Knowledge-enhanced Black-box Attacks for Recommendations

Jingfan Chen[1], Wenqi Fan[2], Guanghui Zhu[1], Xiangyu Zhao[3], Chunfeng Yuan[1], Qing Li[2], Yihua Huang[1]

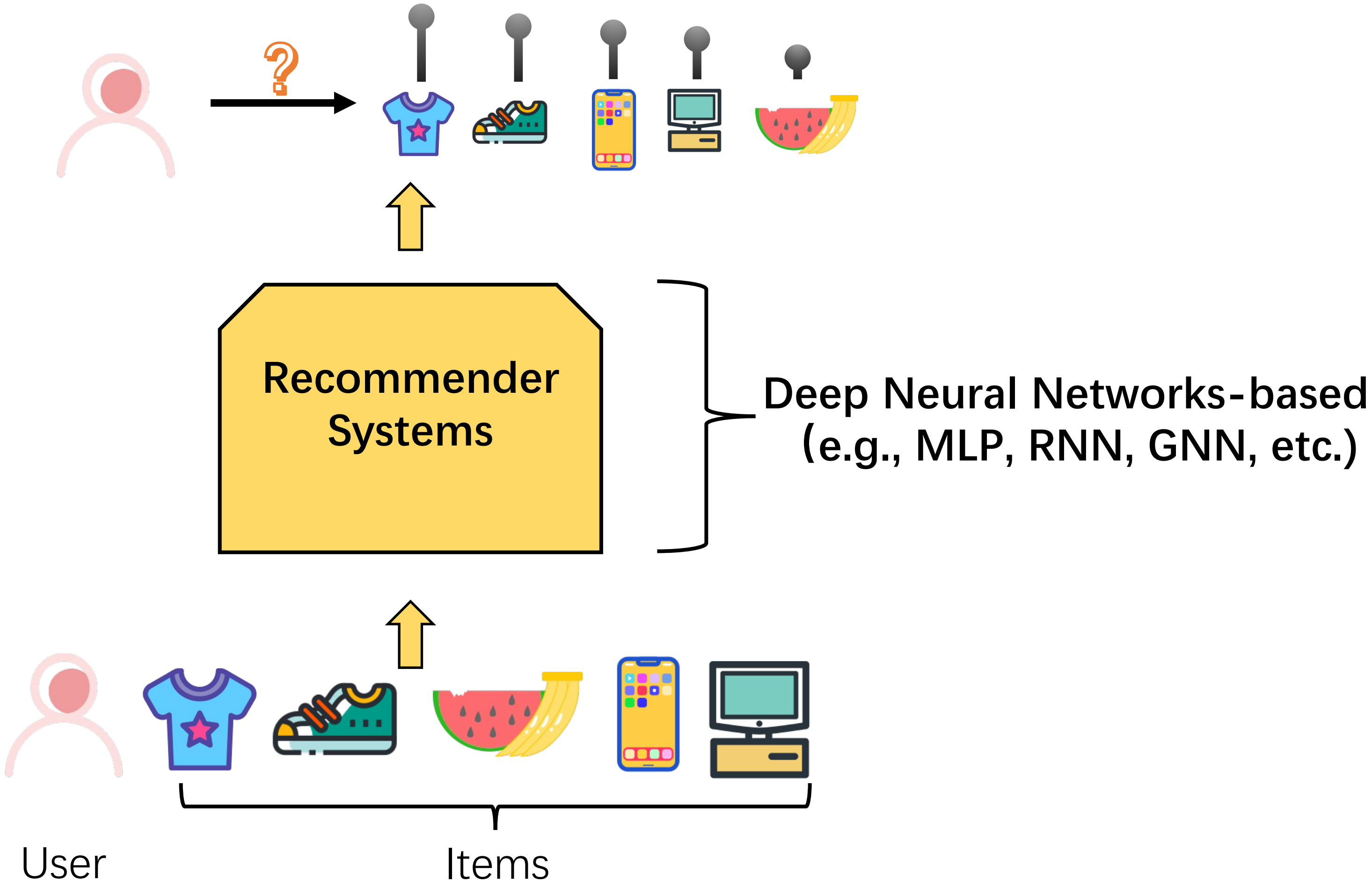
[1]Nanjing University

[2]The Hong Kong Polytechnic University

[3]City University of Hong Kong

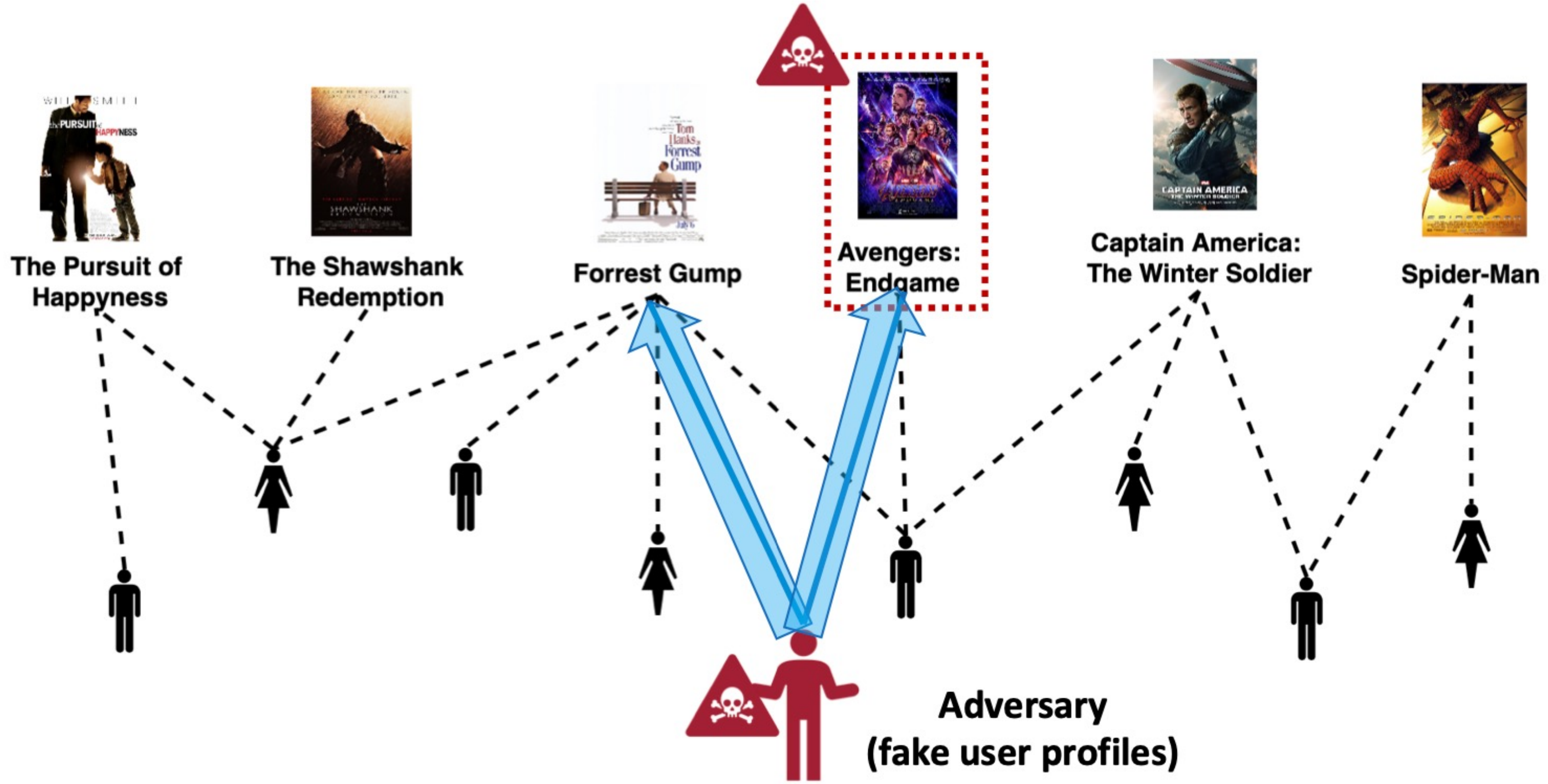
Background

- Deep Recommender Systems
  - Goal: provide a personalized ranked list of items to users



