Reinforcement Learning for Recommender Systems

Xiangyu Zhao

Data Science and Engineering Lab

Michigan State University

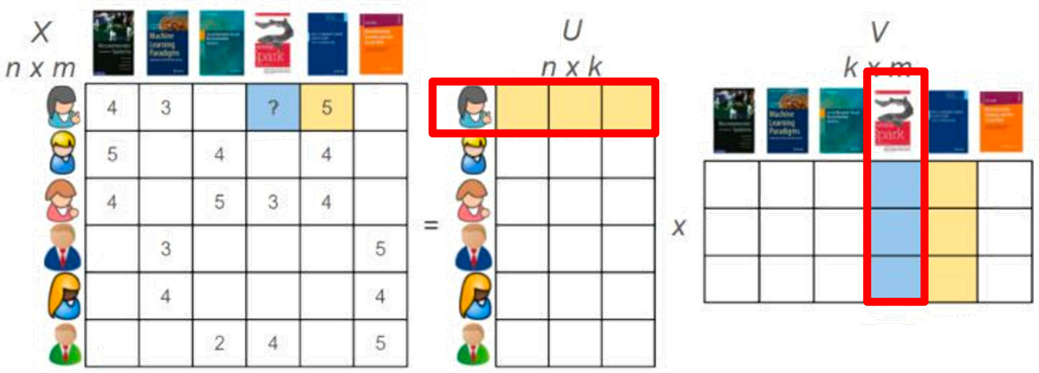
www.cse.msu.edu/~zhaoxi35 , zhaoxi35@msu.edu

Recommender Systems

- Intelligent system that assists users' information seeking tasks

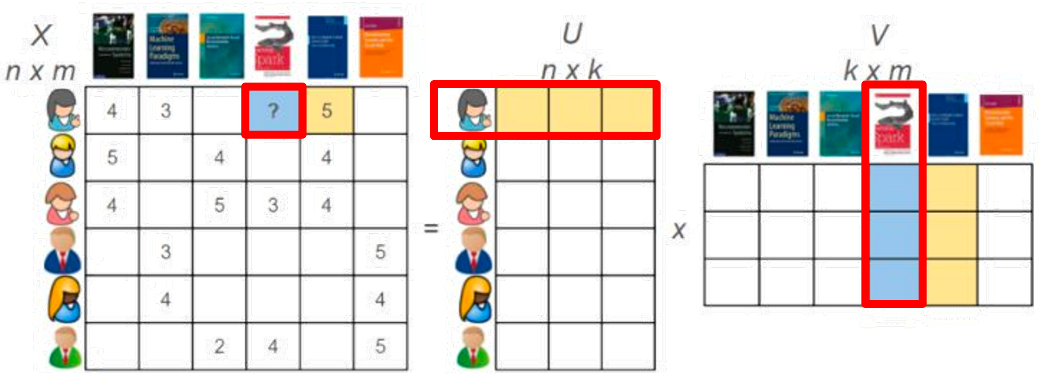
Music **Video** **Ecommerce** **News** **Social Friends** **Location based**

- Goal: Suggesting items that best match users' preferences



Existing Recommendation Policies

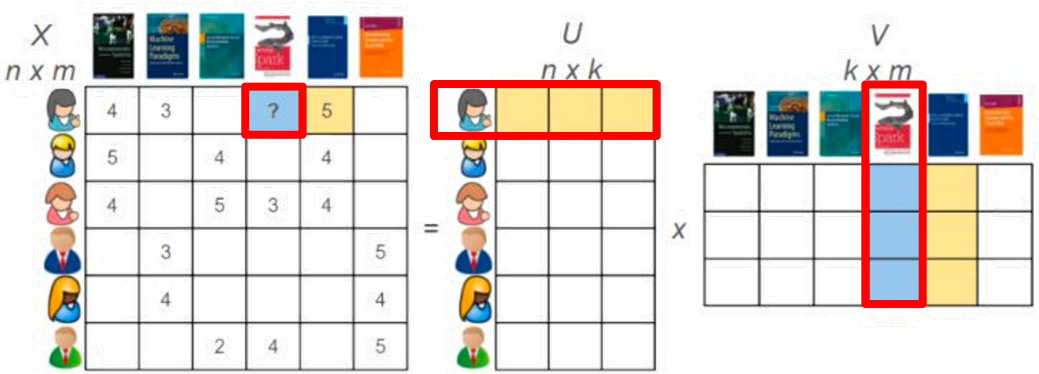
- Considering recommendation as an offline optimization problem
- Following a greedy strategy to maximize the immediate rewards from users

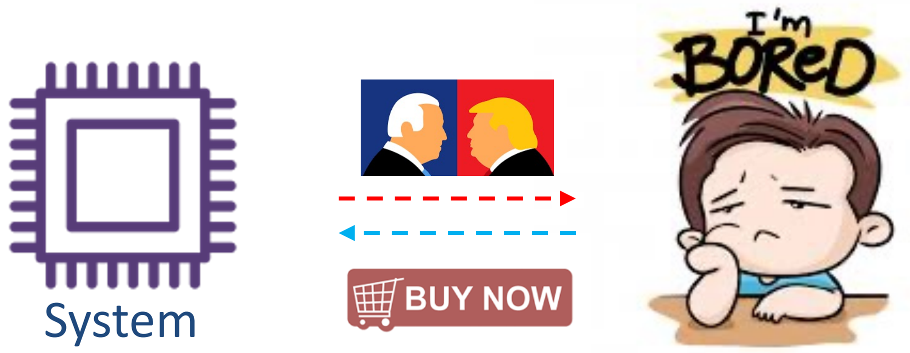


- Disadvantages
  - Overlooking real-time feedback
  - Overlooking the long-term influence on user experience

Existing Recommendation Policies

- Considering recommendation as an offline optimization problem
- Following a greedy strategy to maximize the immediate rewards from users

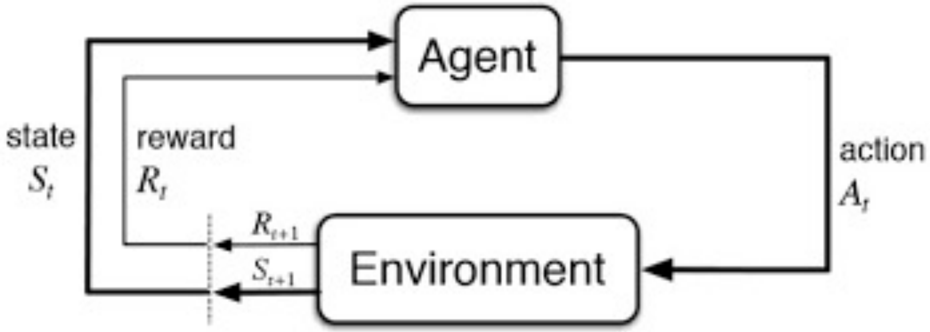




- Disadvantages
  - Overlooking real-time feedback
  - Overlooking the long-term influence on user experience

