) | s, a\right]}_{\text{Estimated by Monte Carlo search}}.$$

Estimated by **Monte Carlo search**



$$\hat{Q}(s, a) = \frac{1}{M} \sum_{m=1}^{M} R^m$$

- Why we need RL?
  - *At step t, the output of privatizer ($\hat{X}_t$) will affect the input of privatizer (X_{t+1}, H_t) at step $t+1$.*
  - *The generated sequence may not be differentiable (e.g. NLP tasks).*
- Why we need MORL?
  - *We have both **privacy reward** and **utility reward**, where a **trade-off** exists.*

Define:

$$Q(s, a) = [Q_1(s, a), Q_2(s, a)]^T,$$
$$J(\alpha) = W^T [J_1(\alpha), J_2(\alpha)]^T,$$
$$W' = [w_0, w_1, w_2]^T,$$
$$\nabla_\alpha J' = [-\nabla_\alpha J_0(\alpha), \nabla_\alpha J_1(\alpha), \nabla_\alpha J_2(\alpha)]^T$$

w_0: control CE loss
w_1: control r_1
w_2: control r_2

W' for trade-off

$$\boxed{\boldsymbol{\alpha = \alpha + lr * W'^T \nabla_\alpha J'}}$$

Deep neural network architecture

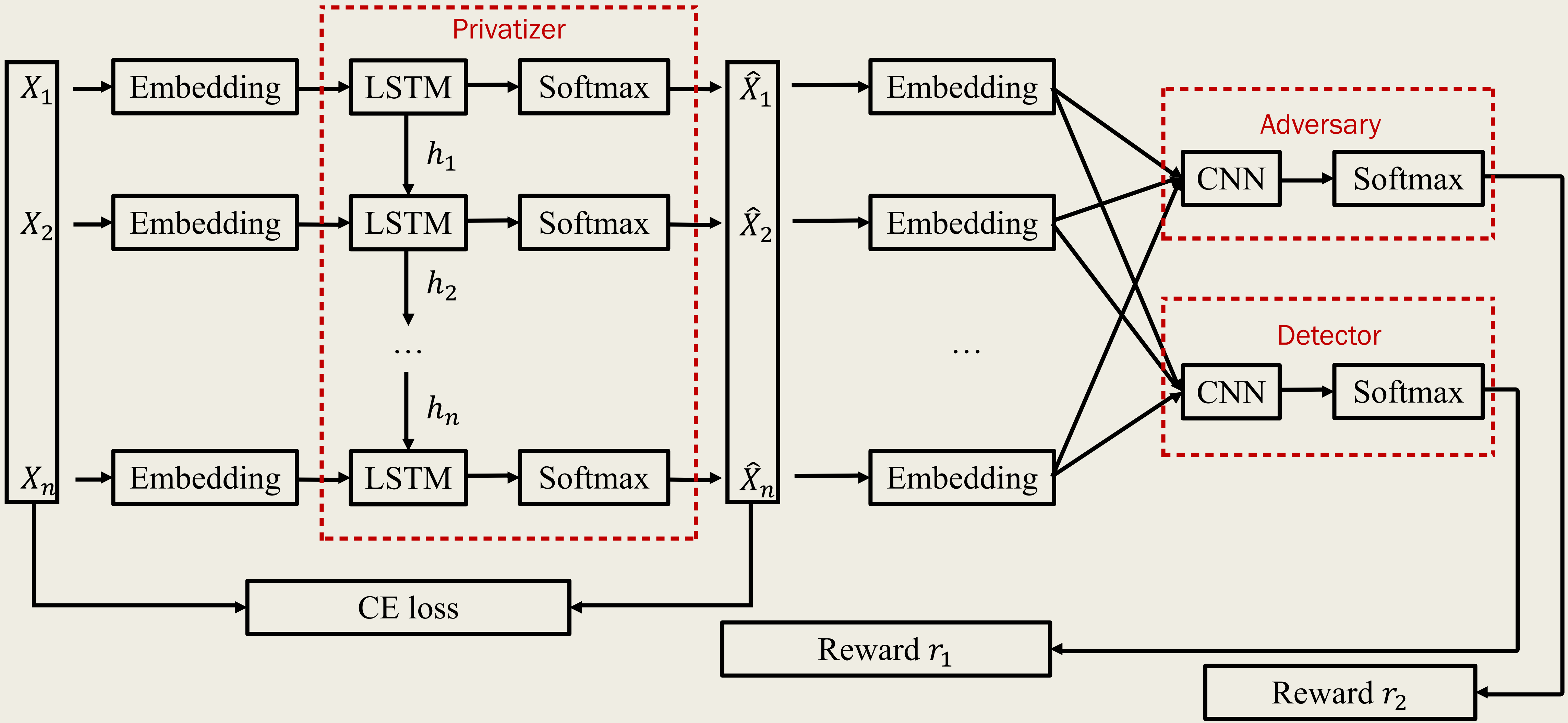


Fig. 2-5. Deep neural network architecture.