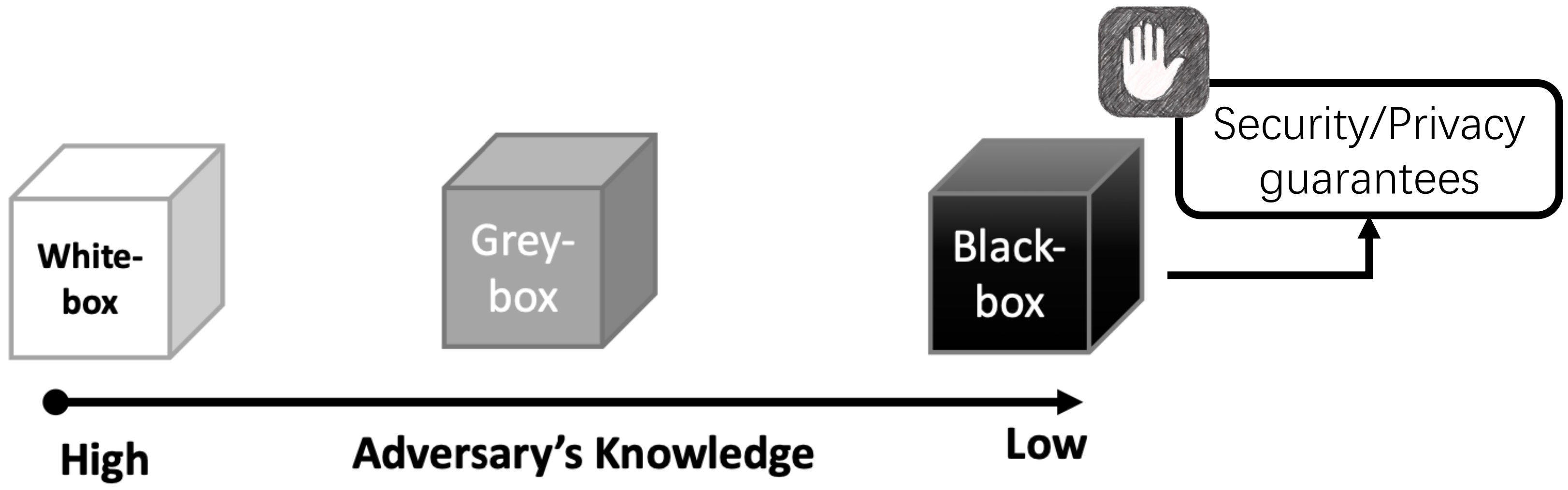
Background

- Attacks in Recommender Systems
  - Data Poisoning Attacks: promote/demote a set of item



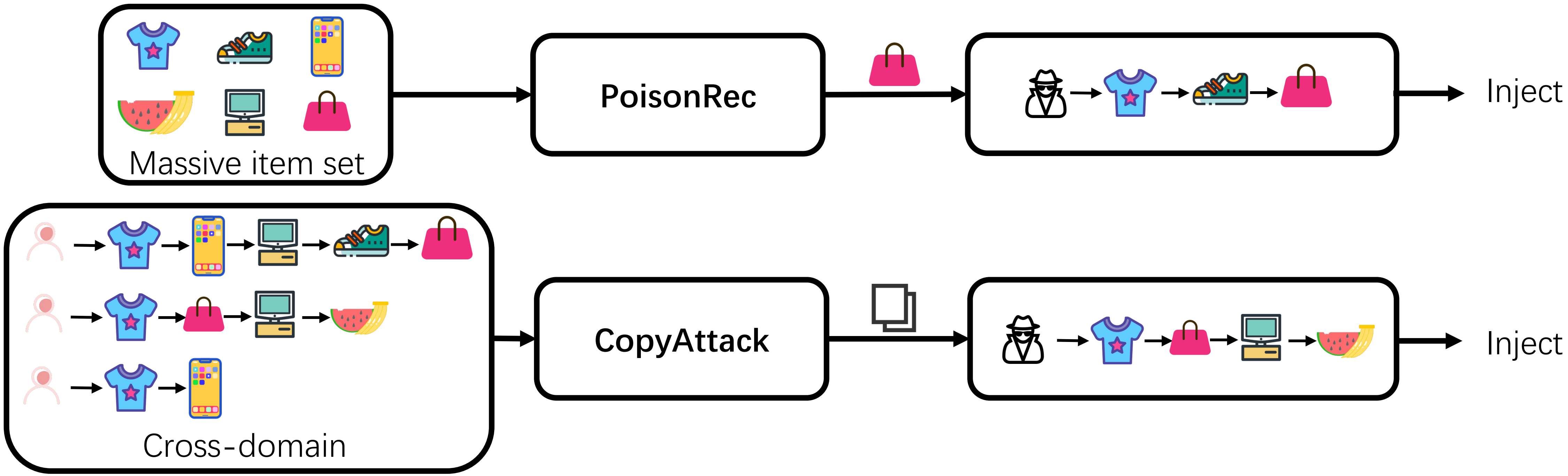
Background

- Black-box attacks vs. White/grey-box attacks
  - **No knowledge** vs. **full/partial knowledge**
  - Practical (privacy and security concerns)



Background

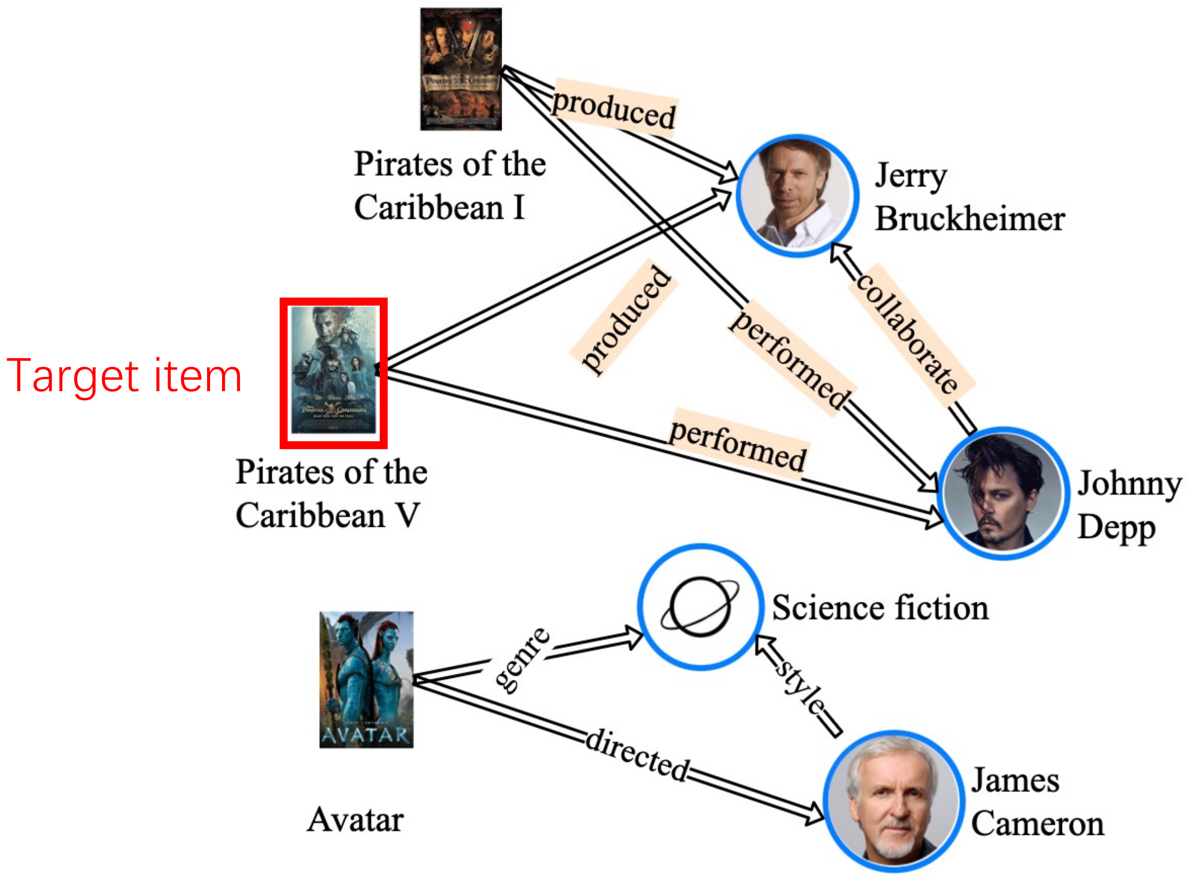
- Challenges in existing black-box attacking methods
  - PoisonRec[1]: massive item sets
  - CopyAttack[2]: lack of cross-domain knowledge



[1] An Adaptive Data Poisoning Framework for Attacking Black-box Recommender Systems (ICDE20)
[2] Attacking Black-box Recommendations via Copying Cross-domain User Profiles (ICDE21)