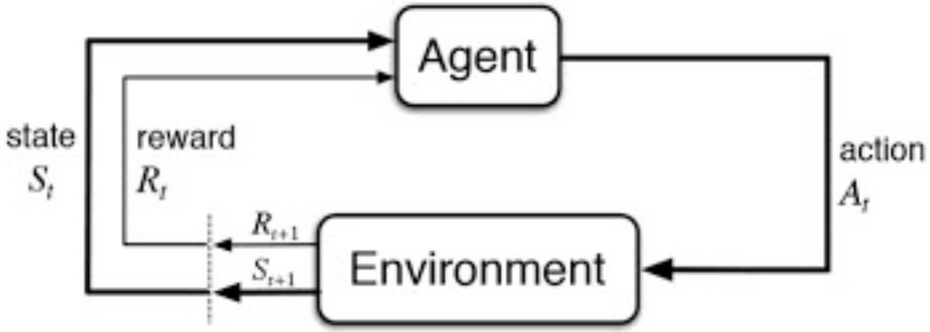
Reinforcement Learning

- Goal: selecting actions to maximize future reward

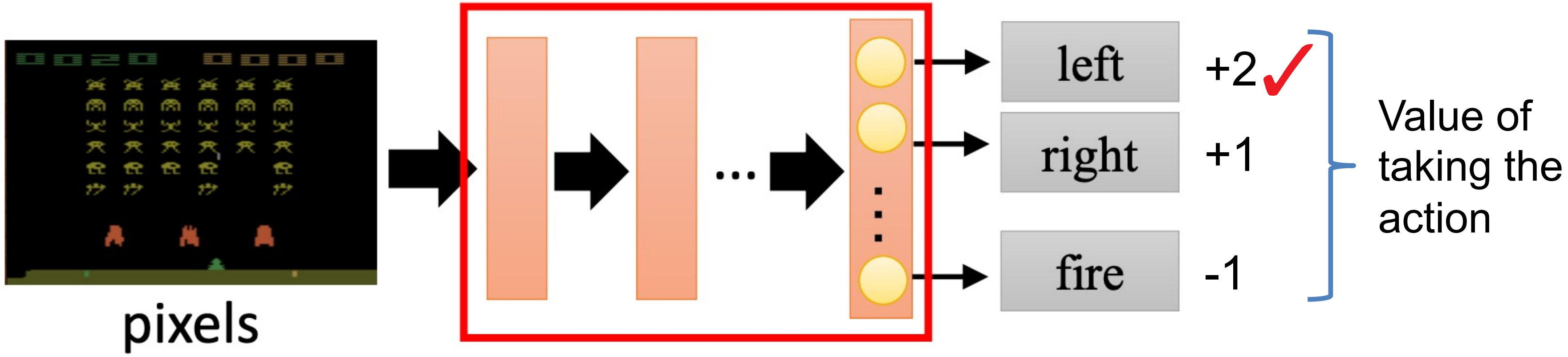


Reinforcement Learning

- Goal: selecting actions to maximize future reward

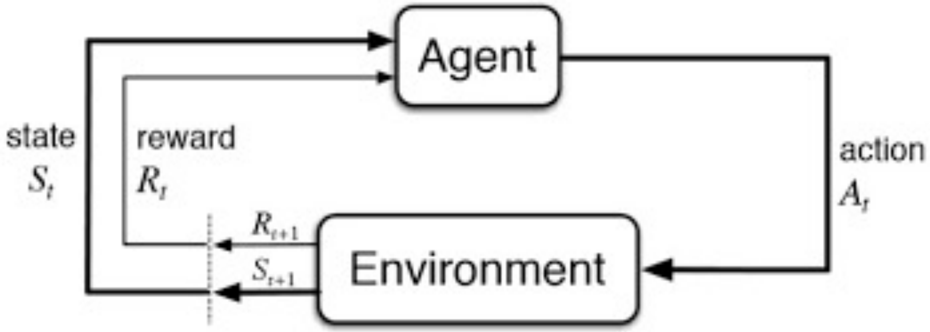


- Value-based Reinforcement Learning

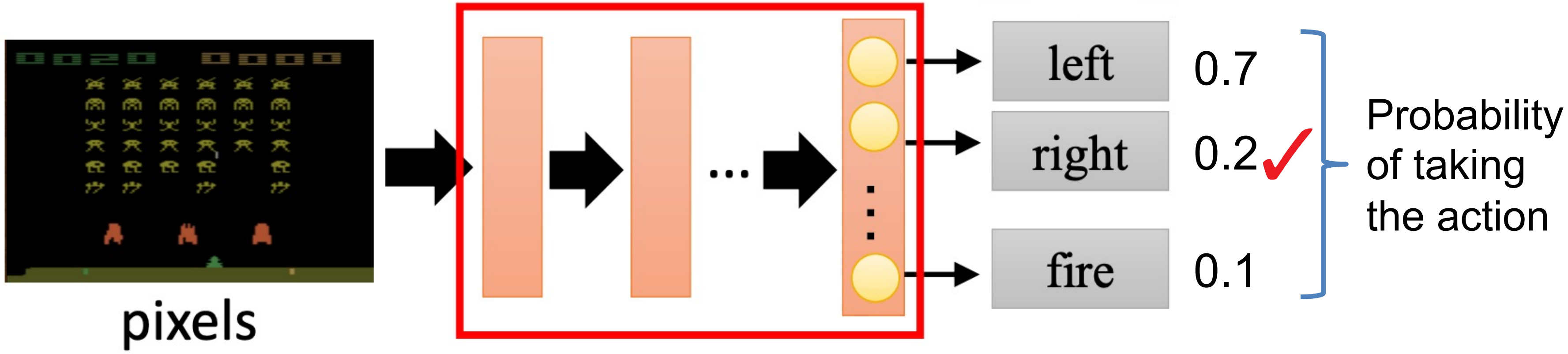


Reinforcement Learning

- Goal: selecting actions to maximize future reward

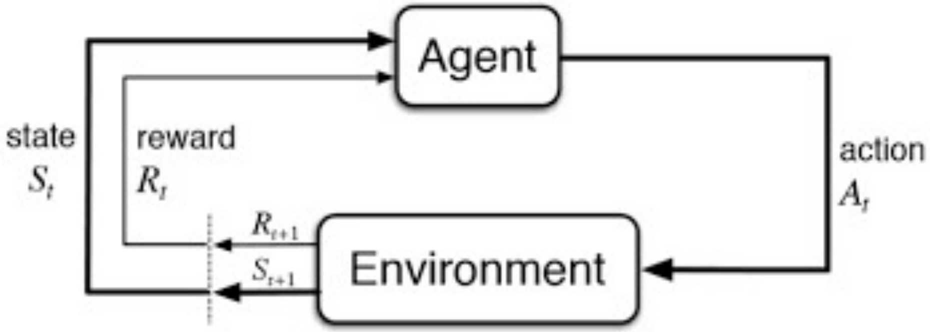


- Policy-based Reinforcement Learning

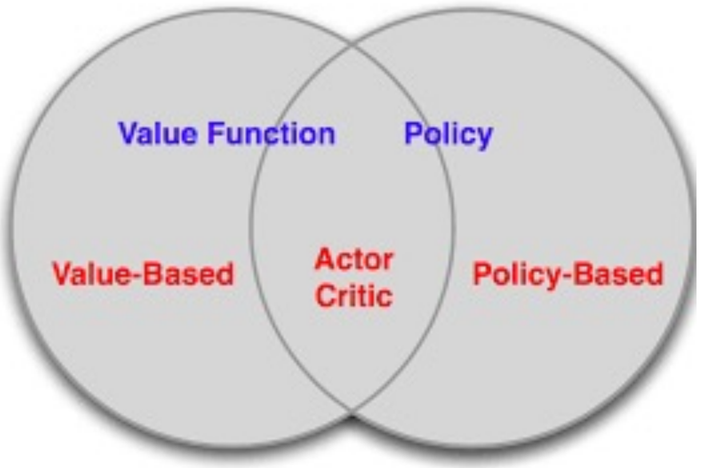


Reinforcement Learning

- Goal: selecting actions to maximize future reward

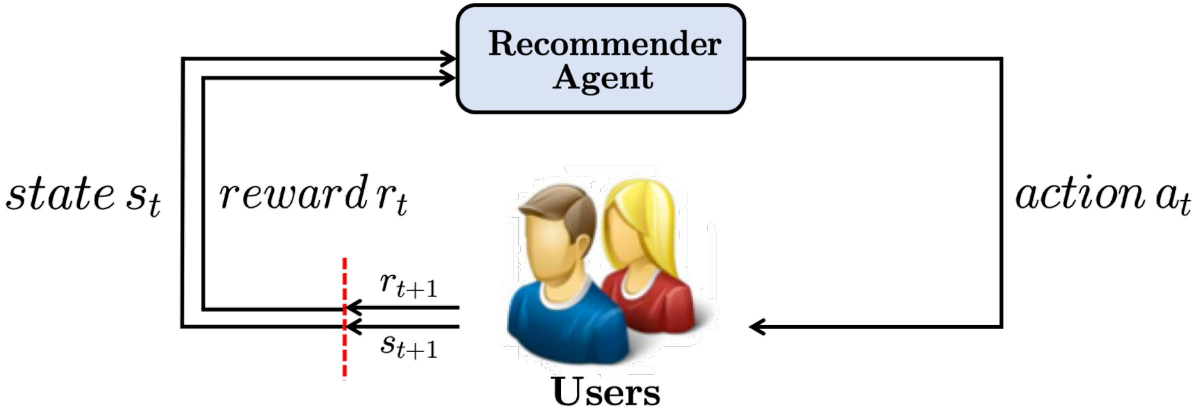


- Actor-Critic

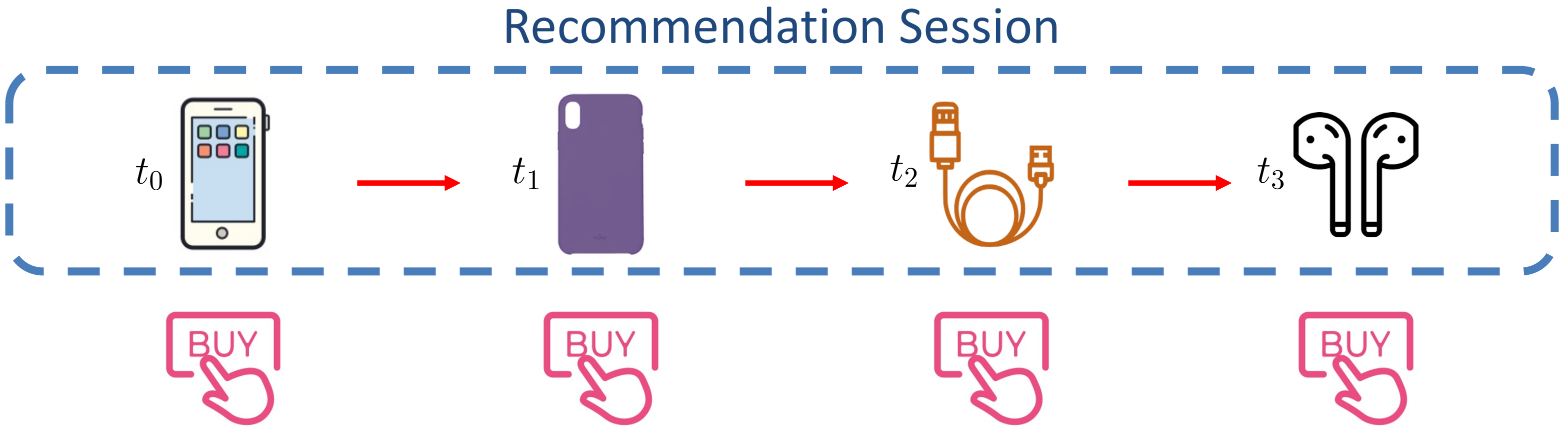


Reinforcement Learning for Recommendation Policies

- Continuously updating the recommendation strategies during the interactions



- Maximizing the long-term reward from users



Outline

- Recommendations in Single Scenario
  - DeepPage - Deep Reinforcement Learning for Page-wise Recommendations (RecSys'2018)
  - DEERS - Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning (KDD'2018)
  - DRN - A Deep Reinforcement Learning Framework for News Recommendation (WWW'2018)
- Recommendations in Multiple Scenarios
  - DeepChain - Whole-Chain Recommendations (CIKM'2020)
  - MA-RDPG - Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning (WWW'2018)
  - RAM - Jointly Learning to Recommend and Advertise (KDD'2020)
  - DEAR - Deep Reinforcement Learning for Online Advertising in Recommender Systems (AAAI'2021)
- Online Environment Simulator
  - UserSim - User Simulation via Supervised Generative Adversarial Network (WWW'2021)
- Surveys
  - Deep Reinforcement Learning for Search, Recommendation, and Online Advertising: A Survey (SIGWEB'2019)
  - Reinforcement Learning based Recommender Systems: A Survey (Arxiv'2021)

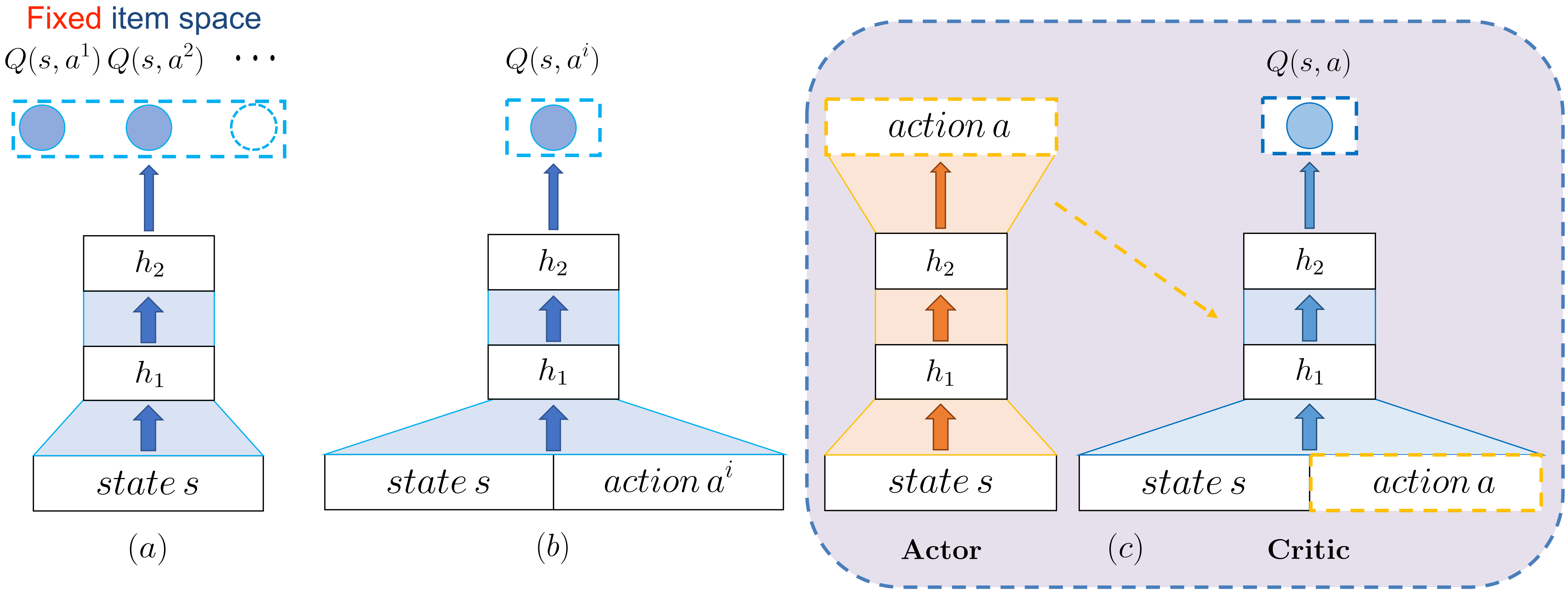
User-System Interactions

- The system recommends a page of items to a user
- The user provides real-time feedback and the system updates its policy
- The system recommends a new page of items

Challenges

- Updating strategy according to user's real-time feedback
- Diverse and complementary recommendations
- Displaying items in a 2-D page

Actor-Critic



$$Q^*(s,a) = \mathbb{E}_{s'}\left[r + \gamma \boxed{\max_{a'}} Q^*(s',a')|s,a\right]$$

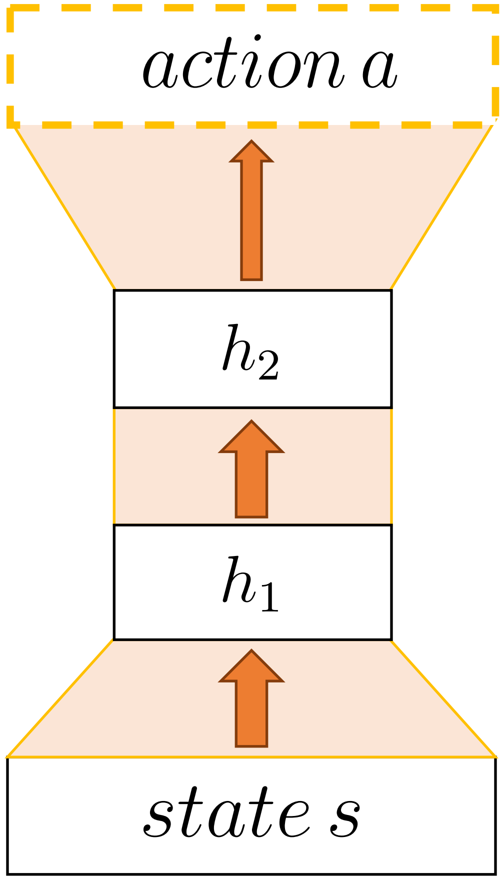
$$Q(s,a) = \mathbb{E}_{s'}\left[r + \gamma Q(s',a')|s,a\right]$$

max → enumerating all possible items

Actor Design

- Goal: Generating a page of recommendations according to user's browsing history

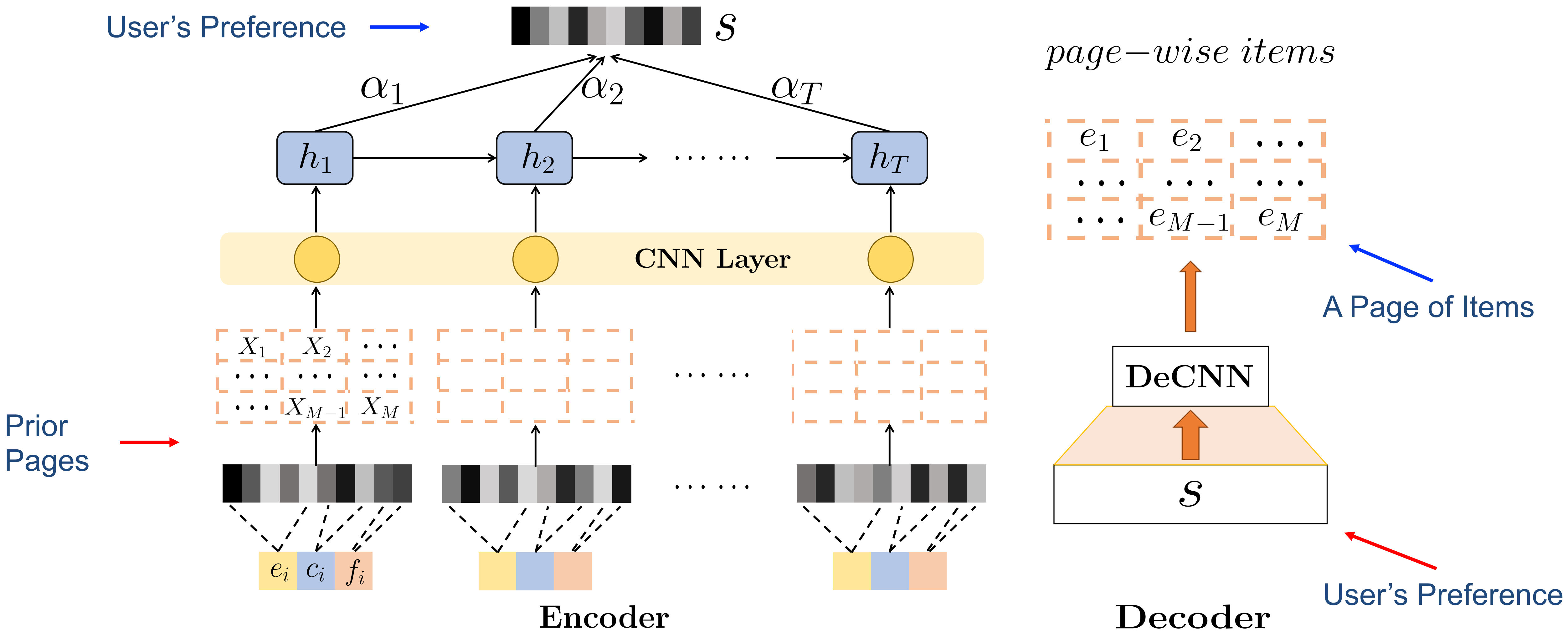
- Challenges
  - Preference from real-time feedback
  - A set of complementary items
  - Displaying items in a page