Content

Experiment with AOL Dataset

UserID	Query	QueryTime	Rank of link	Link
81943	Are people who have asthma prone to get lung cancer	3/7/06 23:26	2	http://kidshealth.org
81943	If you have asthma can it lead to lung cancer	3/7/06 35	6	http://www.lungusa.org

Tab. 3-1. AOL dataset samples.

- Preprocessing:
  - *Filter the dataset by keywords from two topic: **cancer, pregnancy**.*
  - *Classify the dataset into three categories: cancer related, pregnancy related, and other.*
  - *Each category contains **4,000 query sequences** generated by user in one day.*
- Goal:
  - *Utility I (r_0): reduce the **divergence** between **real queries** and **obfuscated queries**.*
  - *Utility II (r_1): prevent detector from **distinguishing obfuscated queries**.*
  - *Privacy (r_2): prevent adversary from **inferring category** from obfuscated queries.*

Experiment with AOL Dataset

Specifically, suppose real query is $\{X_1, \dots, X_n\}$, obfuscated query is $\{\hat{X}_1, \dots, \hat{X}_n\}$, and its category: C_Y.

Taking $\{\hat{X}_1, \dots, \hat{X}_n\}$ as input, the adversary and detector will output $P_{Adv} = [P_{C_1}, P_{C_2}, P_{C_3}]^T$ and $P_{Dec} = [P_{real}, P_{fake}]^T$ respectively.

$$r_0 = CE(\{X_1, \dots, X_n\}, \{\hat{X}_1, \dots, \hat{X}_n\}),$$
Smaller ⟶ higher *utility I*

$$r_1 = 1 - P_{fake},$$
Larger ⟶ higher *utility II*

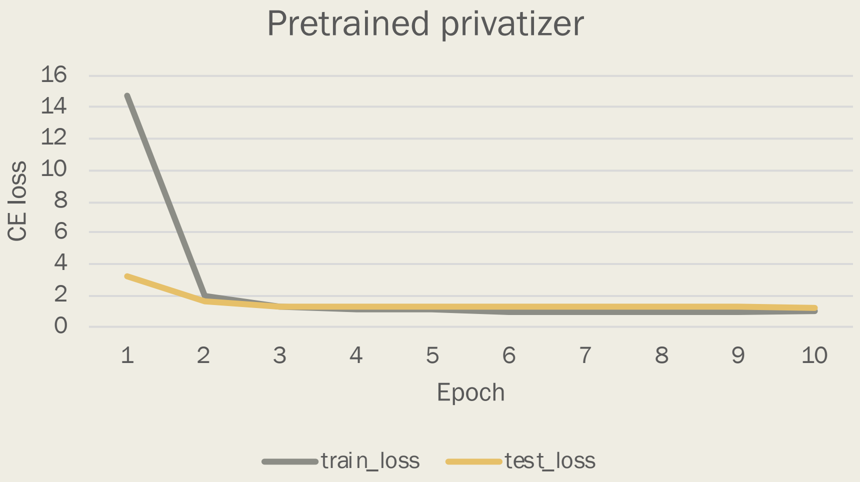
$$r_2 = 1 - P_{Adv}^T C_Y.$$
Larger ⟶ higher *privacy*

C_1: cancer
C_2: pregnancy
C_3: other

- Goal:
  - *Utility I (r_0): reduce the **divergence** between **real queries** and **obfuscated queries.***
  - *Utility II (r_1): prevent detector from **distinguishing obfuscated queries.***
  - *Privacy (r_2): prevent adversary from **inferring category** from obfuscated queries.*

Note: We will call $\boldsymbol{r_0}$ as CE loss, $\boldsymbol{r_1}$ as utility reward, $\boldsymbol{r_2}$ as privacy reward, for convenience.