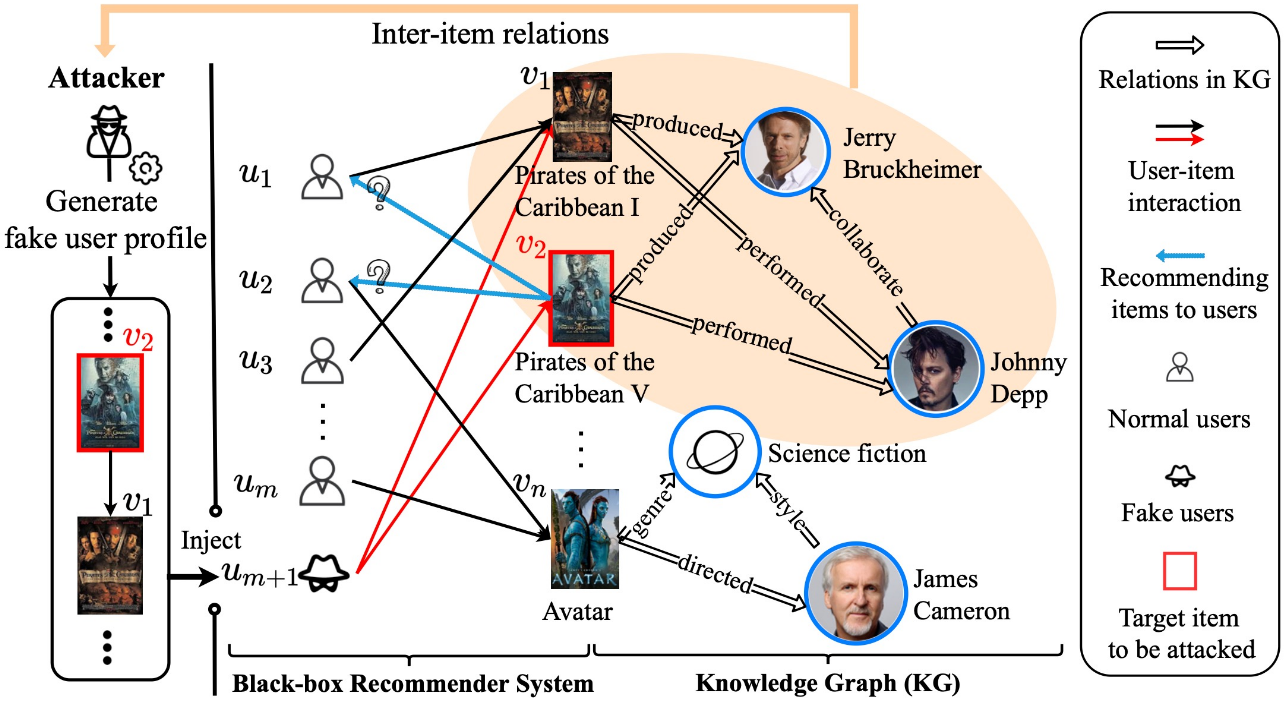
Background

- Side-information: Knowledge Graph (KG)
  - Rich auxiliary knowledge: relations among items and real-world entities
  - The underlying relationships between Target items and other items



Motivation

- Challenges in existing black-box attacking methods
  - PoisonRec[1]: massive item sets
  - CopyAttack[2]: lack of cross-domain knowledge
- Employs the KG to enhance the generation of fake user profiles from the massive item sets



Background

- **Problem Statement**
  - User $U = \{u_1, \cdots, u_m\}$
  - Item $V = \{v_1, \cdots, v_n\}$
  - User-item Interactions Y
  - KG $\mathcal{G} = \{\mathcal{V}, \mathcal{R}\}$, entity−relation−entity triples (p,r,q)
    - E.g., (Avatar, film.director, James Cameron)
- **Goal**: promote a target item $v^* \in V$
- **Method**: Inject fake user profiles $P_t = \{v_0, \cdots, v_{t-1}\}$
  - $U' = U \cup U^F$ where $U^F = \{u_{m+i}\}_{i=1}^{\Delta}$ is a set of fake users
  - Polluted interaction matrix Y'

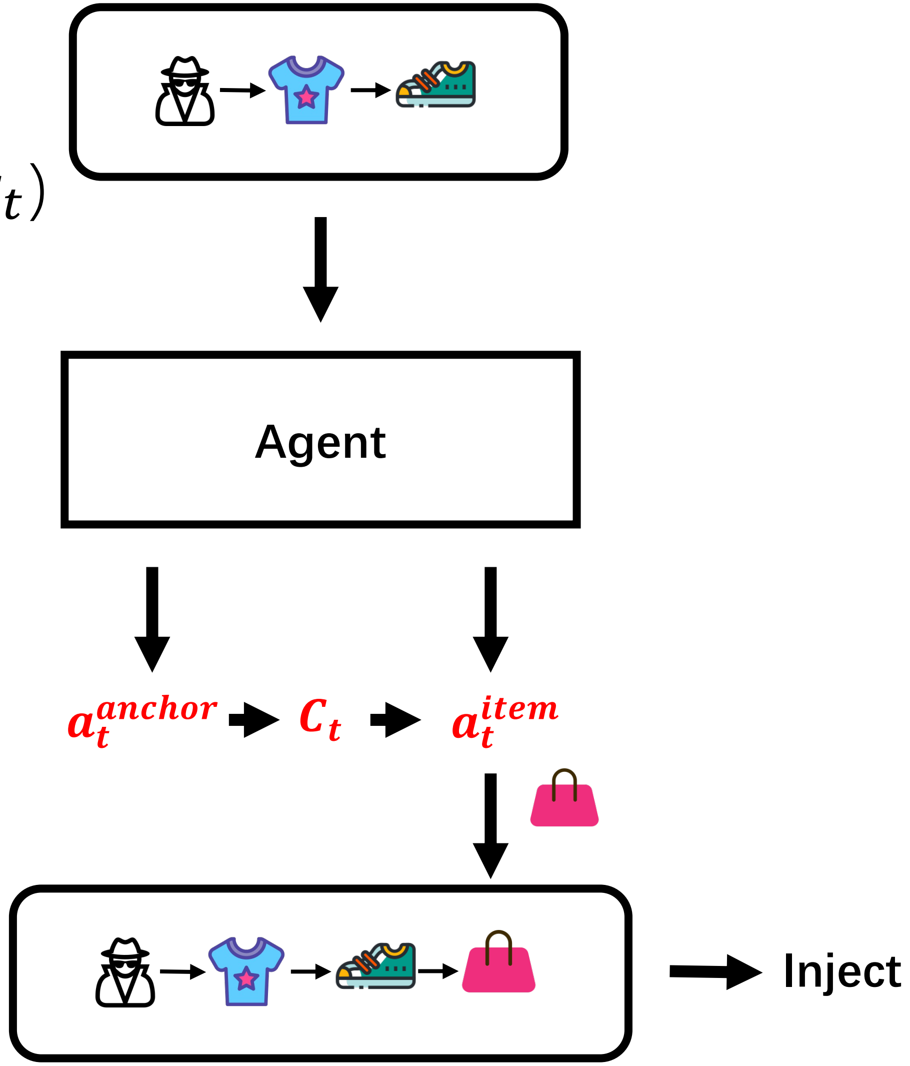
Background

- Challenges in existing black-box attacking methods
  - PoisonRec[1]: massive item sets
  - CopyAttack[2]: lack of cross-domain knowledge
- Employs the KG to enhance the generation of fake user profiles from the massive item sets
- Black-box Setting
  - Reinforcement learning – Query Feedback (Reward)

KGAttack - Attacking RL Environment

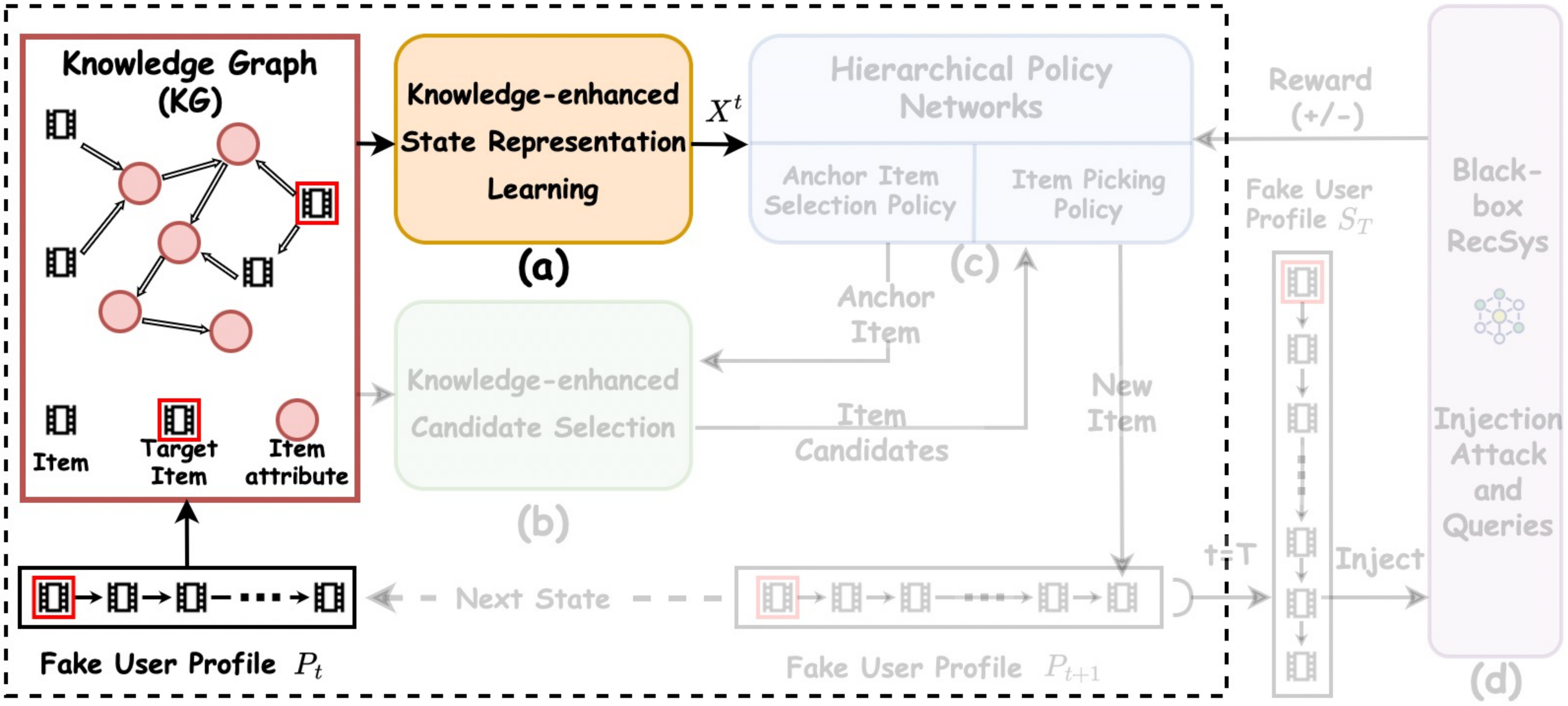
- **State s_t**
  - Fake user profile P_t at time t (representations x_t)
- **Action a_t**
  - Anchor item a_t^{anchor} item candidates pool C_t.
  - Picks an item a_t^{item} from C_t
- **Reward R**
  - Hit ratio of target item on spy users

$$r_t = \begin{cases} \frac{1}{|\hat{U}|} \sum_{i=1}^{|\hat{U}|} \mathrm{HR}(\hat{u}_i, v^*, k), & t = T-1; \\ 0 & t = 0, \ldots, T-2, \end{cases}$$