Actor

Actor Architecture

- Goal: Generating a page of items according to user's browsing history



Embedding Layer

- Three types of information
  - e_i: item's identifier
  - c_i: item's category
  - f_i: user's feedback

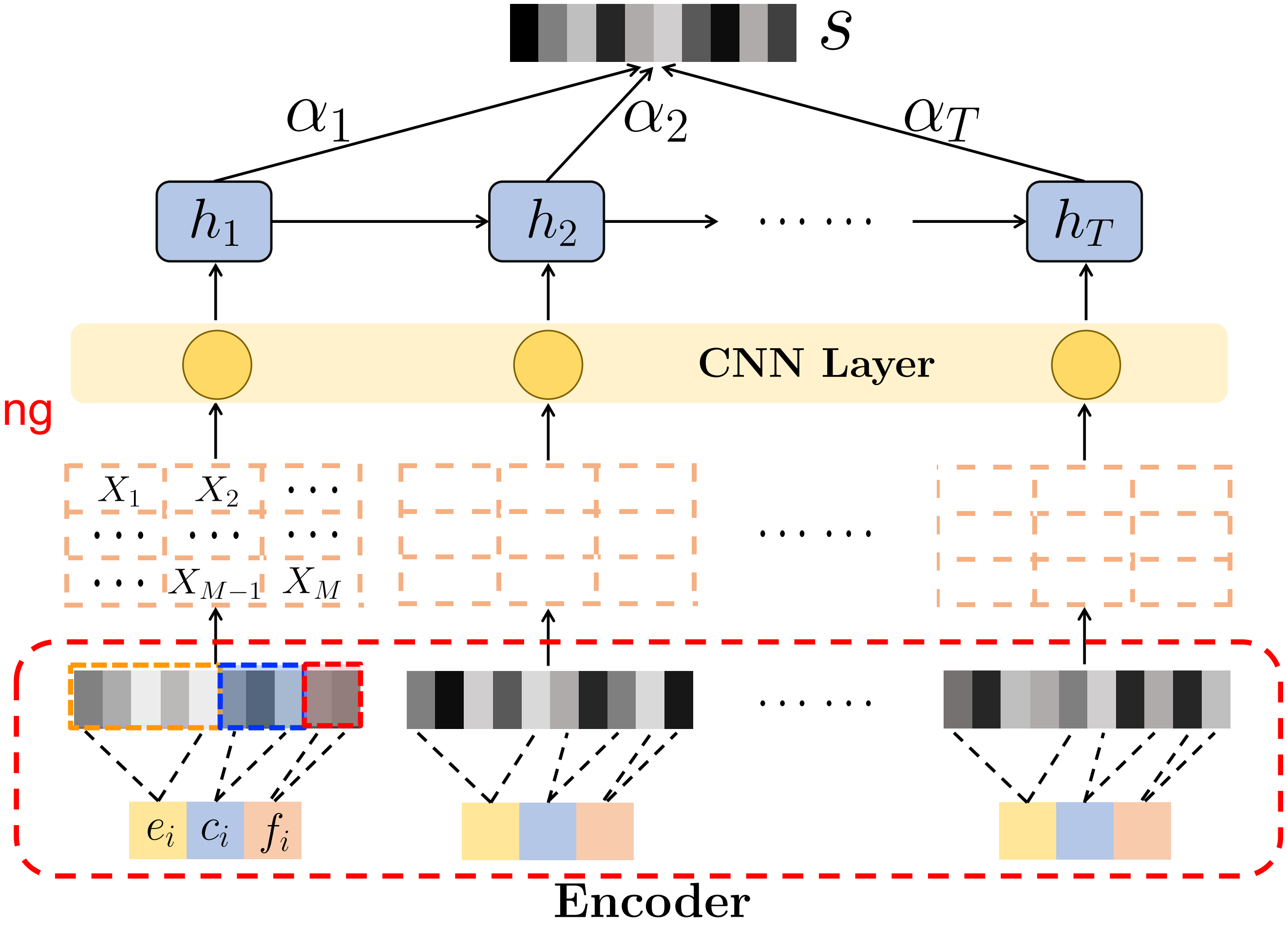
Item Embedding

Feedback Embedding

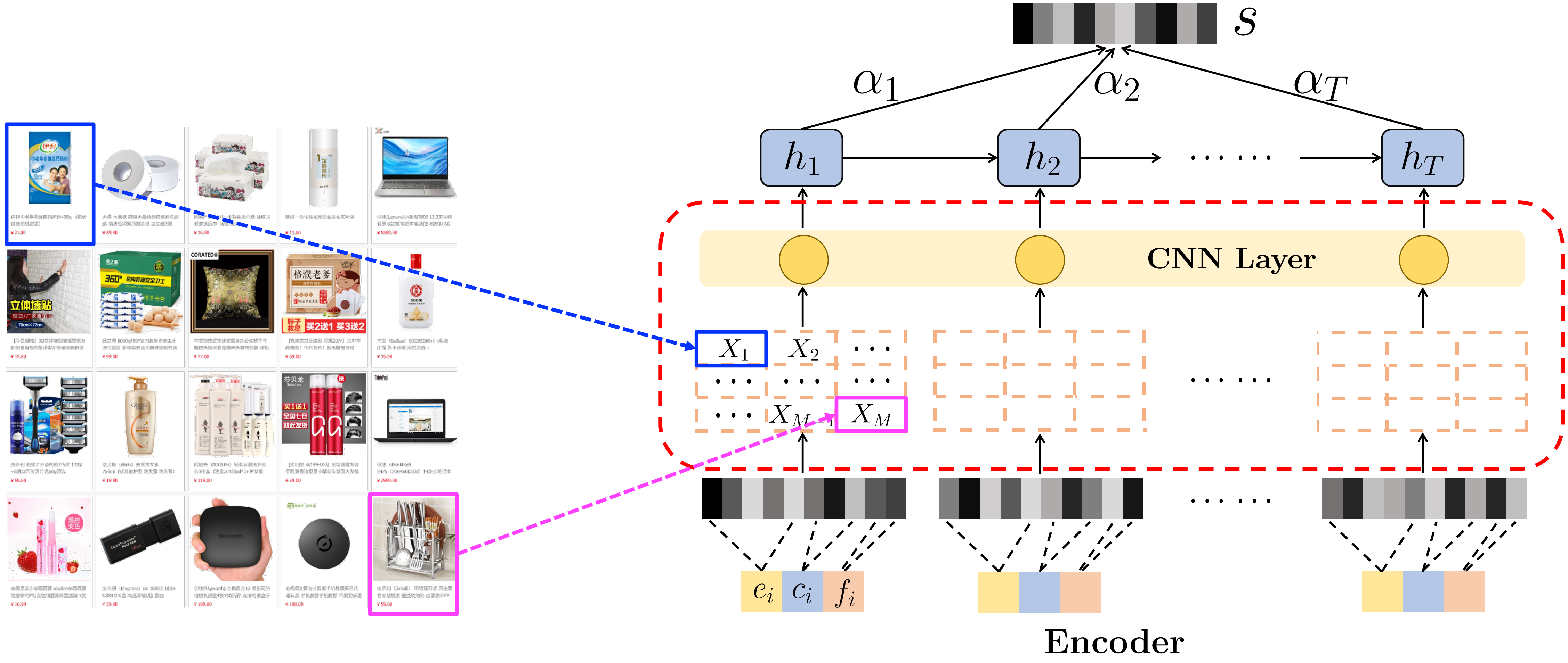
$$\boxed{X_i} = concat(E_i, C_i, F_i)$$
$$= \tanh\left(concat(\underbrace{W_E e_i + b_E}_{\text{Identifier Embedding}}, \underbrace{W_C c_i + b_C}_{\text{Category Embedding}}, \underbrace{W_F f_i + b_F}_{\text{Feedback Embedding}})\right)$$

Identifier Embedding

Category Embedding



Page-wise CNN Layer



RNN & Attention Layer

$$s = \sum_{t=1}^{T} \alpha_t h_t$$

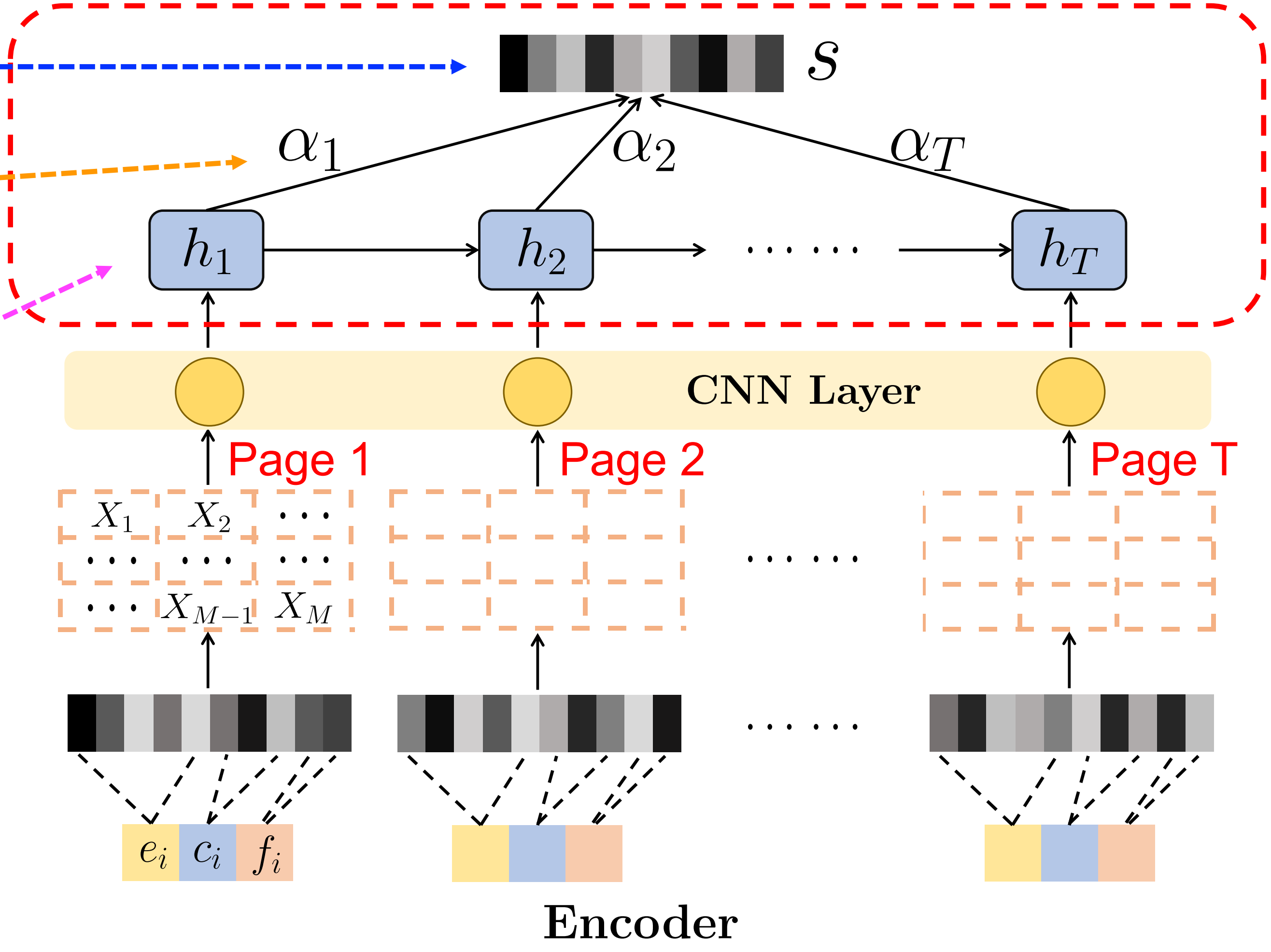
User Preference

$$where\ \alpha_t = \frac{exp(W_\alpha h_t + b_\alpha)}{\sum_j exp(W_\alpha h_j + b_\alpha)}$$