Learning curve of *SaferQ*



- Analysis
  - *Pretrain privatizer by minimizing CE loss (~ Identity translation).*
  - $q_{real} \approx G_\alpha(\cdot\,|q_{real})$, *initially.*

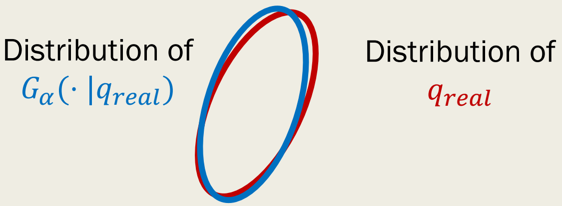
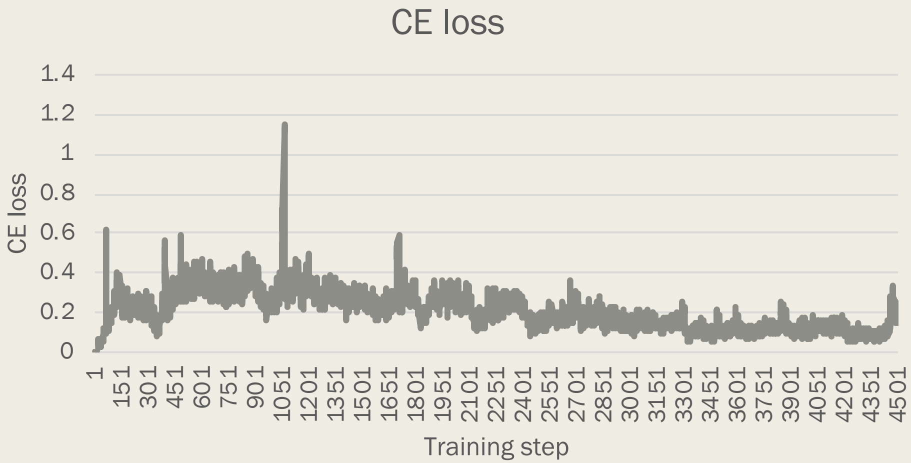
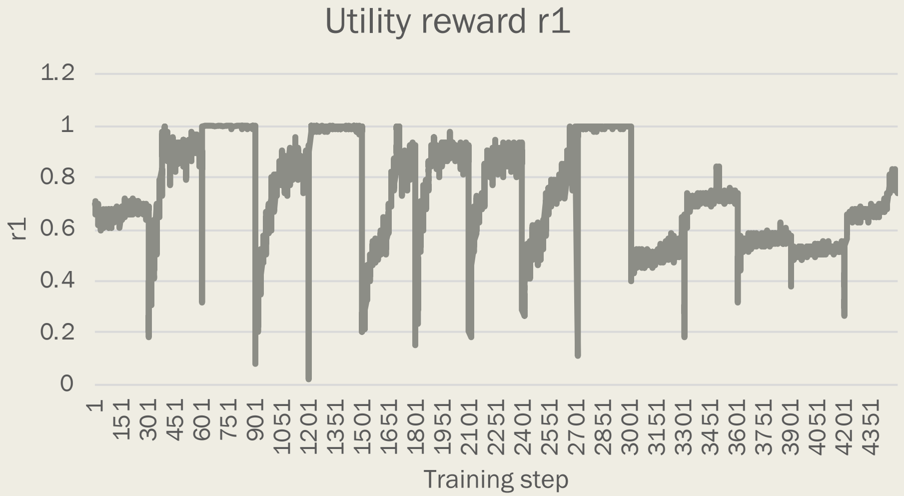


Fig. 3-1. Learning curve of *SaferQ*, W'=[0.02,0.5,0.5].

Learning curve of *SaferQ*



- Training settings
  - *Each epoch contains 150 steps.*
  - *Generator, adversary, and detector will be* ***trained alternatively*** *every* ***two epochs.***