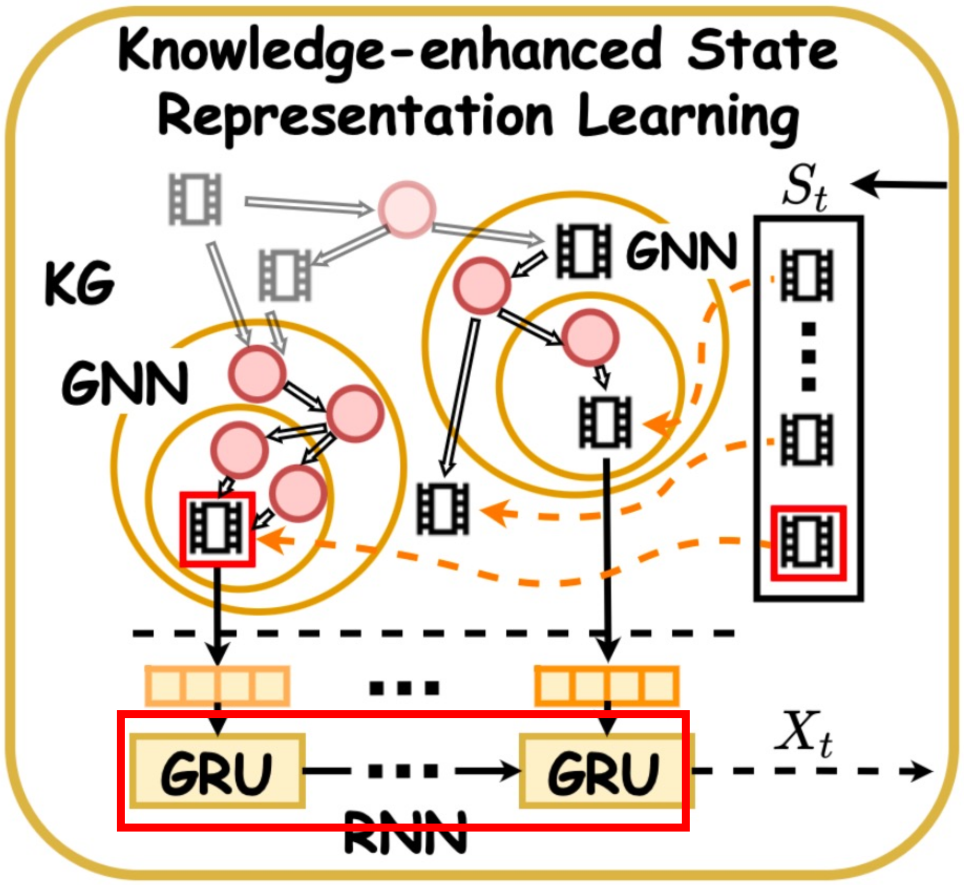


KGAttack – Framework Overview

- **(a):** Using **KG** to enhance the representation of state. **(b):** Using KG to localize relevant item candidates
- **(c):** RL agent, generate user profiles **(d):** Injection attacks and query



KGAttack - Knowledge-enhanced State Representation Learning



- Encode state s_t as representation x_t
  - **Item Initialization (TransE[3]).**

$$\mathcal{L}_{\text{pre-train}} = \sum_{(p,r,q)\in\mathcal{B}^+} \sum_{(p',r,q')\in\mathcal{B}^-} [d(\mathbf{p}+\mathbf{r},\mathbf{q})+\xi-d(\mathbf{p}'+\mathbf{r},\mathbf{q}')]_+$$

  - Item Representation (GNN).

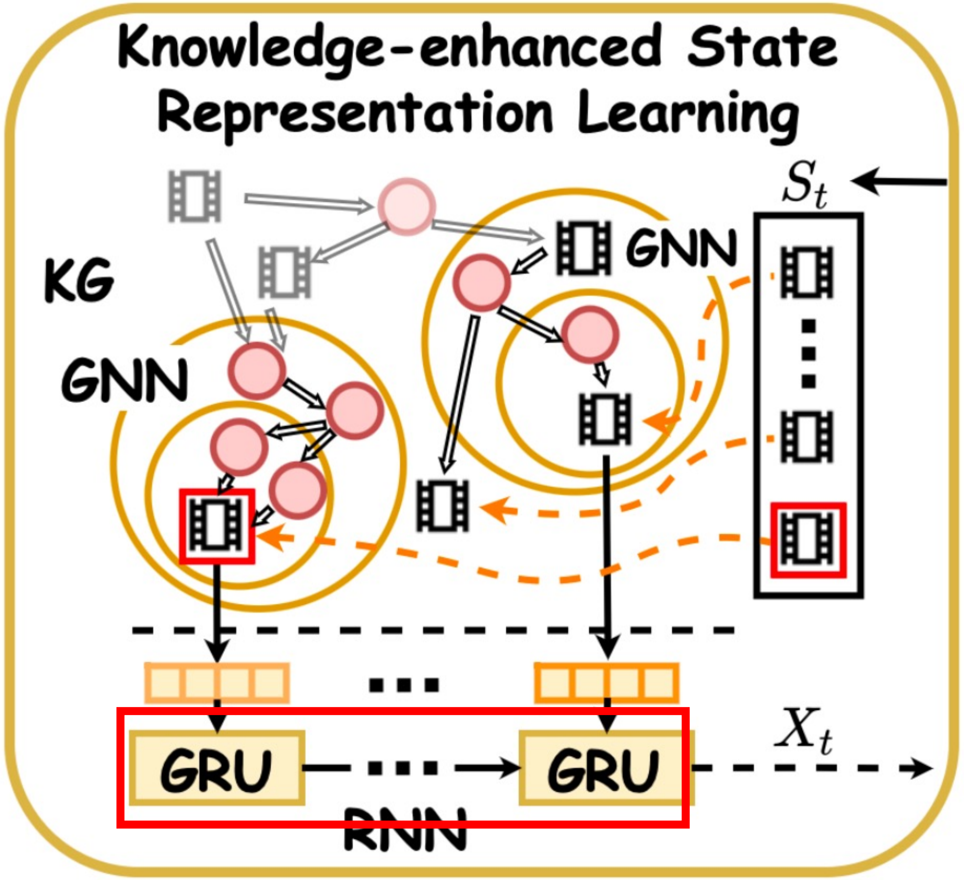
$$\mathbf{e}_i^l = \mathbf{W}_1^l \cdot \mathbf{e}_i^{l-1} + \mathbf{W}_2^l \cdot \sum_{v_j\in\mathcal{N}(v_i)} \alpha_{i,j}^l \mathbf{e}_j^{l-1},$$

$$\alpha_{i,j}^l = \text{softmax}\left((\mathbf{W}_{\text{in}} \cdot \mathbf{e}_i^{l-1})^\top (\mathbf{W}_{\text{out}} \cdot \mathbf{e}_j^{l-1}) / \sqrt{d} \right)$$

  - State Representation Learning.
    - RNN with a gated recurrent unit (GRU)

[3] Translating embeddings for modeling multi-relational data. NeurIPS 2013 (2013)

KGAttack - Knowledge-enhanced State Representation Learning



- Encode state s_t as representation x_t
  - Item Initialization (TransE[3]).

$$\mathcal{L}_{\text{pre-train}} = \sum_{(p,r,q)\in\mathcal{B}^+} \sum_{(p',r,q')\in\mathcal{B}^-} [d(\mathbf{p}+\mathbf{r},\mathbf{q})+\xi-d(\mathbf{p}'+\mathbf{r},\mathbf{q}')]_+$$