Attention

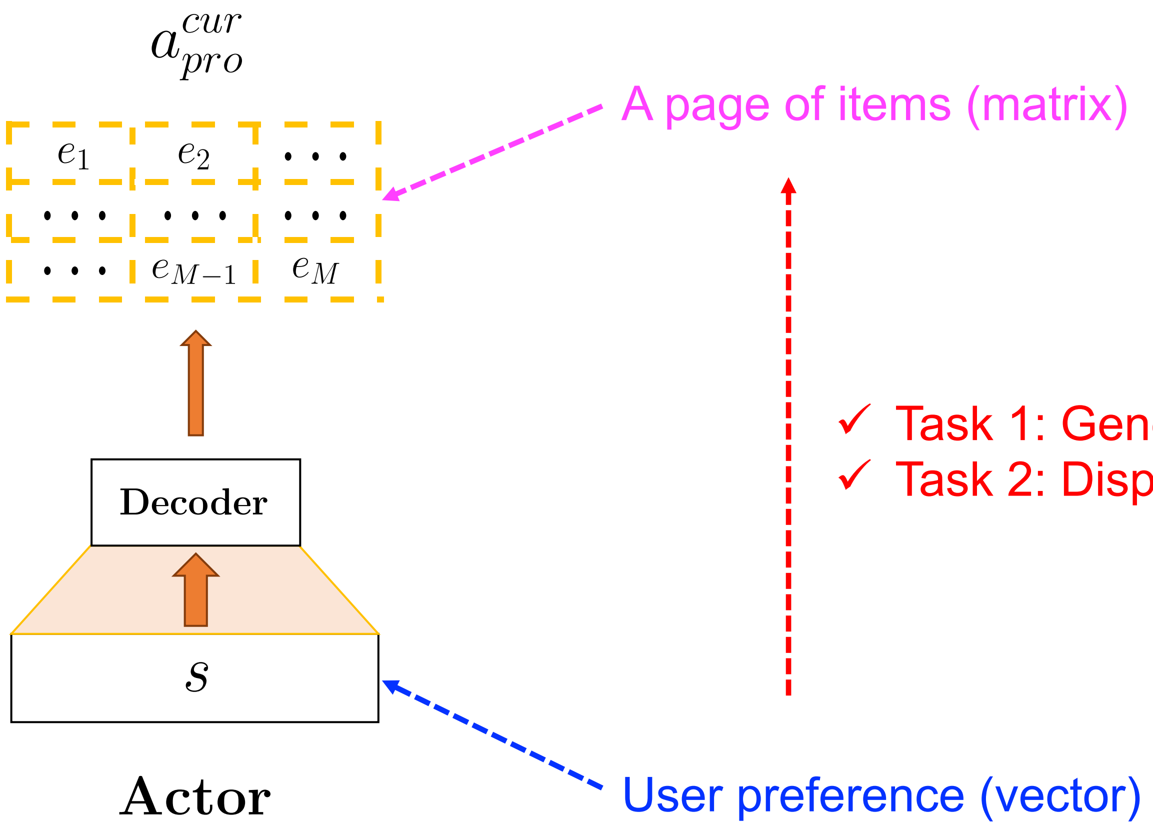
$$z_t = \sigma(W_z E_t + U_z h_{t-1})$$
$$r_t = \sigma(W_r E_t + U_r h_{t-1})$$
$$h_t = (1 - z_t) h_{t-1} + z_t \hat{h}_t$$
$$\hat{h}_t = \tanh[W E_t + U(r_t \cdot h_{t-1})]$$

GRU



Decoder

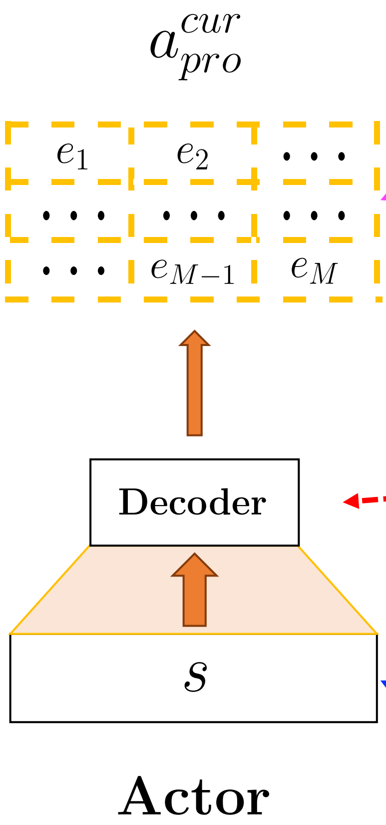
- Goal: Generating a page of items according to user's preference



- ✓ Task 1: Generating a set of items
- ✓ Task 2: Displaying items in a page

Decoder

- **Goal**: Generating a page of items according to user's preference

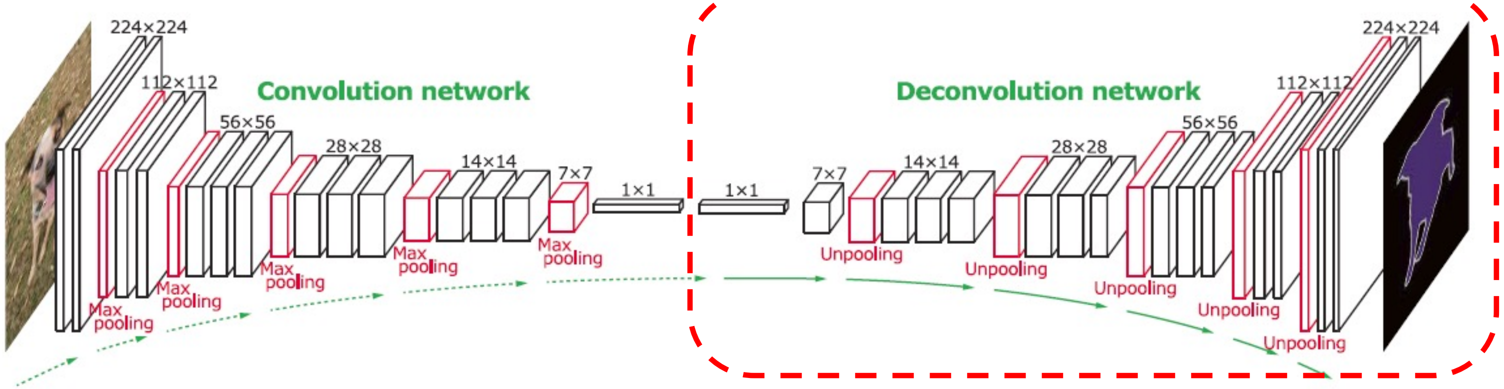


A page of items (matrix)

DeCNN

User preference (vector)

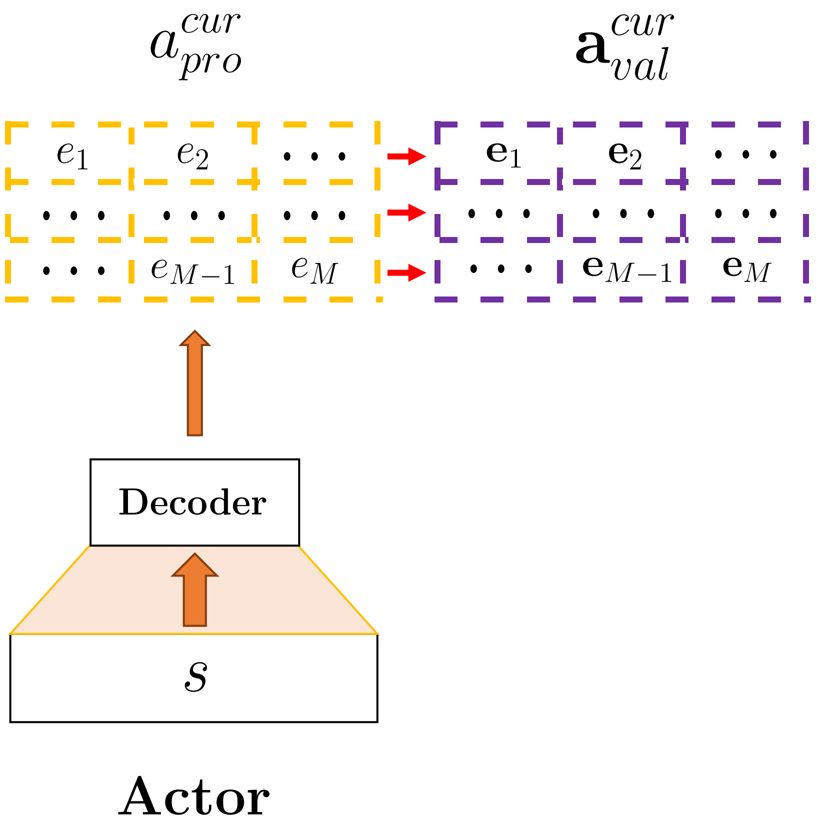
Deconvolution Neural Network



Representation (vector) recover Image (matrix)

Decoder

- Generated Embeddings → Real Embeddings



Generated Embedding

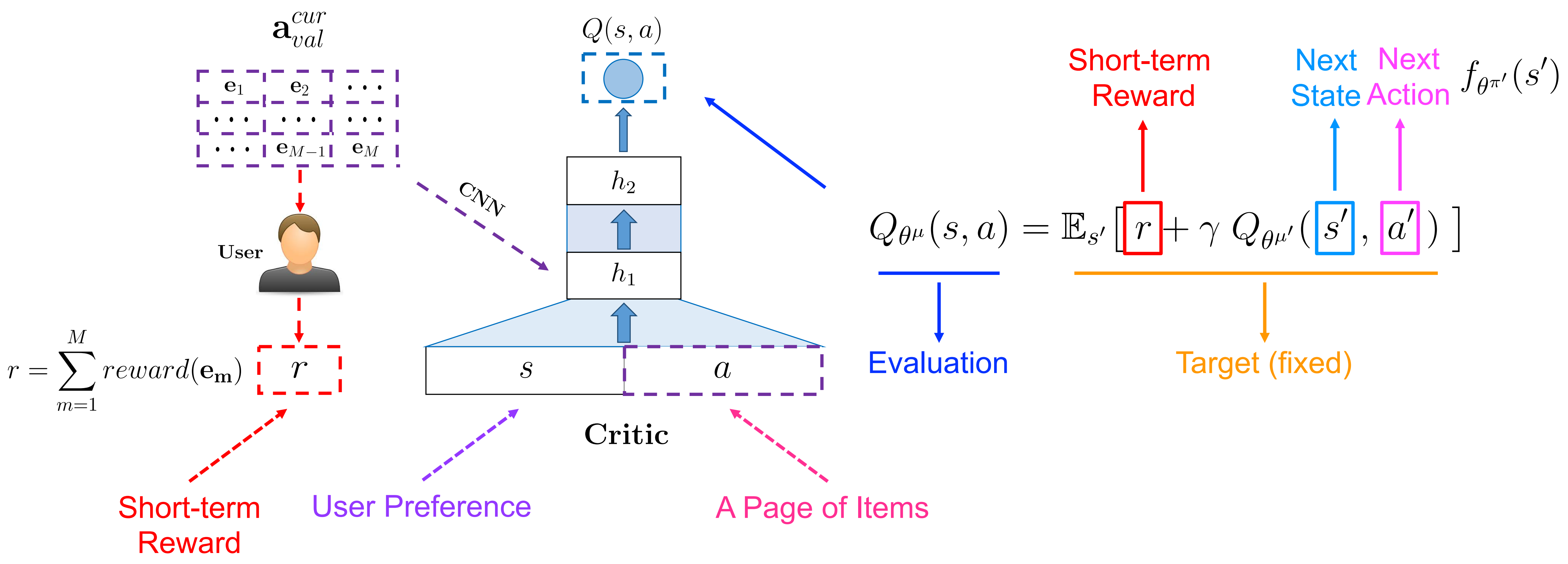
$$\boxed{\mathbf{e}_i} = \arg\max_{\mathbf{e}\in\mathcal{I}} \frac{e_i^\top \cdot \mathbf{e}}{\|e_i\|\|\mathbf{e}\|} = \underbrace{\arg\max_{\mathbf{e}\in\mathcal{I}}}_{\text{Item Space}} \boxed{e_i^\top} \cdot \frac{\boxed{\mathbf{e}}}{\|\mathbf{e}\|}$$

Real Embedding

Item Space

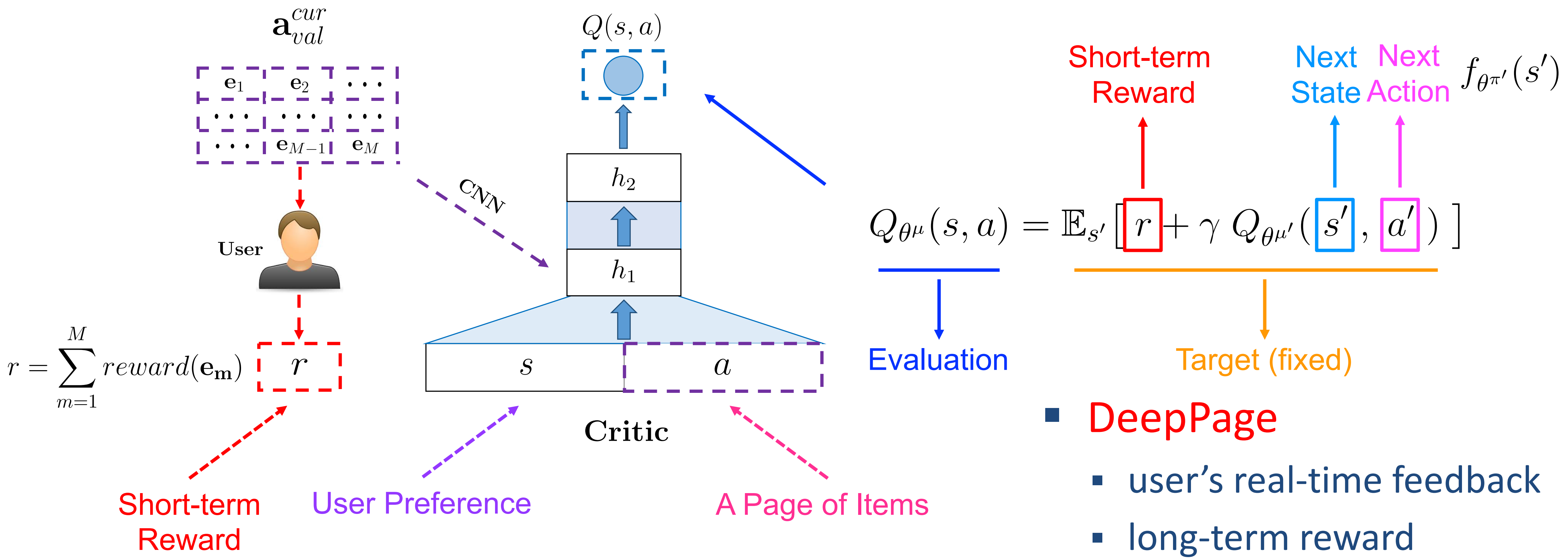
Critic Architecture

- Learning action-value function $Q(s, a)$



Critic Architecture

- Learning action-value function $Q(s, a)$



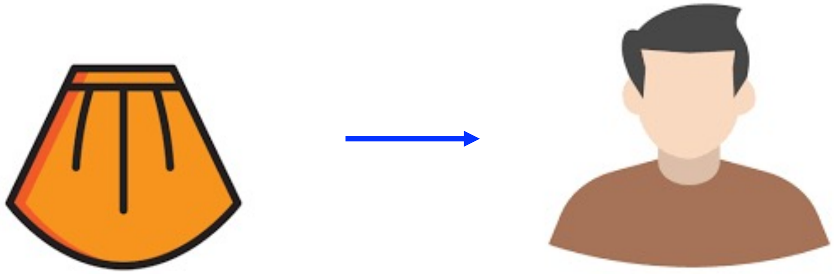
- **DeepPage**
  - user's real-time feedback
  - long-term reward
  - putting items in a page