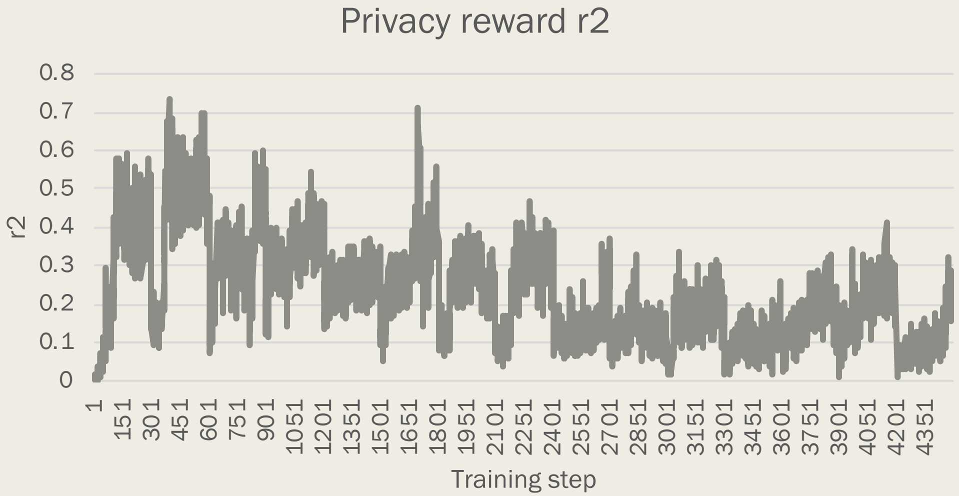
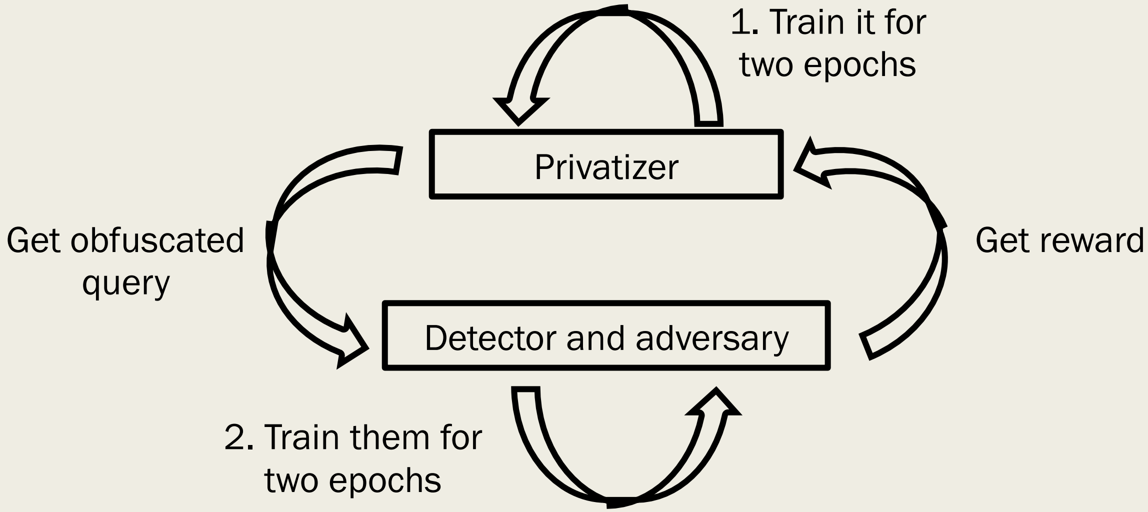
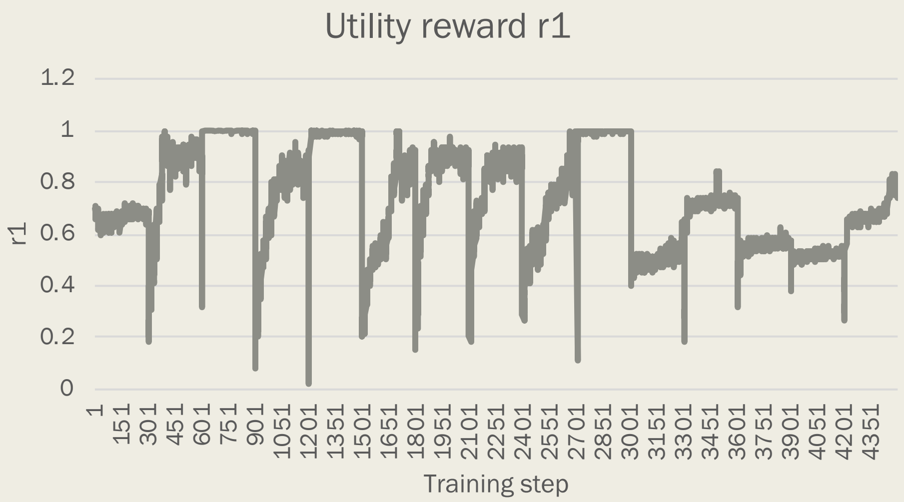


Fig. 3-1. Learning curve of *SaferQ*, W'=[0.02,0.5,0.5].