  - **Item Representation (GNN).**

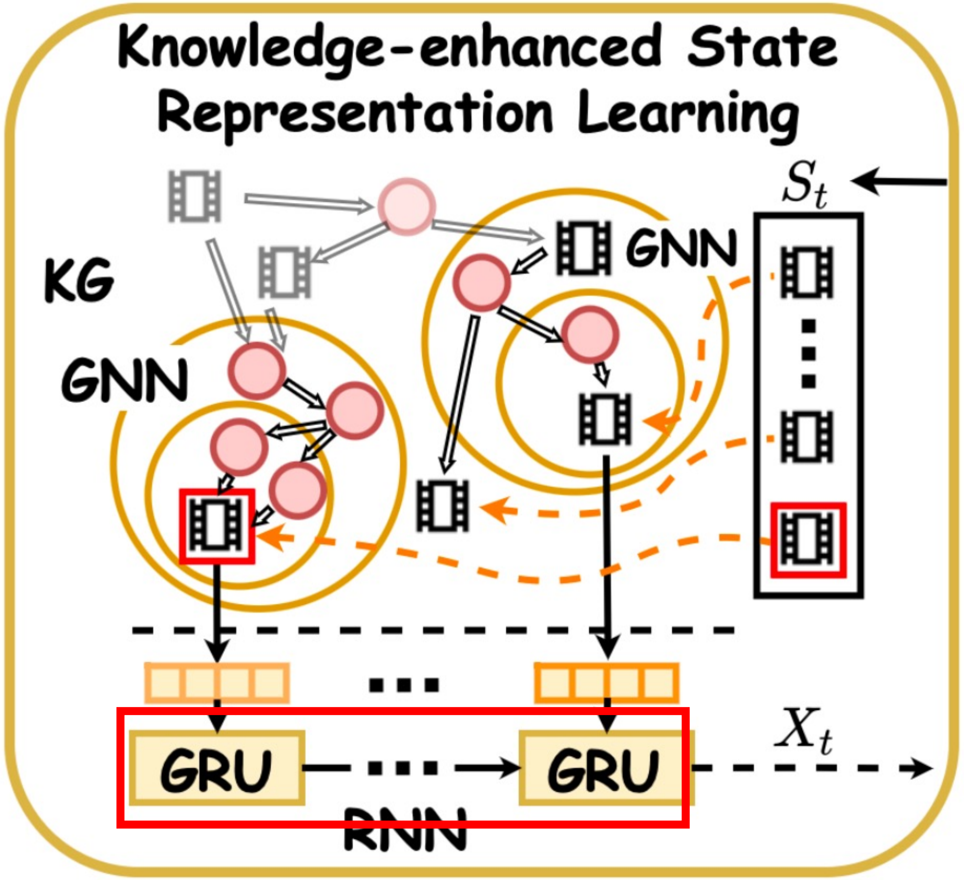
$$\mathbf{e}_i^l = \mathbf{W}_1^l \cdot \mathbf{e}_i^{l-1} + \mathbf{W}_2^l \cdot \sum_{v_j\in\mathcal{N}(v_i)} \alpha_{i,j}^l \mathbf{e}_j^{l-1},$$

$$\alpha_{i,j}^l = \text{softmax}\left(\left(\mathbf{W}_{\text{in}} \cdot \mathbf{e}_i^{l-1}\right)^\top \left(\mathbf{W}_{\text{out}} \cdot \mathbf{e}_j^{l-1}\right)/\sqrt{d} \right)$$

  - State Representation Learning.
    - RNN with a gated recurrent unit (GRU)

[3] Translating embeddings for modeling multi-relational data. NeurIPS 2013 (2013)

KGAttack - Knowledge-enhanced State Representation Learning



- Encode state s_t as representation x_t
  - Item Initialization (TransE[3]).

$$\mathcal{L}_{\text{pre-train}} = \sum_{(p,r,q)\in\mathcal{B}^+} \sum_{(p',r,q')\in\mathcal{B}^-} [d(\mathbf{p}+\mathbf{r},\mathbf{q})+\xi-d(\mathbf{p}'+\mathbf{r},\mathbf{q}')]_+$$

  - Item Representation (GNN).

$$\mathbf{e}_i^l = \mathbf{W}_1^l \cdot \mathbf{e}_i^{l-1} + \mathbf{W}_2^l \cdot \sum_{v_j \in \mathcal{N}(v_i)} \alpha_{i,j}^l \mathbf{e}_j^{l-1},$$

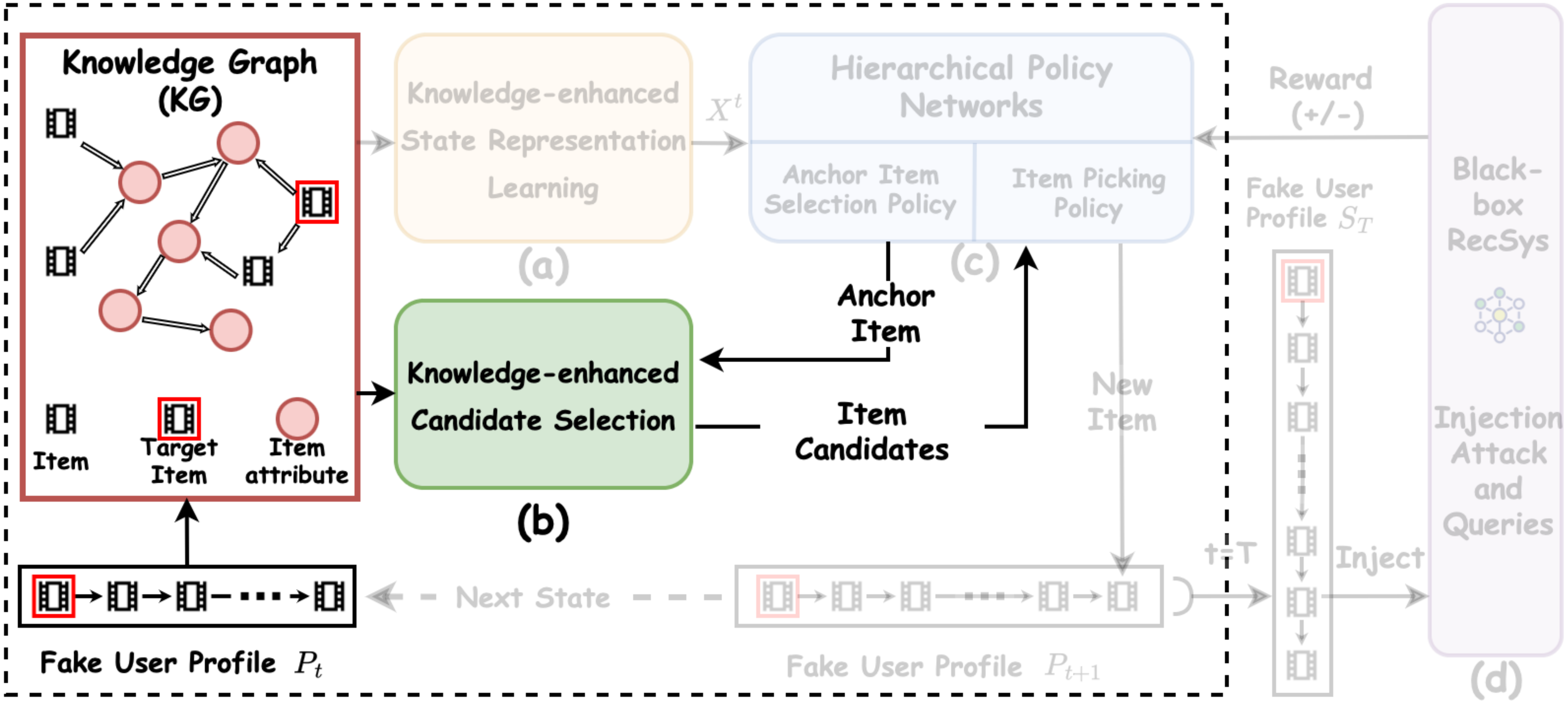
$$\alpha_{i,j}^l = \text{softmax}\left((\mathbf{W}_{\text{in}} \cdot \mathbf{e}_i^{l-1})^\top (\mathbf{W}_{\text{out}} \cdot \mathbf{e}_j^{l-1}) / \sqrt{d} \right)$$

  - **State Representation Learning.**
    - RNN with a gated recurrent unit (GRU)

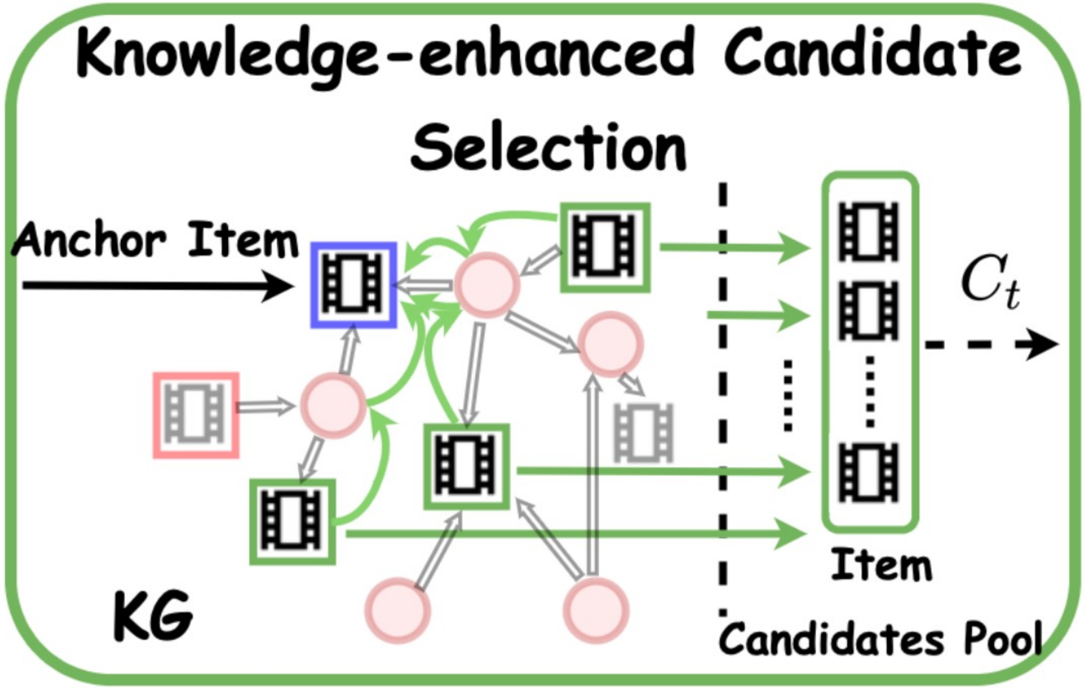
[3] Translating embeddings for modeling multi-relational data. NeurIPS 2013 (2013)