Outline

- Recommendations in Single Scenario
  - DeepPage - Deep Reinforcement Learning for Page-wise Recommendations (RecSys'2018)
  - DEERS - Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning (KDD'2018)
  - DRN - A Deep Reinforcement Learning Framework for News Recommendation (WWW'2018)
- Recommendations in Multiple Scenarios
  - DeepChain - Whole-Chain Recommendations (CIKM'2020)
  - MA-RDPG - Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning (WWW'2018)
  - RAM - Jointly Learning to Recommend and Advertise (KDD'2020)
  - DEAR - Deep Reinforcement Learning for Online Advertising in Recommender Systems (AAAI'2021)
- Online Environment Simulator
  - UserSim - User Simulation via Supervised Generative Adversarial Network (WWW'2021)
- Surveys
  - Deep Reinforcement Learning for Search, Recommendation, and Online Advertising: A Survey (SIGWEB'2019)
  - Reinforcement Learning based Recommender Systems: A Survey (Arxiv'2021)

Why Negative Feedback?

- What users may not like
  - Positive: click or purchase
  - Negative: skip or leave
- Advantage:
  - Avoiding bad recommendation cases

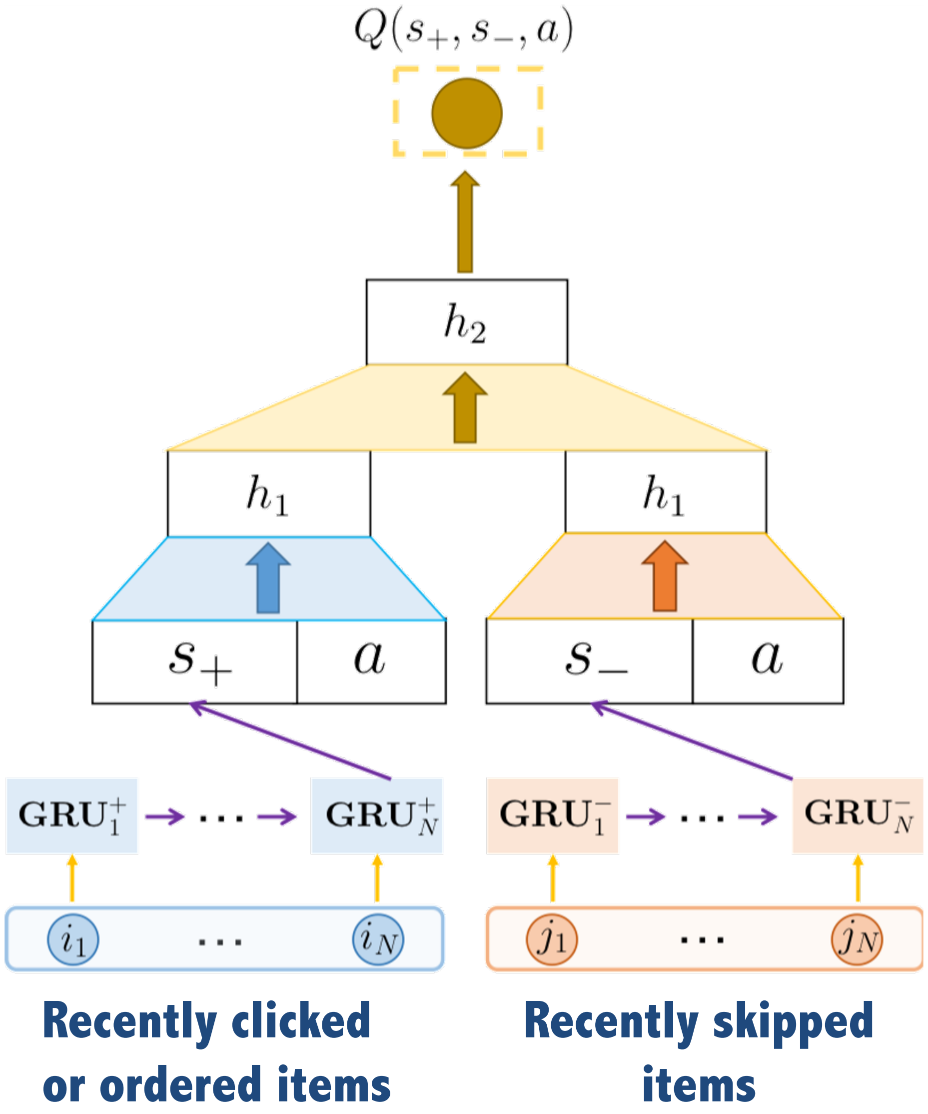


- Challenges
  - Negative feedback could bury the positive ones
  - May not be caused by users disliking them
  - Weak/wrong negative feedback can introduce noise

Novel DQN Architecture

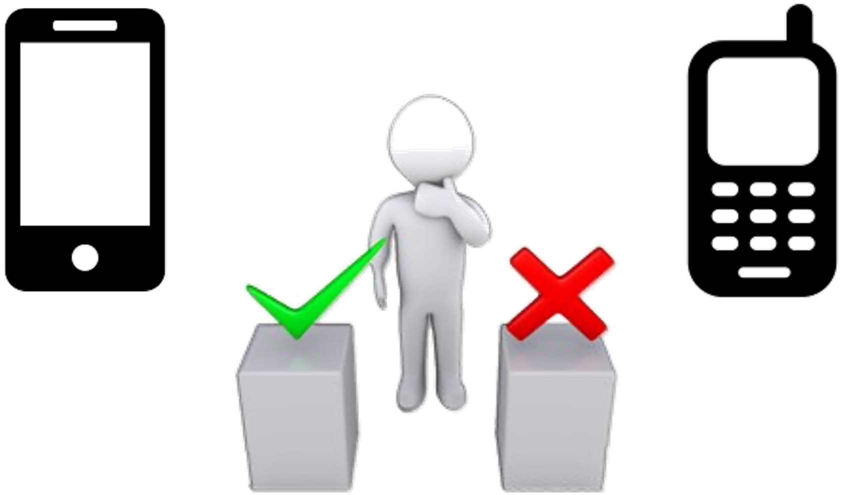
- Intuition:
  - recommend an item that is similar to the clicked/ordered items (left part)
  - while dissimilar to the skipped items (right part)

- RNN with Gated Recurrent Units (GRU) to capture users' sequential preference



Weak or Wrong Negative Feedback

- Recommender systems often recommends items belong to the same category (e.g., cell phone), while users click/order a part of them and skip others



Time	State	Item	Category	Feedback
1	s_1	a_1	A	skip
2	s_2	a_2	B	click
3	s_3	a_3	A	click
4	s_4	a_4	C	skip
5	s_5	a_5	B	skip
6	s_6	a_6	A	skip
7	s_7	a_7	C	order

- The partial order of user’s preference over these two items in category B
- At time 2, we name a5 as the competitor item of a2

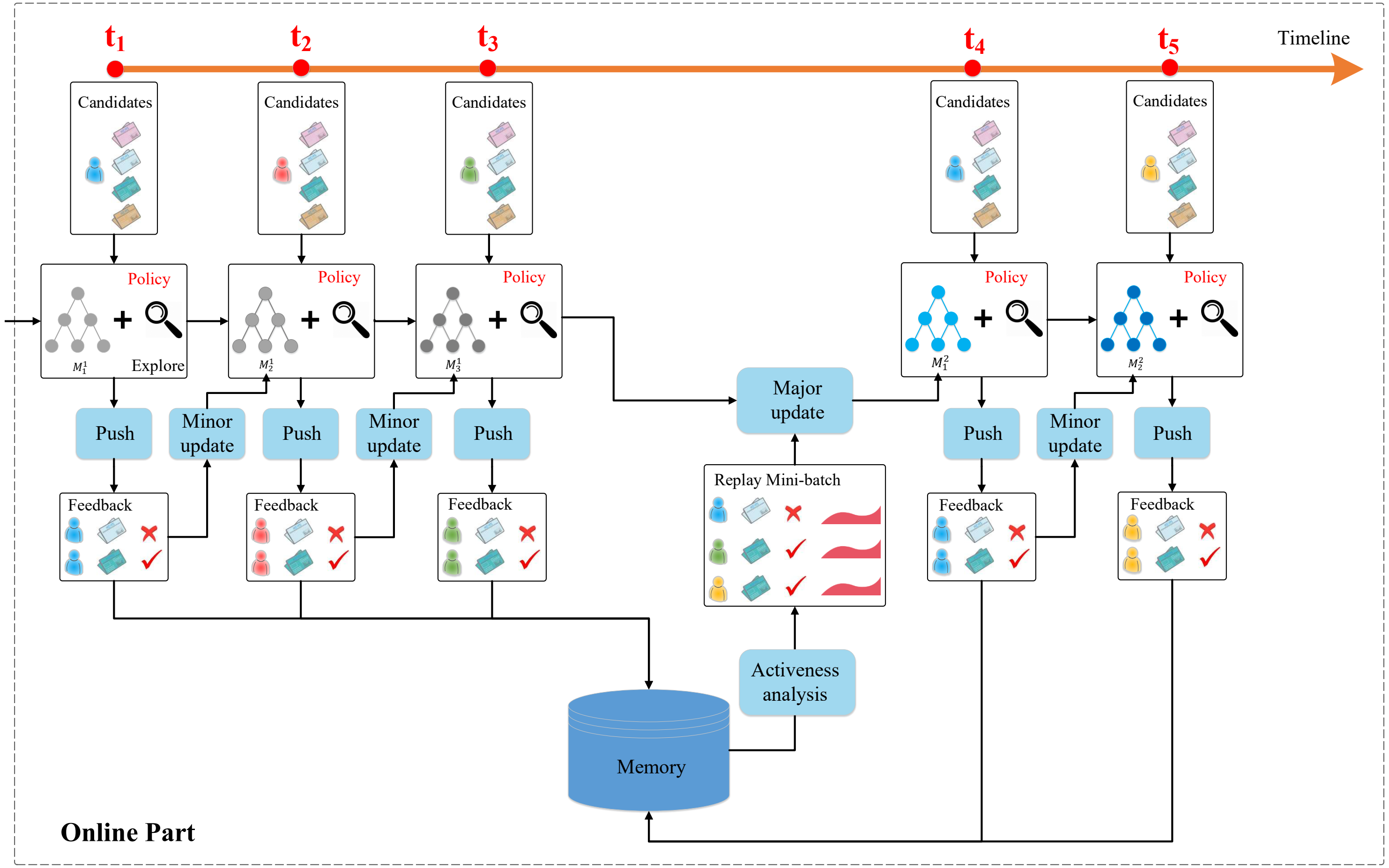
$$L(\theta) = \mathbb{E}_{s,a,r,s'}\left[\left(y - Q(s_+, s_-, a; \theta)\right)^2 - \alpha\left(Q(s_+, s_-, a; \theta) - Q(s_+, s_-, a^E; \theta)\right)^2\right]$$

Outline

- Recommendations in Single Scenario
  - DeepPage - Deep Reinforcement Learning for Page-wise Recommendations (RecSys'2018)
  - DEERS - Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning (KDD'2018)
  - DRN - A Deep Reinforcement Learning Framework for News Recommendation (WWW'2018)
- Recommendations in Multiple Scenarios
  - DeepChain - Whole-Chain Recommendations (CIKM'2020)
  - MA-RDPG - Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning (WWW'2018)
  - RAM - Jointly Learning to Recommend and Advertise (KDD'2020)
  - DEAR - Deep Reinforcement Learning for Online Advertising in Recommender Systems (AAAI'2021)
- Online Environment Simulator
  - UserSim - User Simulation via Supervised Generative Adversarial Network (WWW'2021)
- Surveys
  - Deep Reinforcement Learning for Search, Recommendation, and Online Advertising: A Survey (SIGWEB'2019)
  - Reinforcement Learning based Recommender Systems: A Survey (Arxiv'2021)