Learning curve of *SaferQ*

- Analysis
  - *Every two epochs, the discriminator and adversary will be enhanced, thus the privatizer has a drop in reward.*
  - *Dynamic environment in RL (reward metric is changing; become harder every two epochs).*





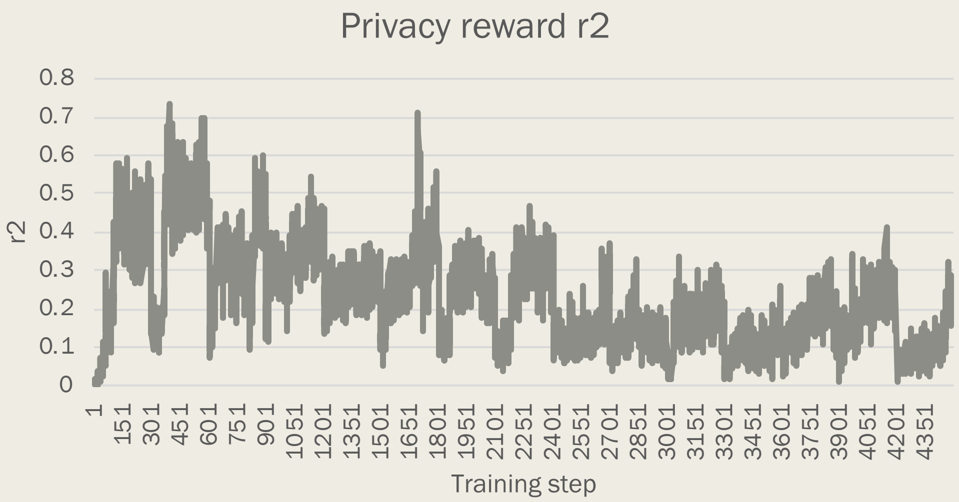
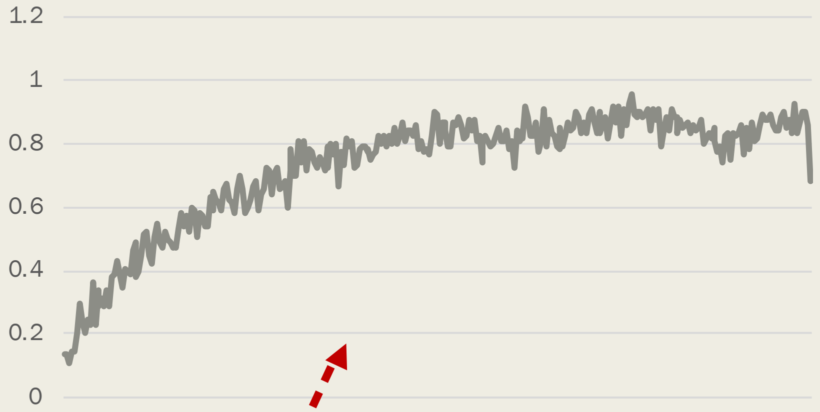
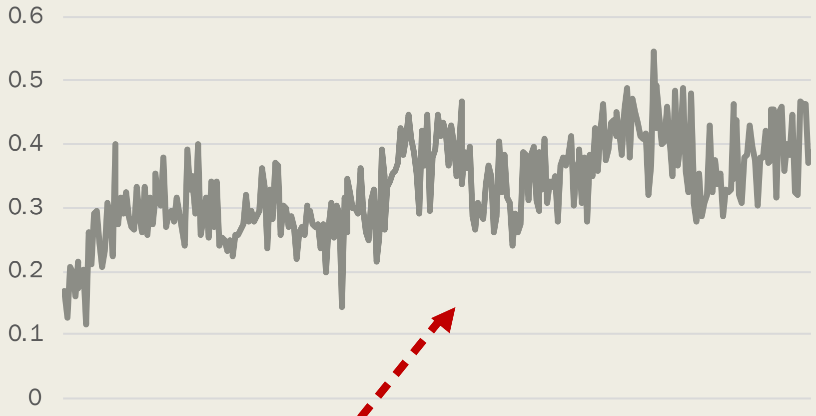


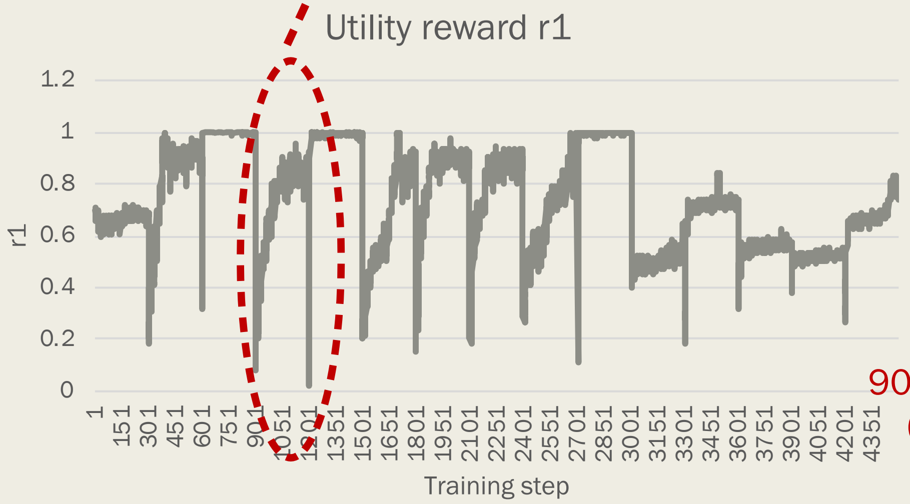
Fig. 3-1. Learning curve of *SaferQ*, W'=[0.02,0.5,0.5].