KGAttack – Framework Overview

- **(a):** Using **KG** to enhance the representation of state. **(b):** Using **KG** to localize relevant item candidates
- **(c):** RL agent, generate user profiles **(d):** Injection attacks and query



KGAttack - Knowledge-enhanced Candidate Selection



- Reduce the massive action space
  - **H-hop relevant entities of anchor item**

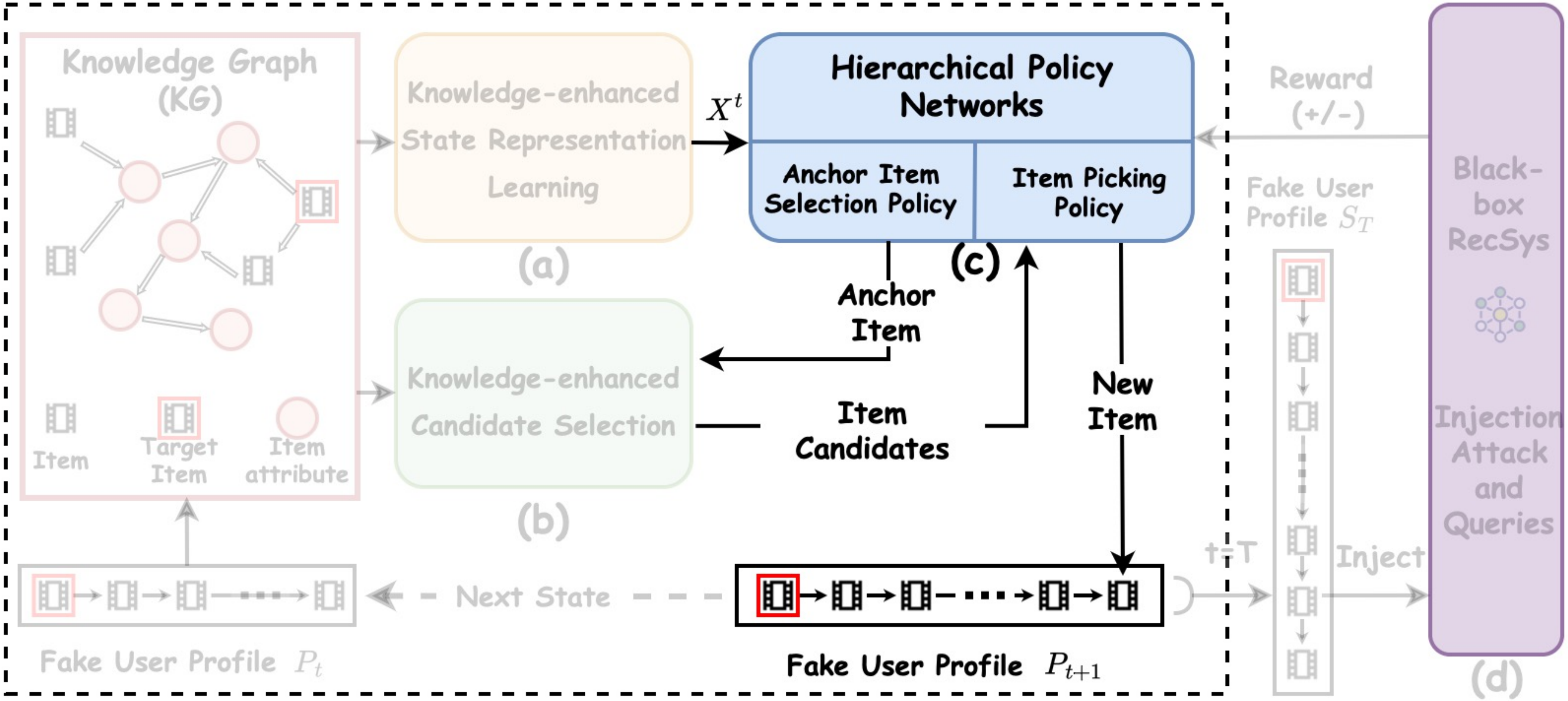
$$\mathcal{E}_t^h = \{q | (p, r, q) \in \mathcal{G}, p \in \mathcal{E}_t^{h-1}\}, h = 1, 2, ..., H,$$

  - **Collect items candidates**

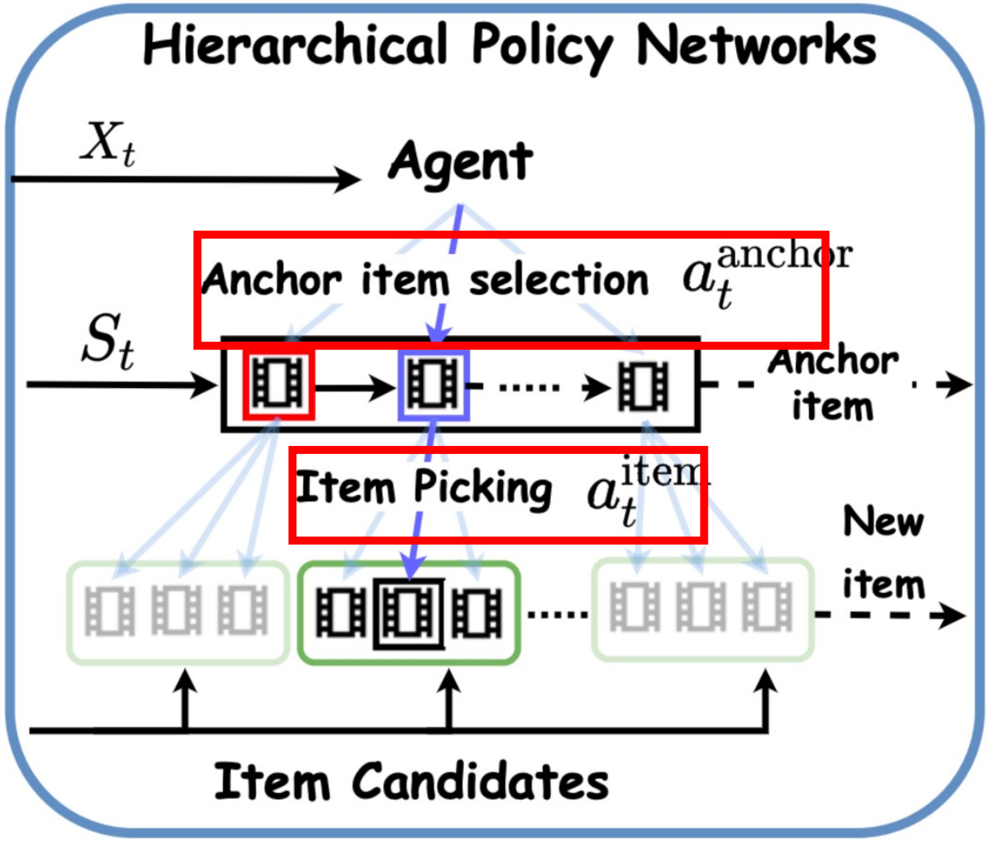
$$C_t = \{v | v \in \bigcup_{h=1}^{H} \mathcal{E}_t^h, v \in V\}$$

KGAttack – Framework Overview

- **(a):** Using **KG** to enhance the representation of state. **(b):** Using KG to localize relevant item candidates
- **(c):** RL agent, generate user profiles **(d):** Injection attacks and query



KGAttack – Hierarchical Policy Networks



- Generate fake user profiles sequentially
  - **Anchor Item Selection**

$$\pi_\theta^{\text{anchor}}(a_t^{\text{anchor}}|s_t) = \text{Softmax}(\mathbf{W}_{\text{A},2}\text{ReLU}(\mathbf{W}_{\text{A},1}\mathbf{x}_t) + \mathbf{m}_t)$$