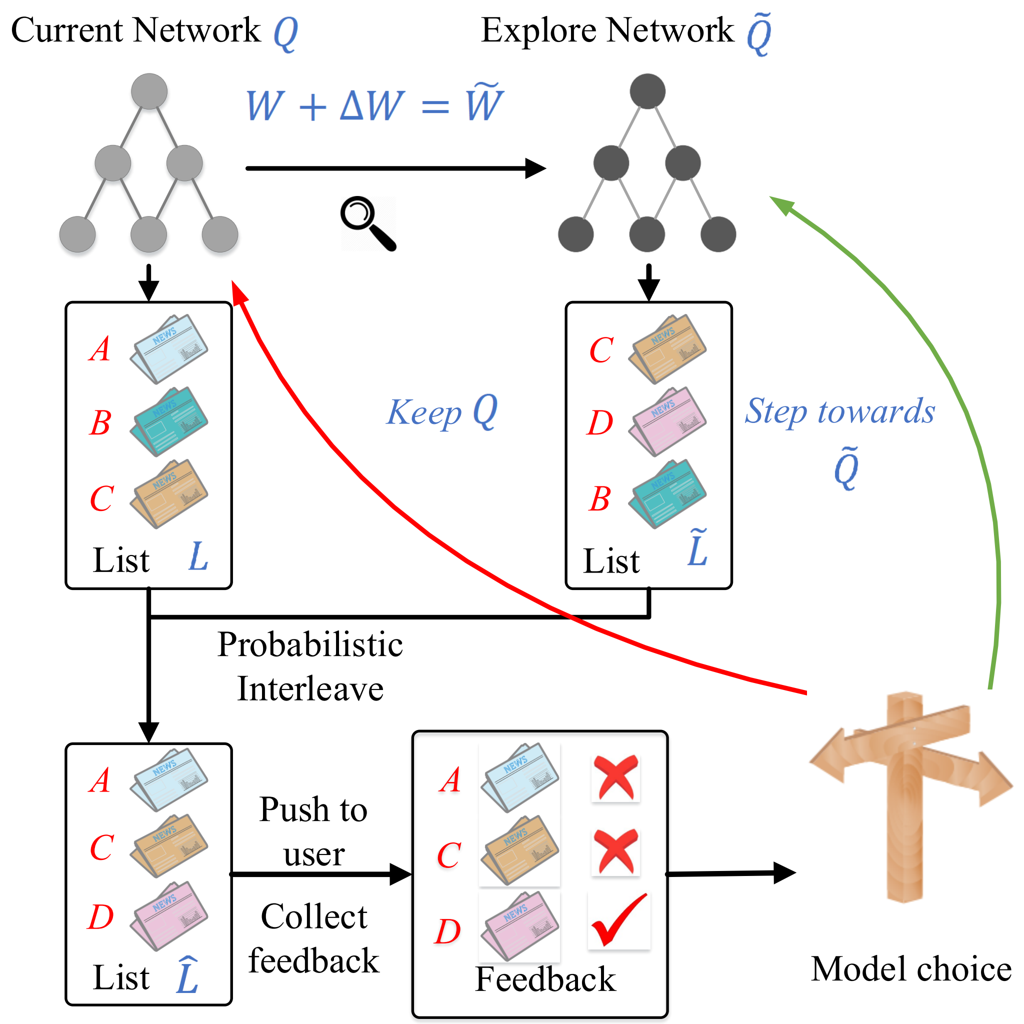
Framework

- Push
- Feedback
- Minor Update
- Major Update



Effective Exploration

- Random exploration
  - Harm the user experience in short term
- Multi-armed Bandit
  - Large variance
  - Long time to converge
- Steps
  - Get recommendation from Q and $\tilde{Q}$
  - Probabilistic interleave these two lists
  - Get feedback from user and compare the performance of two network
  - If $\tilde{Q}$ performs better, update Q towards it

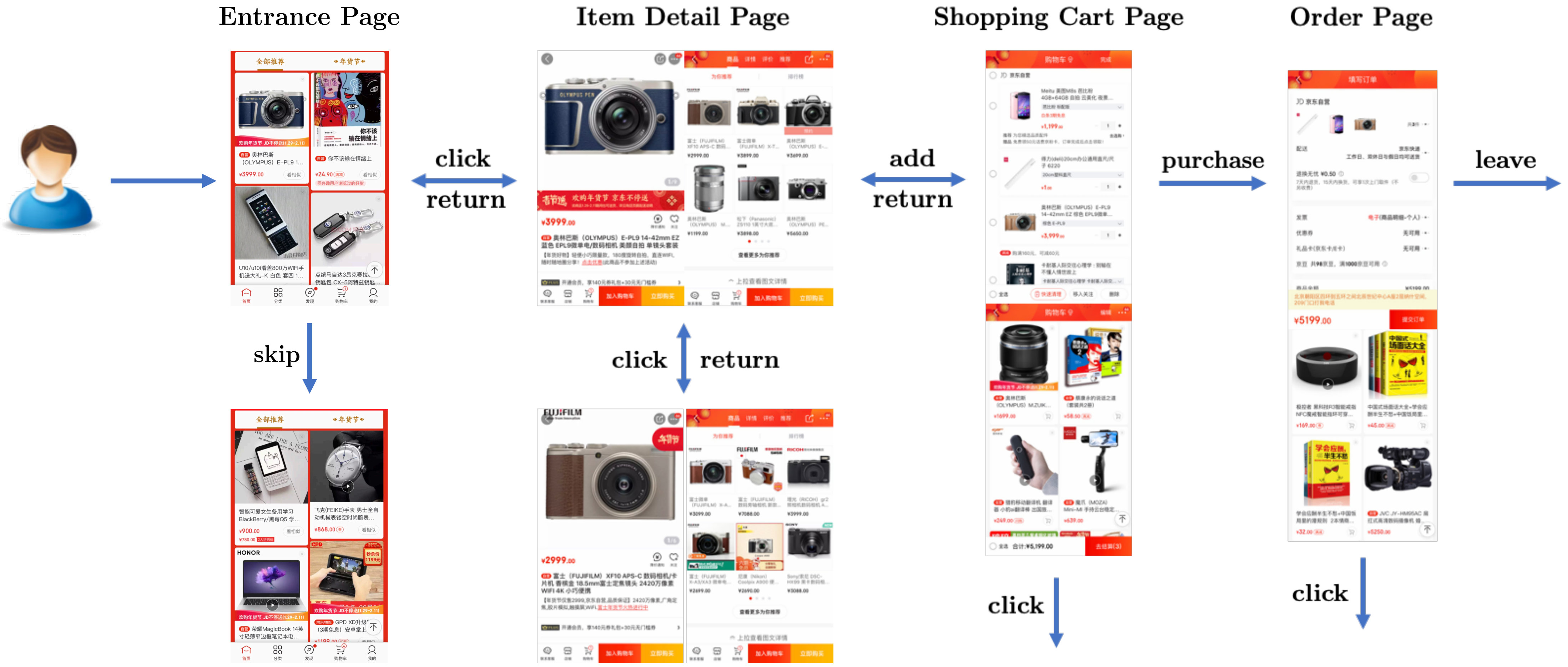


Outline

- Recommendations in Single Scenario
  - DeepPage - Deep Reinforcement Learning for Page-wise Recommendations (RecSys'2018)
  - DEERS - Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning (KDD'2018)
  - DRN - A Deep Reinforcement Learning Framework for News Recommendation (WWW'2018)
- Recommendations in Multiple Scenarios
  - DeepChain - Whole-Chain Recommendations (CIKM'2020)
  - MA-RDPG - Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning (WWW'2018)
  - RAM - Jointly Learning to Recommend and Advertise (KDD'2020)
  - DEAR - Deep Reinforcement Learning for Online Advertising in Recommender Systems (AAAI'2021)
- Online Environment Simulator
  - UserSim - User Simulation via Supervised Generative Adversarial Network (WWW'2021)
- Surveys
  - Deep Reinforcement Learning for Search, Recommendation, and Online Advertising: A Survey (SIGWEB'2019)
  - Reinforcement Learning based Recommender Systems: A Survey (Arxiv'2021)

Background

- Users sequentially interact with multiple scenarios
  - Different scenario has different objective



Motivation

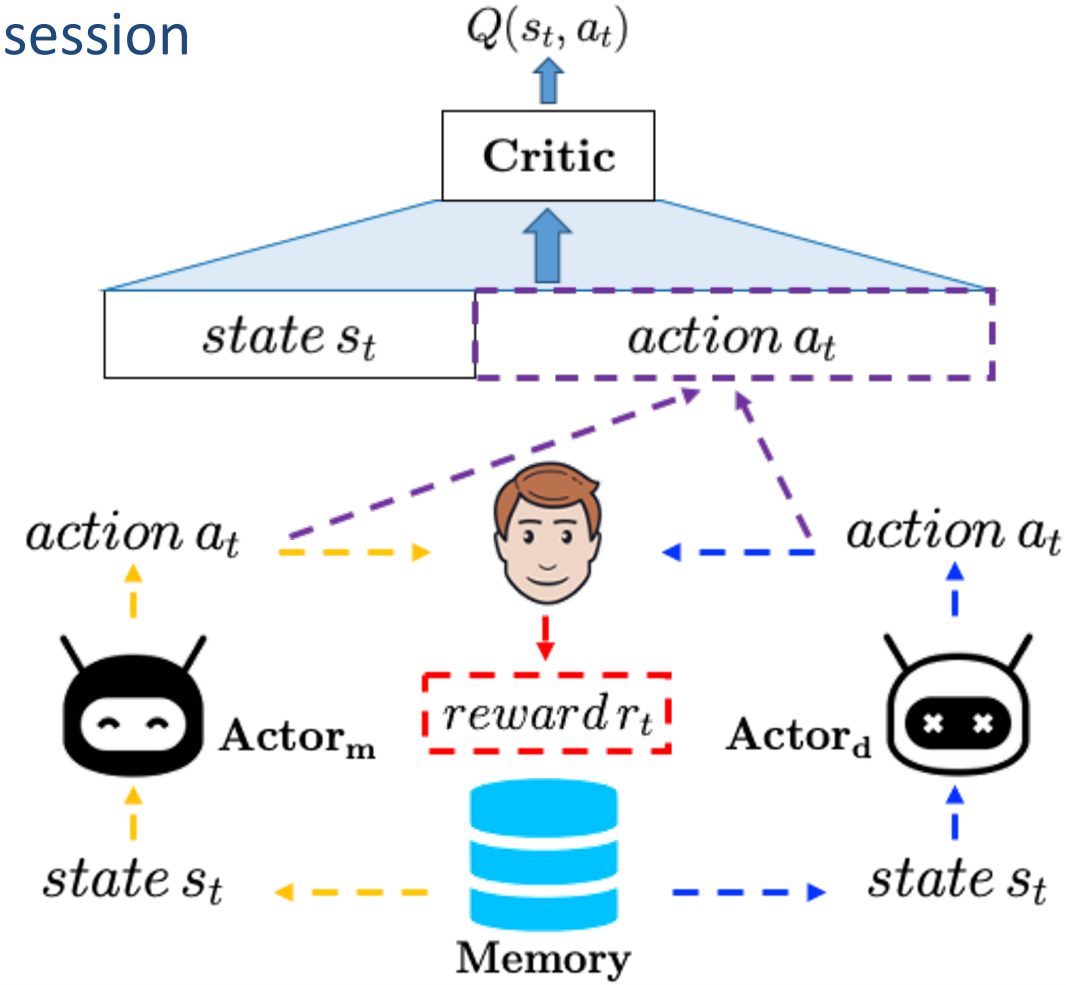
- Optimizing each recommender agent for each scenario
  - Ignoring sequential dependency
  - Missing information
  - Sub-optimal overall objective

Entrance Page

Item Detail Page

Whole-Chain Recommendation

- Goal
  - Jointly optimizing multiple recommendation strategies
  - Maximizing the overall performance of the whole session
- Advantages
  - Agents are sequentially activated
  - Agents share the same memory
  - Agents work collaboratively
- Actor-Critic
  - Actor: recommender agent in one scenario
  - Critic: controlling actors



Entrance Page

$$
\begin{aligned}
y_t = \Big[& p_m^s(s_t, a_t) \cdot \gamma Q_{\mu'}\big(s_{t+1}, \pi'_m(s_{t+1})\big) \\
+ \; & p_m^c(s_t, a_t) \cdot \Big(r_t + \gamma Q_{\mu'}\big(s_{t+1}, \pi'_d(s_{t+1})\big)\Big) \\
+ \; & p_m^l(s_t, a_t) \cdot r_t \Big] \mathbf{1}_{\mathbf{m}}
\end{aligned}
$$

- 1st row: skip behavior
- 2nd row: click behavior
- 3rd row: leave behavior

Optimization

Entrance Page

$$
\begin{aligned}
y_t = \Big[& p_m^s(s_t, a_t) \cdot \gamma Q_{\mu'}(s_{t+1}, \pi_m'(s_{t+1})) \\
+ \; & p_m^c(s_t, a_t) \cdot \big(r_t + \gamma Q_{\mu'}(s_{t+1}, \pi_d'(s_{t+1}))\big) \\
+ \; & p_m^l(s_t, a_t) \cdot r_t \Big] \mathbf{1_m} \\
+ \Big[& p_d^c(s_t, a_t) \cdot \big(r_t + \gamma Q_{\mu'}(s_{t+1}, \pi_d'(s_{t+1}))\big) \\
+ \; & p_d^s(s_t, a_t) \cdot \gamma Q_{\mu'}(s_{t+1}, \pi_m'(s_{t+1})) \\
+ \; & p_d^l(s_t, a_t) \cdot r_t \Big] \mathbf{1_d}
\end{aligned}
$$

Item Detail Page

Why Model-based RL?