Learning curve of *SaferQ*





In every two epochs, r_1 and r_2 will increase significantly.



900-1200 steps
(epoch 7-8)

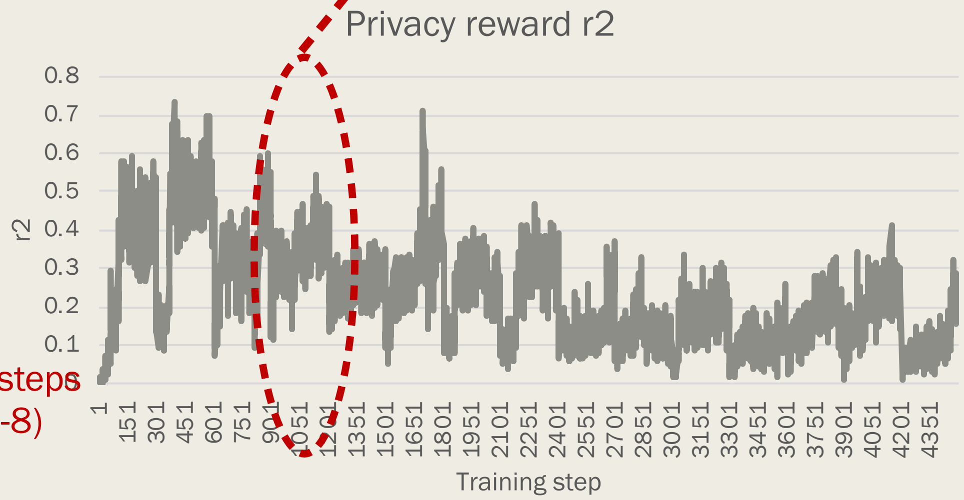
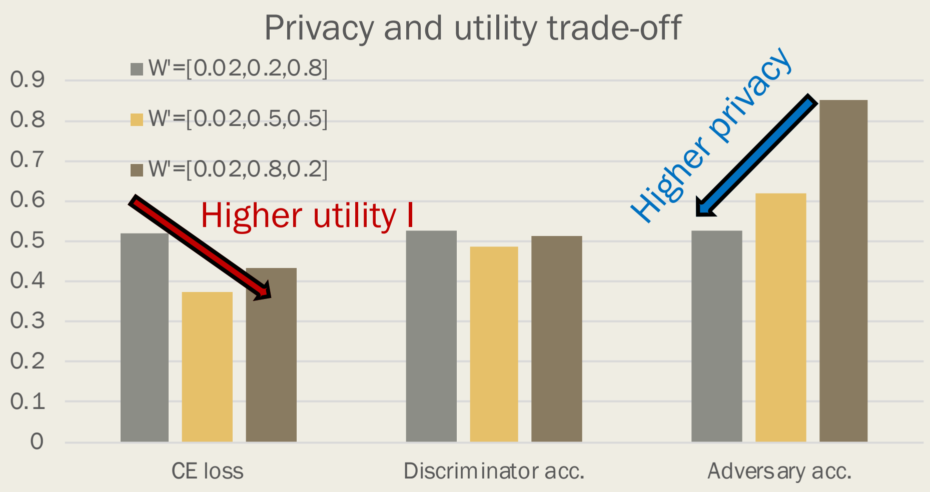
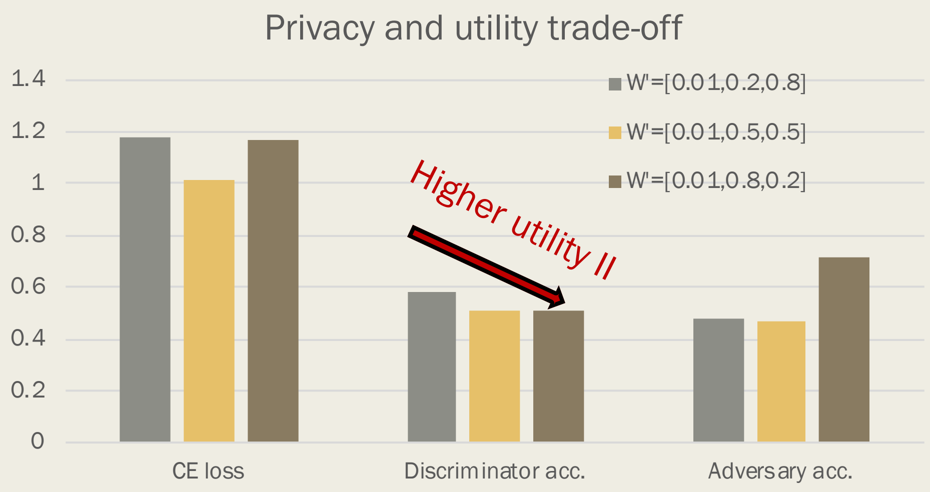