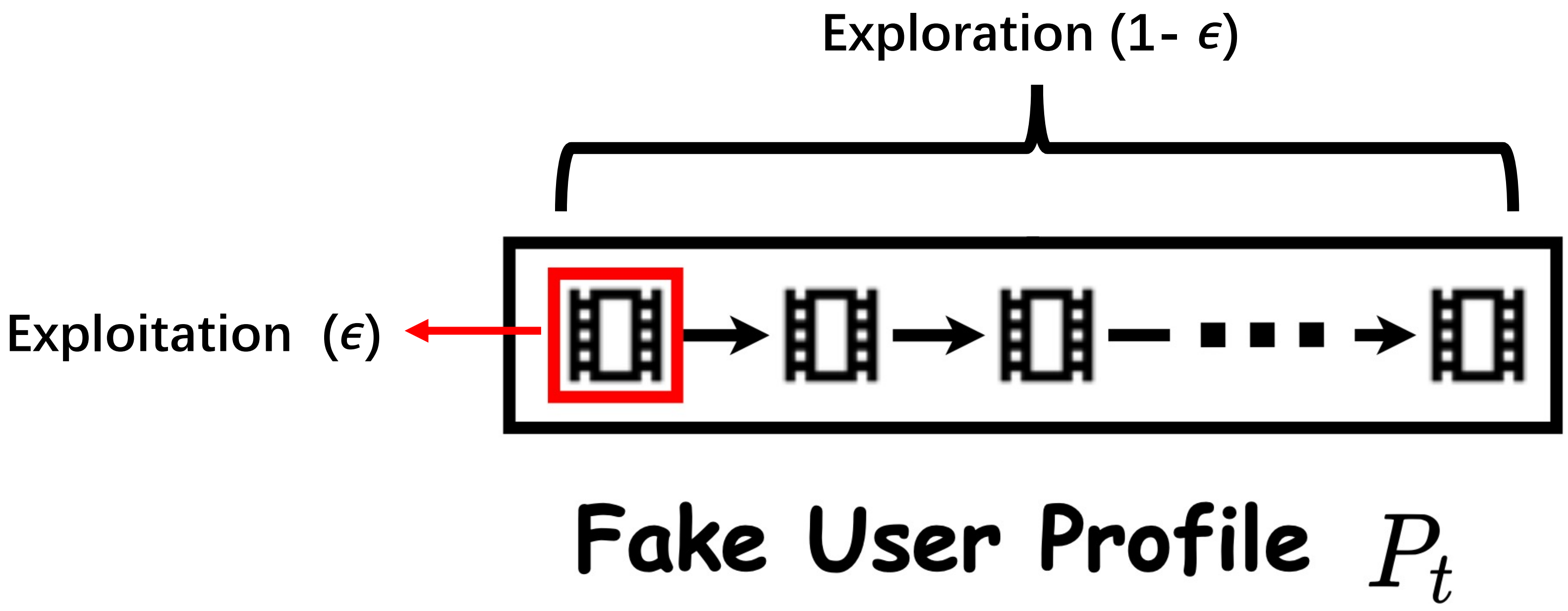
  - **Item Picking**

$$\hat{\mathbf{x}}_t = \text{ReLU}(\mathbf{W}_{\text{I},1}\mathbf{x}_t)$$

$$\pi_\phi^{\text{item}}(a_t^{\text{item}}|s_t) = \frac{\exp(\mathbf{W}_{\text{I},2}[\hat{\mathbf{x}}_t; \mathbf{e}_t])}{\sum_{v_j \in C_t} \exp(\mathbf{W}_{\text{I},2}[\hat{\mathbf{x}}_t; \mathbf{e}_j])}$$

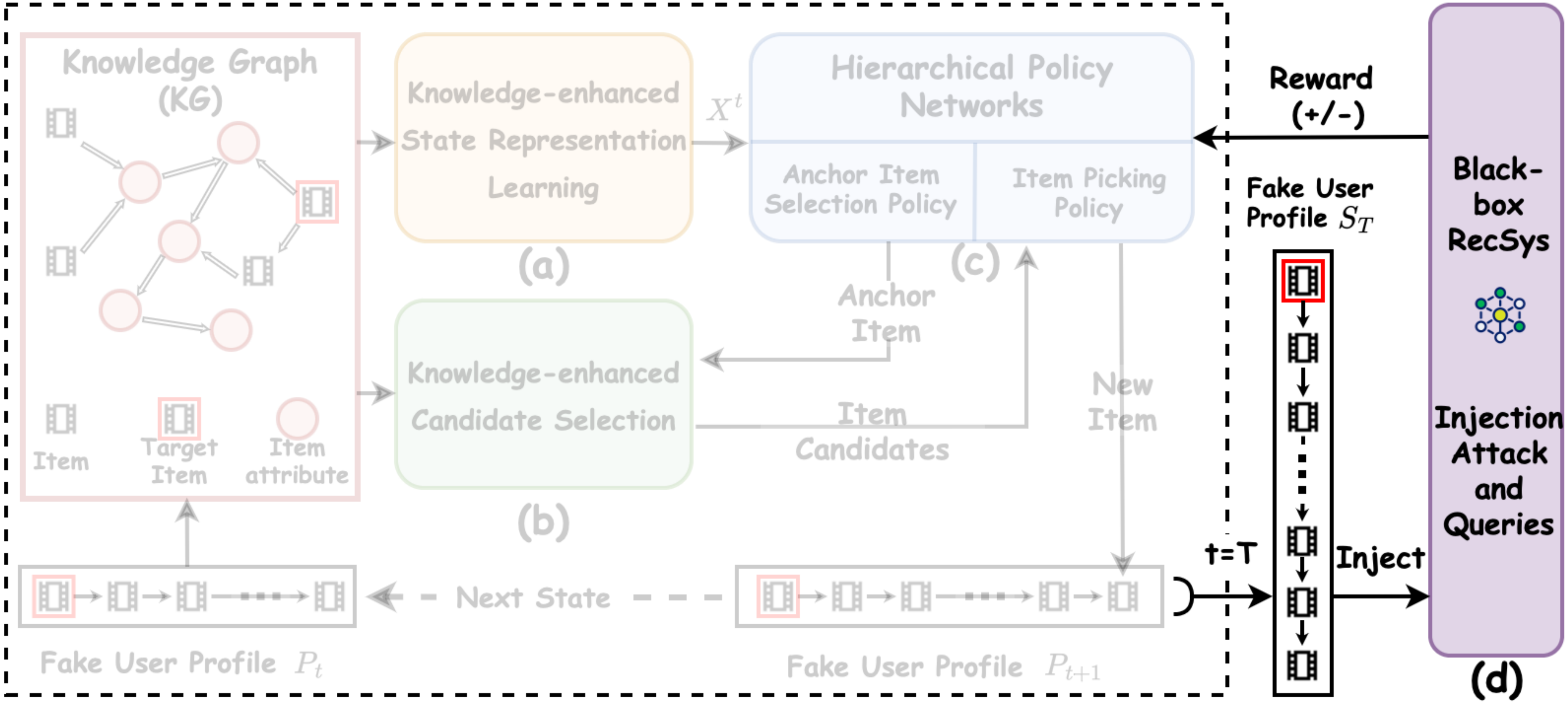
KGAttack – Hierarchical Policy Networks

- Anchor item Selection:
  - Exploitation: Target item
  - Exploration: Select by Policy network



KGAttack – Framework Overview

- **(a):** Using **KG** to enhance the representation of state. **(b):** Using KG to localize relevant item candidates
- **(c):** RL agent, generate user profiles **(d):** Injection attacks and query



KGAttack – Model Training

- **First stage**: Trajectory generation
  - Generate N fake user profile

- **Second stage**: Policy Networks update
  - Two actor networks and critic network are updated

Algorithm 1 KGAttack

Randomly initialize the Actor π_θ, π_ϕ and Critic V_ω with parameters θ, ϕ and ω.
Initialize replay memory buffer $\mathcal{D}$
for episode number c in $[0, \Delta/N)$ **do**
  // (i) Trajectory Generation

  for fake user i in $[m + cN + 1, m + (c+1)N + 1]$ **do**
    Initialize state s_0 based on $P_{0,u_i} = \{v_*\}$
    for step t in $[0, T-1]$ **do**
      Select anchor item v_t^{anchor} according to $\pi_\theta^{\text{anchor}}$ with anchor ratio ϵ
      generate the item candidates C_{t,u_i} according to v_t^{anchor}
      Pick a new item v_t according to π_θ^{item} and C_{t,u_i}
      Obtain state $s_{t+1} = \{s_t, v_t\}$ and reward r_t
      Push $\{s_t, a_t^{\text{item}}, a_t^{\text{anchor}}, r_t, s_{t+1}\}$ into the memory buffer $\mathcal{D}$
    end for
  end for

  //(ii) Networks Update
  Get transitions from replay memory buffer $\mathcal{D}$
  Update the critic network V_ω by minimizing the loss in Equation (2)
  Update the actor networks π_θ, π_ϕ by maximizing Equation (13) via stochastic gradient ascent with Adam.
  Clean replay memory buffer $\mathcal{D}$
end for

Experiments

- **Datasets**
  - MovieLens-1M, Book-Crossing, Last.FM

	Attribute	MovieLens-1M	Book-Crossing	Last.FM
Dataset	# Users	5,950	13,097	1,874
	# Items	3,532	306,776	17,612
	# Interactions	574,619	1,149,772	92,780
	# Items in KG	2,253	14,114	3,844
KG	# Entities	182,011	77,903	9,366
	# Relations	12	25	60
	# KG triples	1,241,995	151,500	15,518
	Avg. 1-hop NBR	27	15	5
	Avg. 2-hop NBR	298	24	14
	Avg. 3-hop NBR	1,597	82	60

- **Evaluation Metrics**
  - HR@K, NDCG@K (K=10, 20)

- **Baselines**
  - Traditional methods: RandomAttack, TargetAttack, TargetAttack-KG
  - RL-based methods: PoisonRec, PoisonRec-KG
  - KGAttack variants: KGAttack-Target, KGAttack-Seq

Experiments – Overall Performance (Pinsage)

Q1: How effective/evasive is KGAttack in evasion attack tasks ?