- Advantages
  - Reducing training data amount requirement
  - Performing accurate optimization of the Q-function

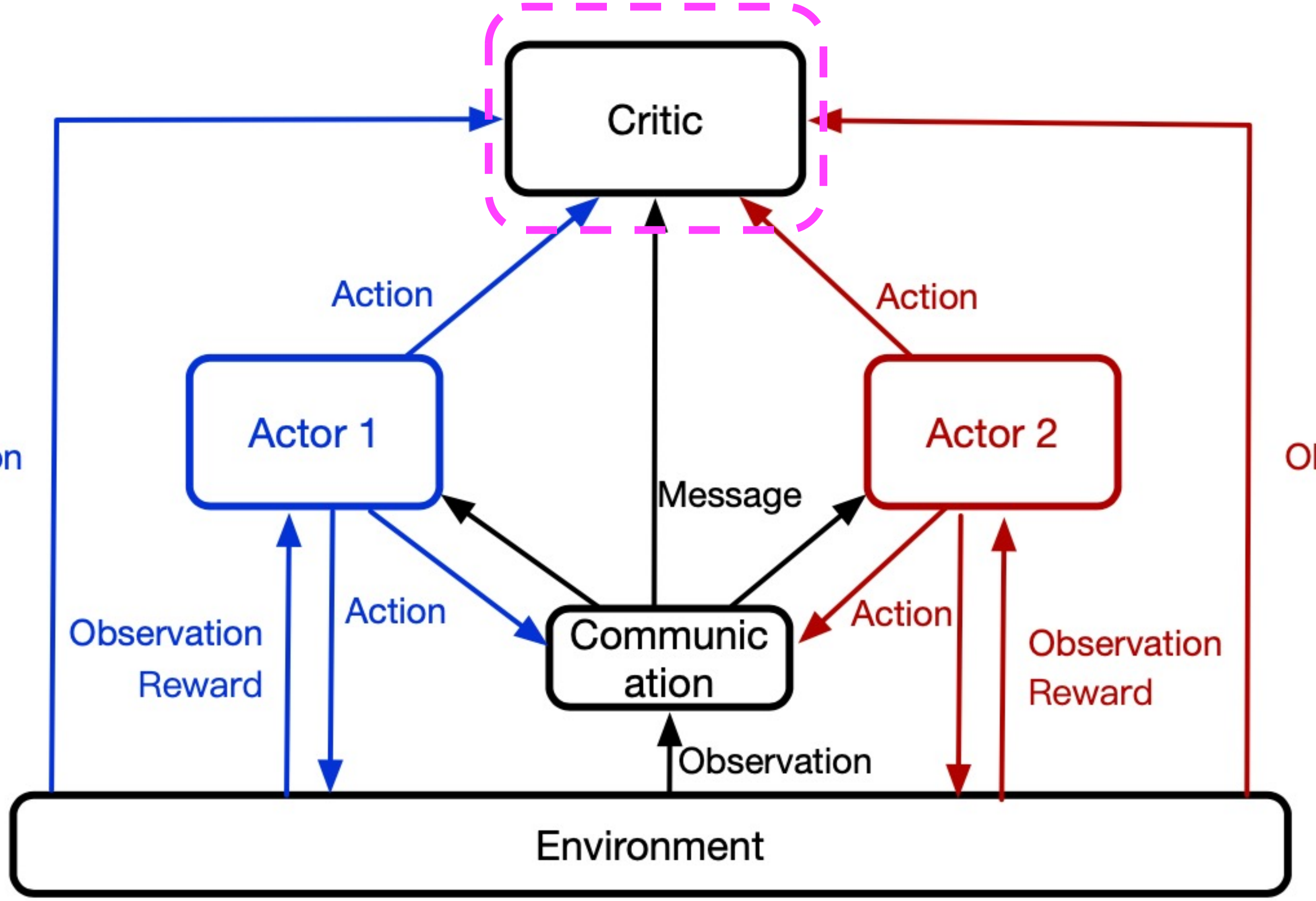
$$
\begin{aligned}
y_t = \Big[& p_m^s(s_t, a_t) \cdot \gamma Q_{\mu'}(s_{t+1}, \pi'_m(s_{t+1})) \\
+ \ & p_m^c(s_t, a_t) \cdot \big(r_t + \gamma Q_{\mu'}(s_{t+1}, \pi'_d(s_{t+1}))\big) \\
+ \ & p_m^l(s_t, a_t) \cdot r_t \Big] \mathbf{1_m} \\
+ \Big[& p_d^c(s_t, a_t) \cdot \big(r_t + \gamma Q_{\mu'}(s_{t+1}, \pi'_d(s_{t+1}))\big) \\
+ \ & p_d^s(s_t, a_t) \cdot \gamma Q_{\mu'}(s_{t+1}, \pi'_m(s_{t+1})) \\
+ \ & p_d^l(s_t, a_t) \cdot r_t \Big] \mathbf{1_d}
\end{aligned}
$$

Model-based

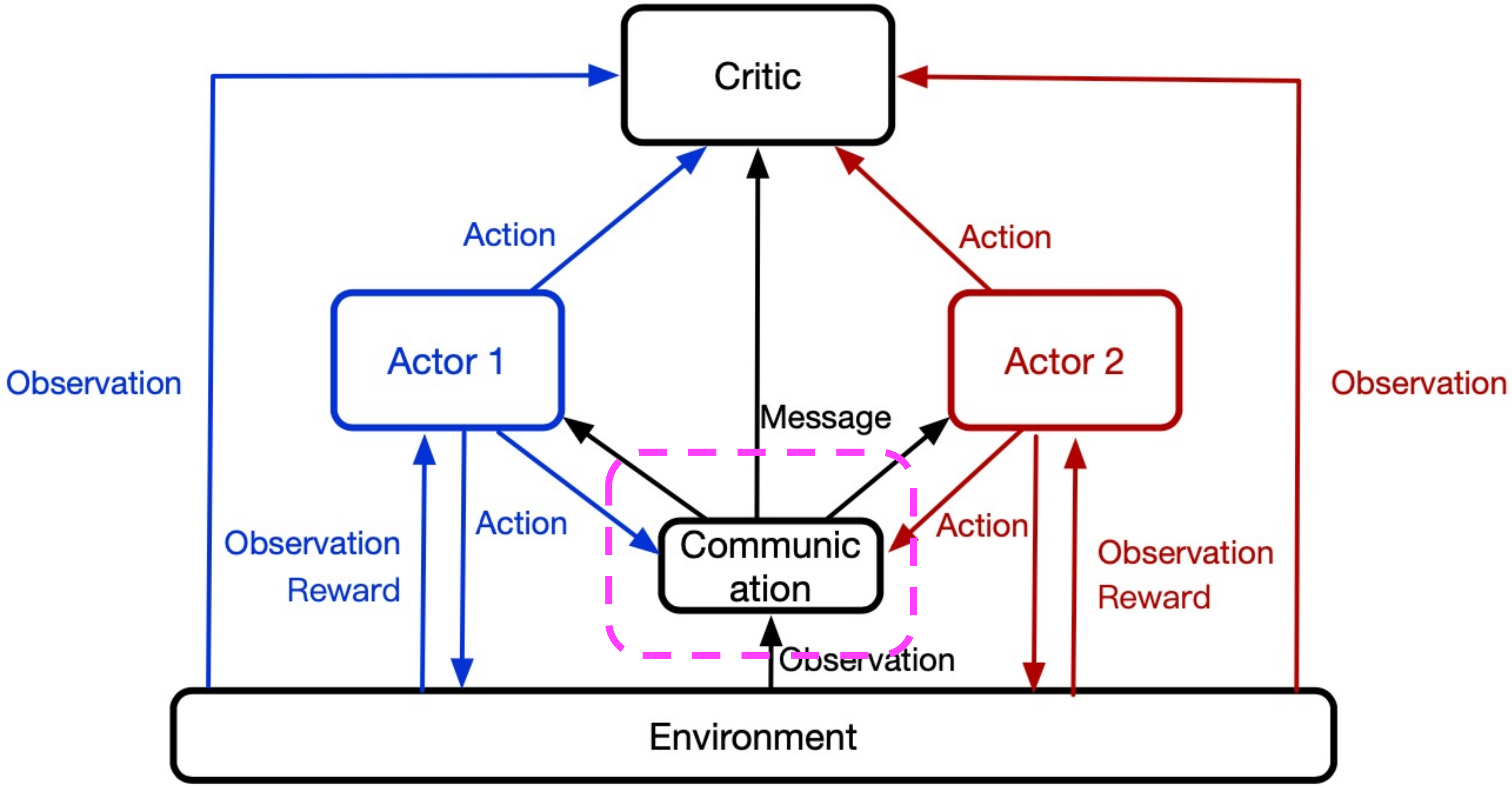
Outline

- Recommendations in Single Scenario
  - DeepPage - Deep Reinforcement Learning for Page-wise Recommendations (RecSys'2018)
  - DEERS - Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning (KDD'2018)
  - DRN - A Deep Reinforcement Learning Framework for News Recommendation (WWW'2018)
- Recommendations in Multiple Scenarios
  - DeepChain - Whole-Chain Recommendations (CIKM'2020)
  - MA-RDPG - Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning (WWW'2018)
  - RAM - Jointly Learning to Recommend and Advertise (KDD'2020)
  - DEAR - Deep Reinforcement Learning for Online Advertising in Recommender Systems (AAAI'2021)
- Online Environment Simulator
  - UserSim - User Simulation via Supervised Generative Adversarial Network (WWW'2021)
- Surveys
  - Deep Reinforcement Learning for Search, Recommendation, and Online Advertising: A Survey (SIGWEB'2019)
  - Reinforcement Learning based Recommender Systems: A Survey (Arxiv'2021)