Fig. 3-1. Learning curve of *SaferQ*, W'=[0.02,0.5,0.5].

Privacy and utility trade-off of *SaferQ*





- Analysis
  - *Increase W_0 will make the CE loss smaller.*
  - *Increase W_2/W_1 will decrease the accuracy of adversary for private category inference, which means higher privacy, while the accuracy of discriminator will increase, and CE loss will increase.*
  - *Based on the theory of GAN, if both the privatizer and the discriminator are well trained, the accuracy of discriminator should converge to 0.5.*

More examples

Good

[INFO] Target query: , 'negative', 'pregnancy', 'test', 'result', 'plan', 'plan', 'parenthood', 're', 'be', 'the', 'uterus', 'be', 'be', 'the', 'uterus', 'false', 'negative', 'pregnancy', 'result', 'false', 'negative', 'pregnancy', 'result', 'story', '<EOS>', '*'] target category: 1

W=[0.02,0.8,0.2]

[INFO] Predicted query: ['<SOS>', 'birthday', 'pregnancy', 'test', 'result', 'plan', 'plan', 'on', 're', 'be', 'the', 'uterus', 'be', 'be', 'the', 'uterus', 'ghost', 'birthday', 'pregnancy', 'result', 'ghost', 'negative', 'you', 'result', 'story', '<EOS>', '*'] pred category: 1

W=[0.02,0.5,0.5]

[INFO] Predicted query: ['<SOS>', 'negative', 'pregnancy', 'test', 'result', 'plan', 'plan', 'parenthood', 're', 'be', 'clothes', 'uterus', 'be', 'be', 'the', 'uterus', 'false', 'negative', 'pregnancy', 'result', 'false', 'negative', 'pregnancy', 'result', 'story', '<EOS>', '*'] pred category: 1

W=[0.02,0.2,0.8]

[INFO] Predicted query: ['<SOS>', 'negative', 'skin', 'test', 'creen', 'plan', 'plan', 'parenthood', 're', 'be', 'the', 'uterus', 'be', 'cleaner', 'the', 'uterus', 'false', 'negative', 'effluvium', 'creen', 'false', 'negative', 'cancer', 'result', 'story', '<EOS>', '*'] pred category: 2

Bad

[INFO] Target query: , ['<SOS>', 'how', 'to', 'tell', 'the', 'sex', 'of', 'your', 'baby', 'birth', 'chart', 'how', 'to', 'tell', 'the', 'sex', 'of', 'your', 'baby', '<EOS>', '*'] target category: 1

W=[0.02,0.8,0.2]

[INFO] Predicted query: ['<SOS>', 'how', 'to', 'tell', 'the', 'sex', 'of', 'your', 'baby', 'birth', 'chart', 'how', 'to', 'tell', 'the', 'sex', 'of', 'your', 'baby', '<EOS>', '*'] pred category: 1

W=[0.02,0.5,0.5]

[INFO] Predicted query: ['<SOS>', 'how', 'to', 'tell', 'clothes', 'sex', 'of', 'your', 'w', 'birth', 'tumor', 'how', 'w', 'tell', 'clothes', 'sex', 'of', 'your', 'w', '<EOS>', '*'] pred category: 0

W=[0.02,0.2,0.8]

[INFO] Predicted query: ['<SOS>', 'how', 'to', 'tell', 'the', 'sex', 'of', 'your', 'baby', 'birth', 'chart', 'how', 'to', 'tell', 'the', 'sex', 'of', 'your', 'baby', '<EOS>', '*'] pred category: 1

Conclusion and future works

- We propose *SaferQ* for web user query obfuscation, which is capable of achieving trade-off between privacy and utility.
- We implement *SaferQ* by *PyTorch* and *Ray*, supporting GPU and multi-processing MC sampling.
- In the future, we plan to:
  - *Use pretrained model to improve the performance of SaferQ.*
  - *Utilize larger dataset and more categories.*
  - *Leverage federated learning to train SaferQ.*
  - *Reduce the sampling complexity for Q-value estimation.*

Thanks for watching!