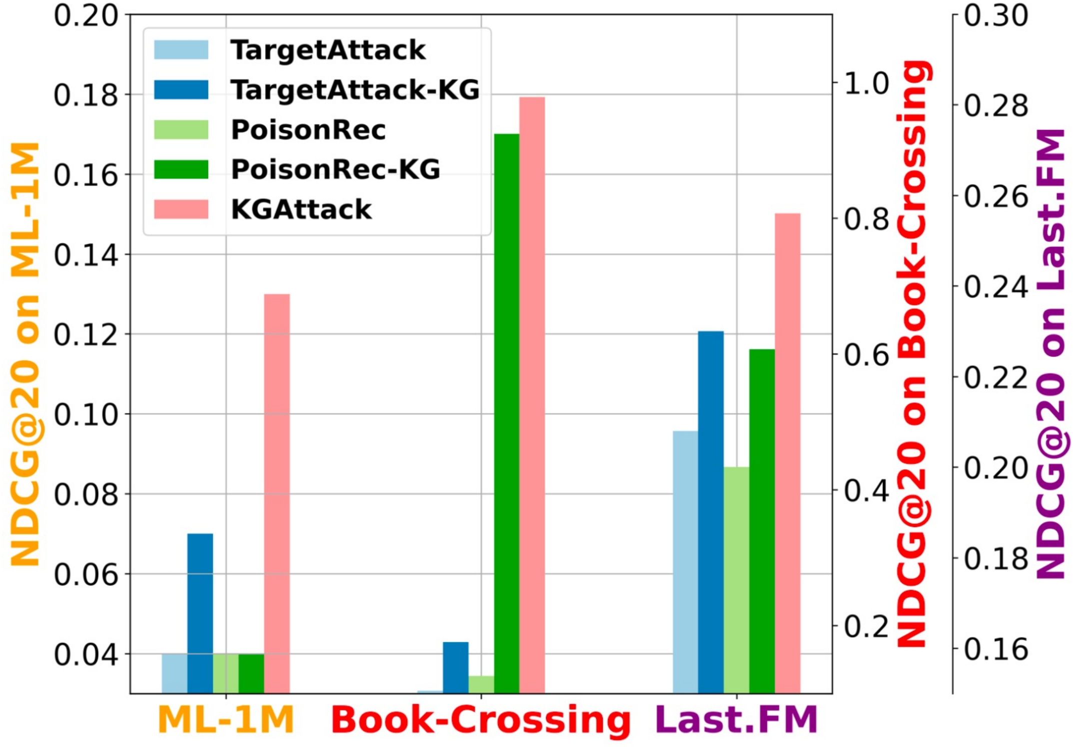
- DRL-based attacking methods
- KGAttack
- Hierarchical policy networks

Dataset	MovieLens-1M (ML-1M)				Book-Crossing				Last.FM			
	H@20	H@10	N@20	N@10	H@20	H@10	N@20	N@10	H@20	H@10	N@20	N@10
Without Attack	0.000	0.000	0.000	0.000	0.191	0.095	0.065	0.042	0.193	0.012	0.073	0.005
RandomAttack	0.000	0.000	0.000	0.000	0.202	0.092	0.069	0.041	0.152	0.092	0.054	0.040
TargetAttack	0.464	0.056	0.118	0.017	0.706	0.370	0.226	0.141	0.242	0.042	0.064	0.014
TargetAttack-KG	0.398	0.028	0.099	0.008	0.862	0.606	0.342	0.276	0.282	0.110	0.085	0.043
PoisonRec	0.610	0.138	0.162	0.047	0.930	0.748	0.428	0.381	0.442	0.148	0.125	0.052
PoisonRec-KG	0.628	0.108	0.163	0.035	0.930	0.748	0.427	0.380	0.438	0.148	0.123	0.051
KGAttack-Target	0.554	0.009	0.144	0.029	**0.940**	0.780	0.437	0.396	0.442	0.144	0.125	0.051
KGAttack-Seq	0.504	0.009	0.132	0.031	0.932	0.750	0.425	0.379	0.436	0.148	0.123	0.051
KGAttack	**0.672**	**0.184**	**0.183**	**0.063**	0.934	**0.788**	**0.459**	**0.422**	**0.452**	**0.152**	**0.130**	**0.053**

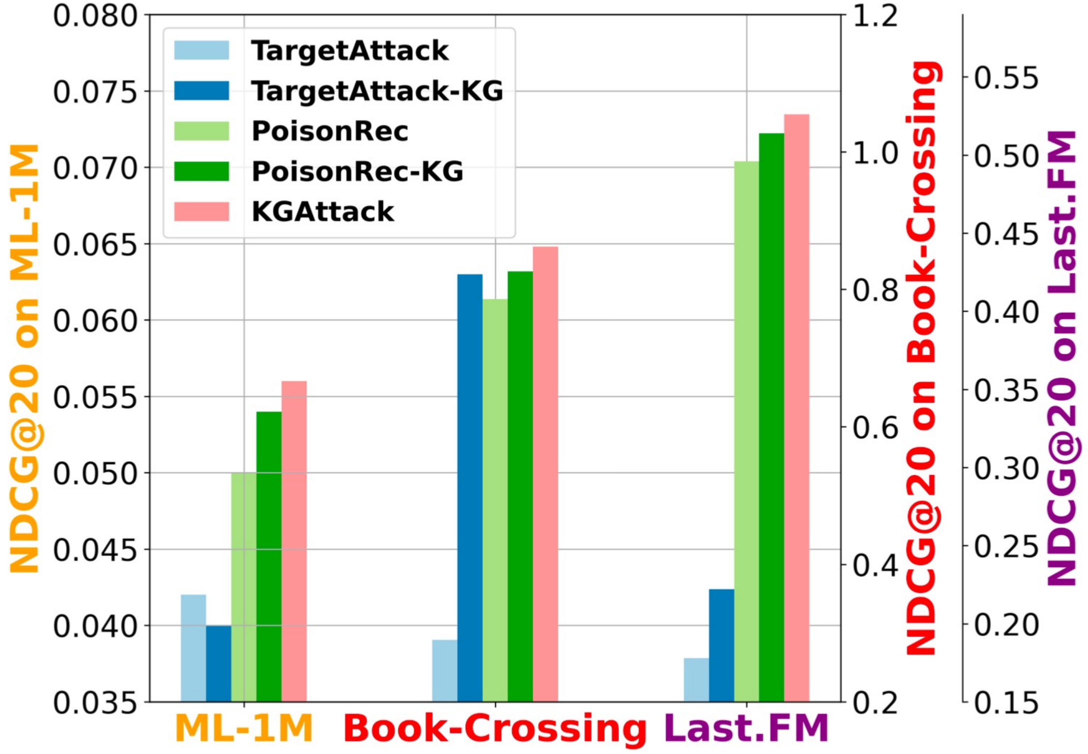
Experiments – Overall Performance (KGCN/NeuMF)

Q2: How effective/evasive is KGAttack in poison attack tasks ?

- KG-incorporated methods on KGCN.
- KGAttack almost beat all baselines on these two target models



(a) KGCN: HR@20



(c) NeuMF: HR@20

Experiments – Ablation Study

Q3: How effective is each component in KGAttack?

- **KGAttack (-KGE) / (-GNN)** vs. **KGAttack**
- **KGAttack (-Relevant)** vs. **KGAttack**
- **KGAttack (-HPN)** vs. **KGAttack**

Models	MoveLens-1M		Book-Crossing		Last.FM	
	H@20	N@20	H@20	N@20	H@20	N@20
KGAttack (-KGE)	0.598	0.163	0.928	0.442	0.422	0.119
KGAttack (-GNN)	0.630	0.161	0.926	0.442	0.446	0.124
KGAttack (-Relevant)	0.628	0.163	0.930	0.427	0.438	0.123
KGAttack (-HPN)	0.532	0.140	0.926	0.421	0.430	0.121
KGAttack	**0.672**	**0.183**	**0.934**	**0.459**	**0.460**	**0.130**

Experiments – Parameter Analysis

Q4: How anchor ratio ϵ affects performance?

- Prefers selecting anchor item via hierarchical policy networks
- Encouraging the target item as the anchor item excessively will degrade the attacking performance

ϵ	0.1	0.3	0.5	0.7	0.9
MovieLens-1M	0.582	0.534	0.620	0.622	**0.660**
Book-Crossing	0.916	0.920	**0.934**	0.928	0.930
Last.FM	0.432	0.444	0.442	**0.460**	0.448

Conclusions

- Propose a knowledge-enhanced attacking framework for black-box recommender systems (**KGAttack**)
  - Leverage knowledge graph (KG) to enhance the generation of fake user profiles
  - In KGAttack, the knowledge graph can be seamlessly integrated into hierarchical policy networks to effectively perform adversarial attacks

Thank You

Jingfan Chen: jingfan.chen@smail.nju.edu.cn

Please see my homepage for more details:

https://cjfcsjt.github.io

Knowledge-enhanced Black-box Attacks for Recommendations

Jingfan Chen[1], Wenqi Fan[2], Guanghui Zhu[1], Xiangyu Zhao[3], Chunfeng Yuan[1], Qing Li[2], Yihua Huang[1]

[1]Nanjing University

[2]The Hong Kong Polytechnic University

[3]City University of Hong Kong