Any questions?

Supplemental materials

Related works

- Baseline solution:
  - *Delete words with high privacy risk (e.g. cancer, pregnancy)*
  - *Mutate words randomly*
  - *Mutate words with differential privacy*
  - *Define some simple privacy risk metrics and reduce them (e.g. incognito)*
- Limitations:
  - *May not capture the correlation among words in the query*
  - *Context-free, no adversary*

Network architecture v2 ($O(T)$ when sampling)

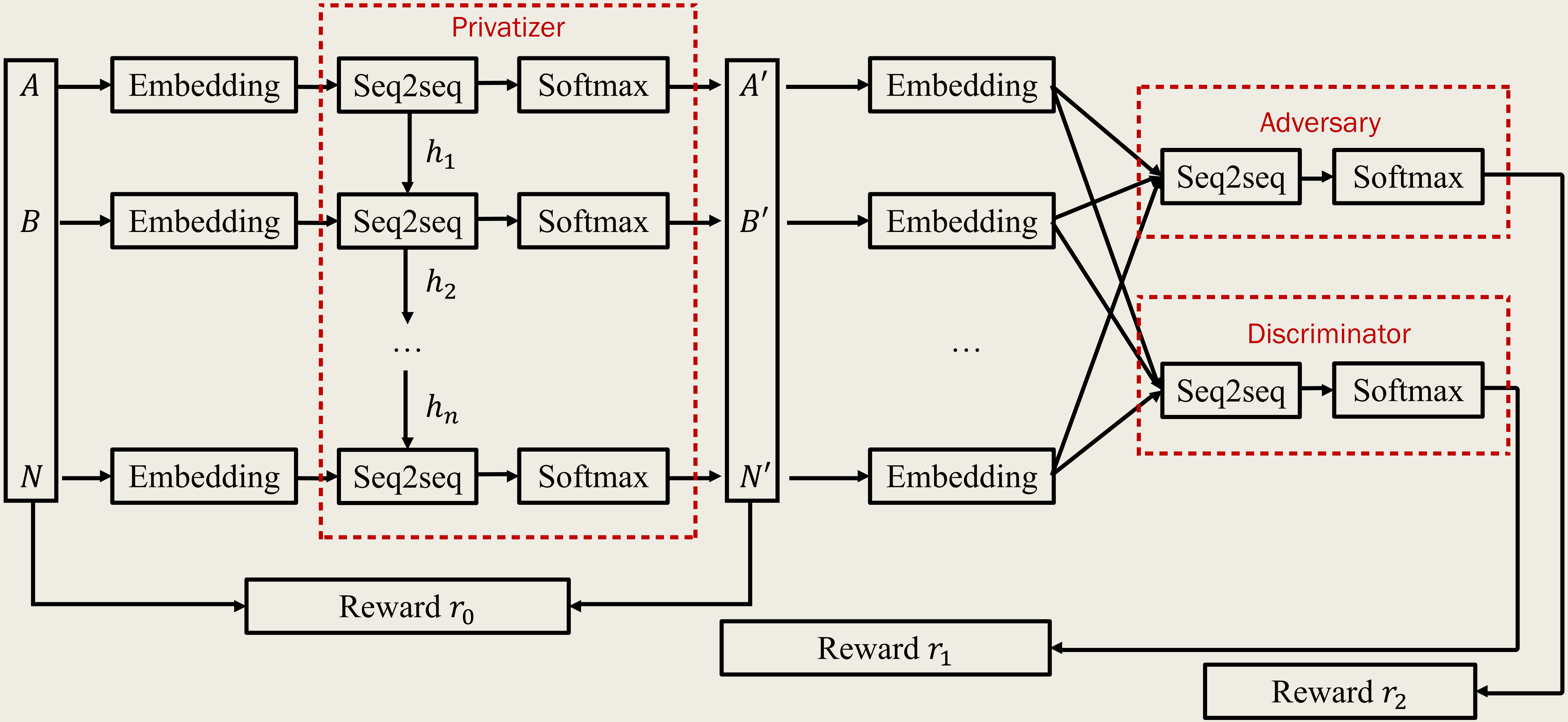


Fig. 2-3. Network architecture v1.

36