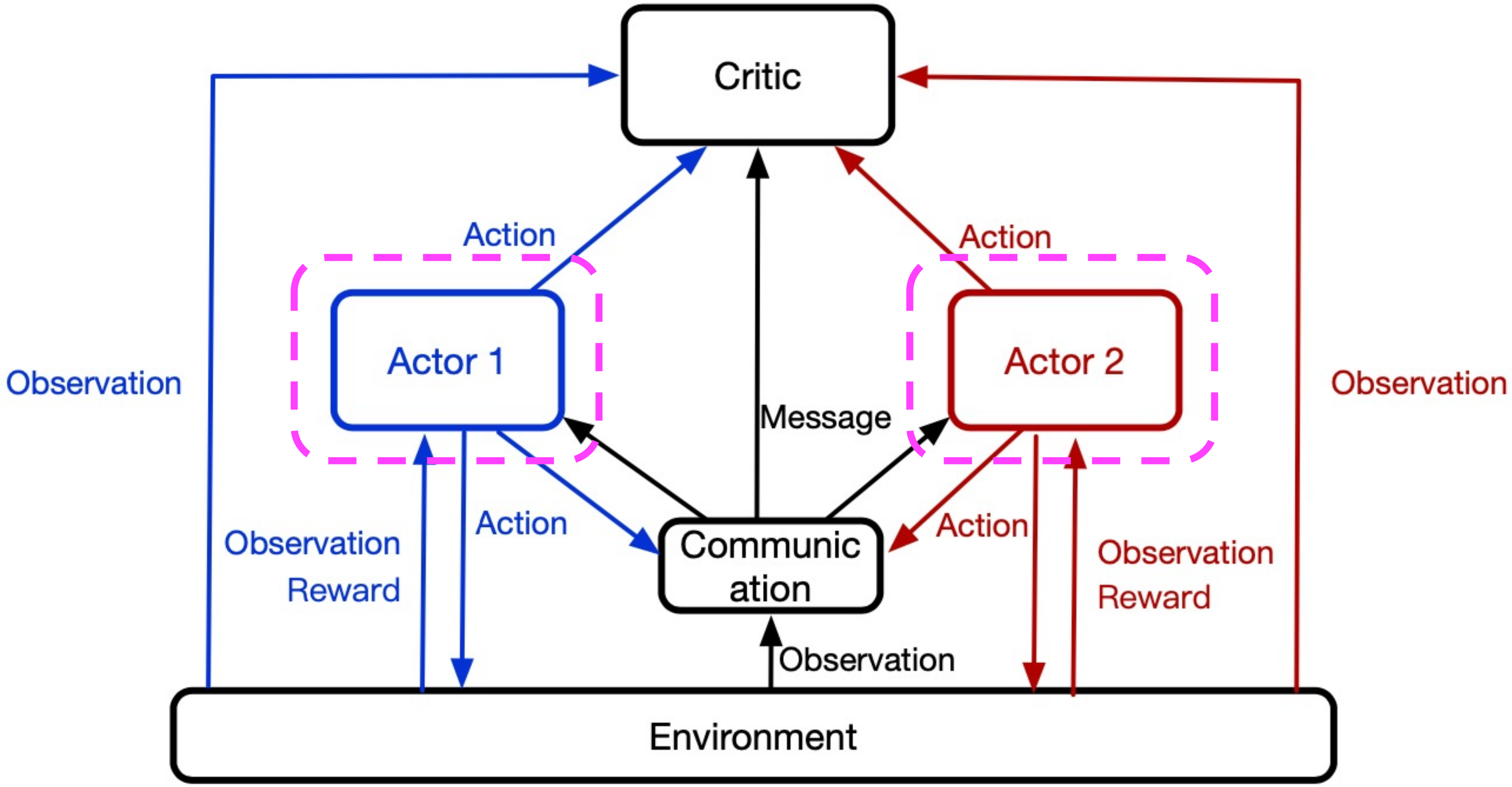
Overall Model Architecture



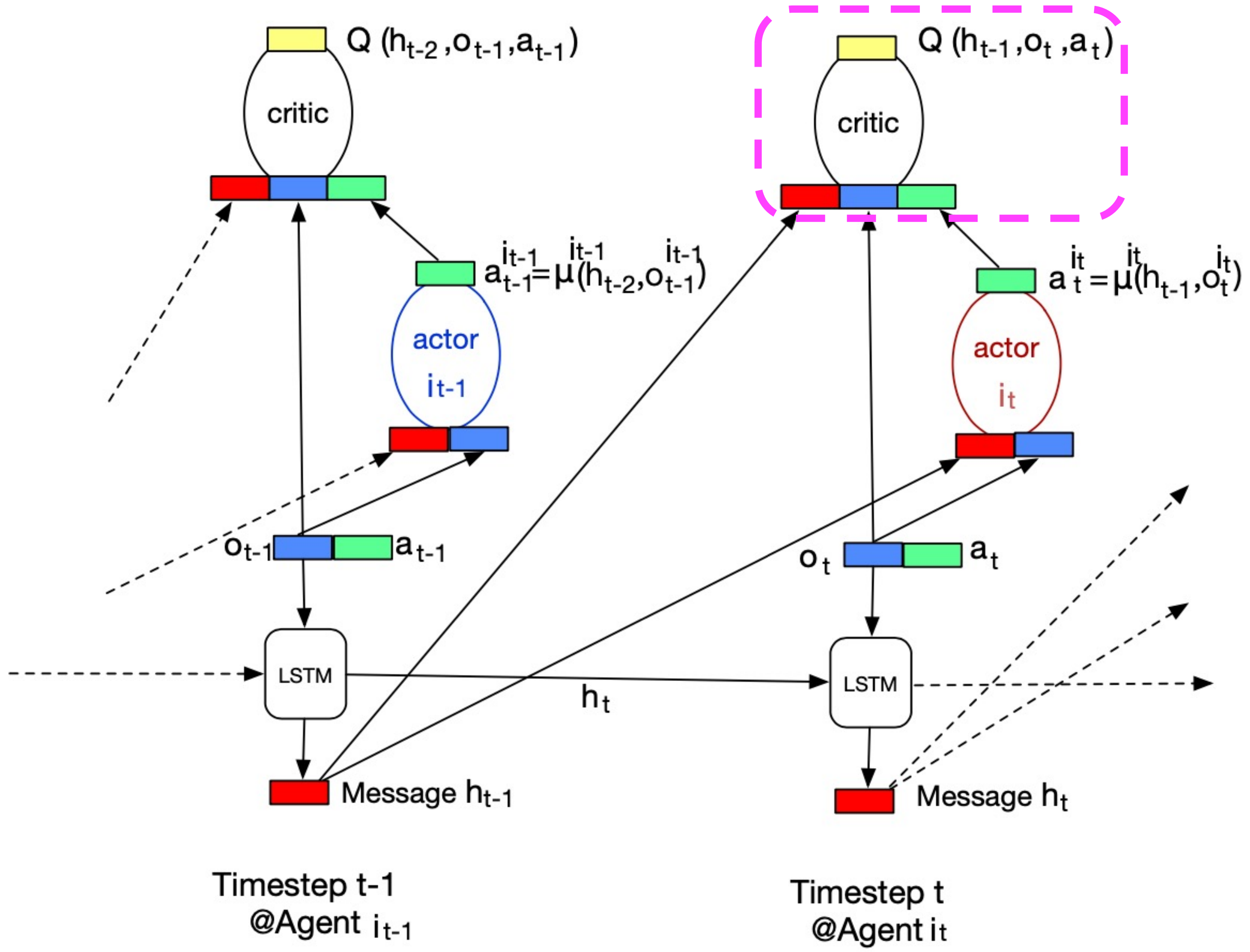
Overall Model Architecture



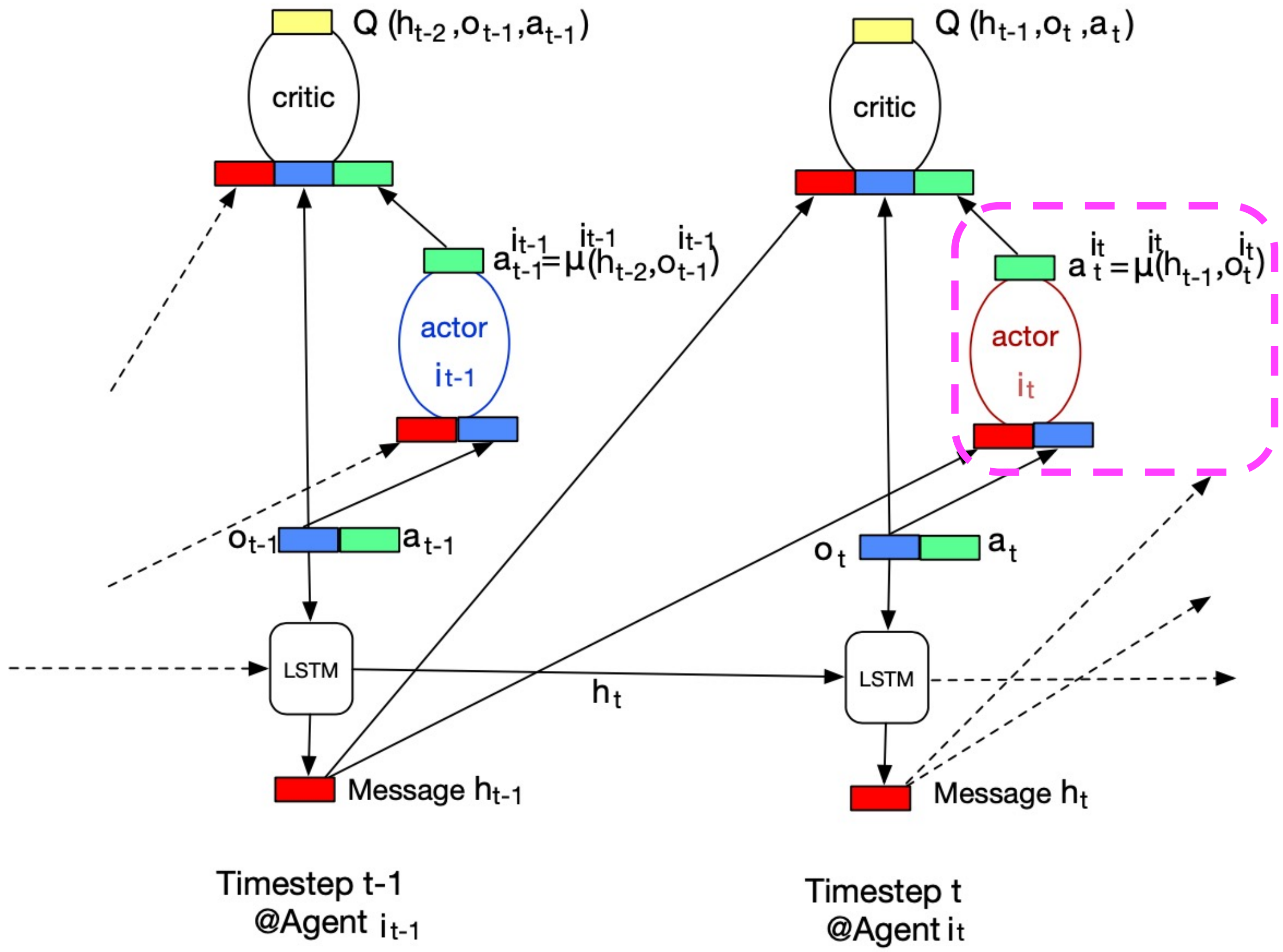
Overall Model Architecture



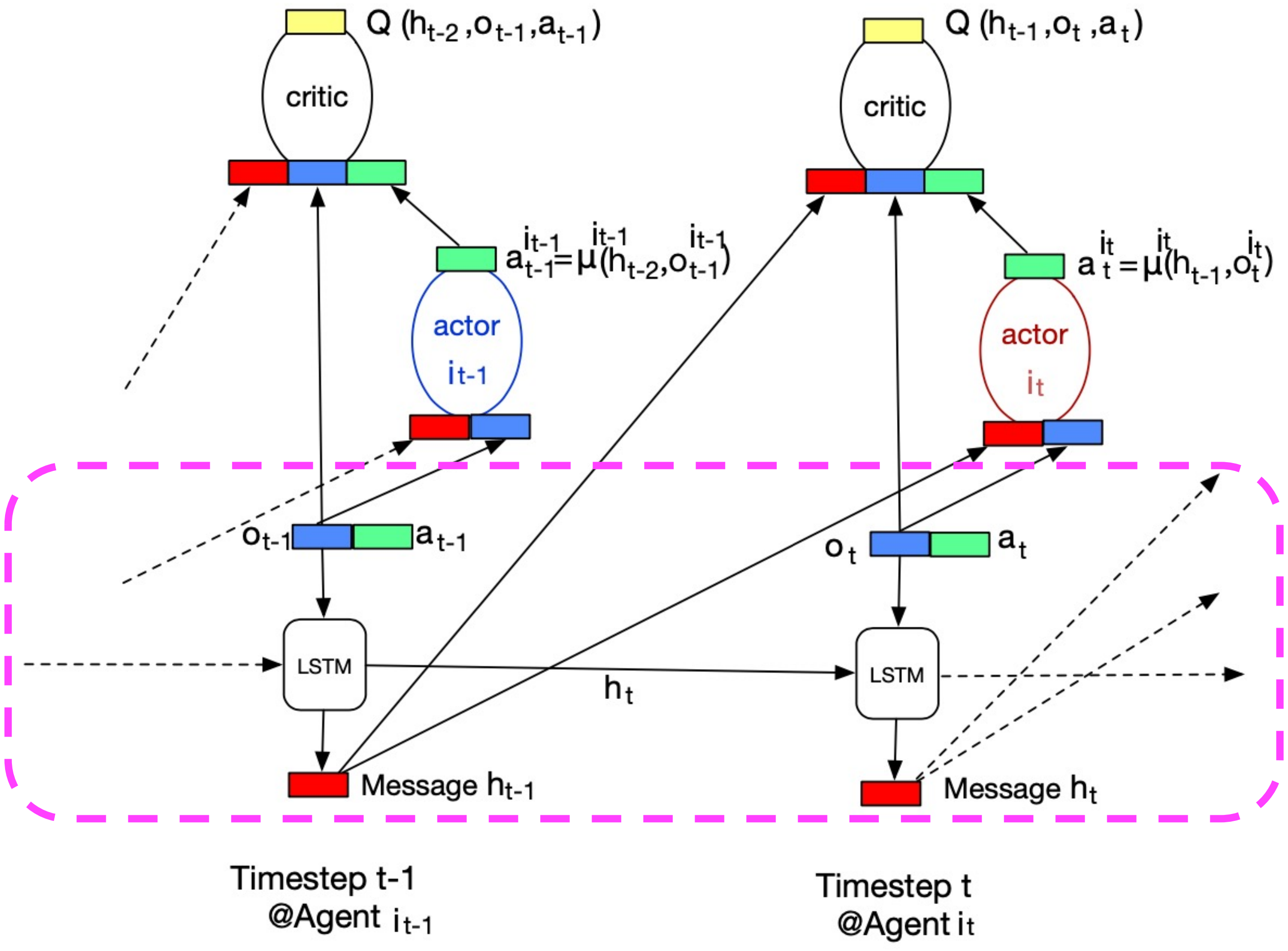
Detailed Structure of MA-RDPG



Detailed Structure of MA-RDPG



Detailed Structure of MA-RDPG

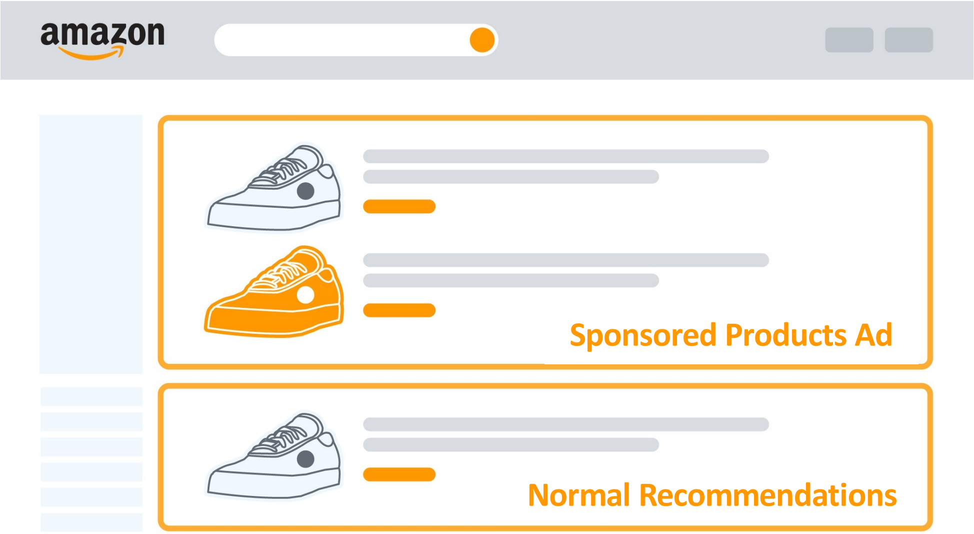


Outline

- Recommendations in Single Scenario
  - DeepPage - Deep Reinforcement Learning for Page-wise Recommendations (RecSys'2018)
  - DEERS - Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning (KDD'2018)
  - DRN - A Deep Reinforcement Learning Framework for News Recommendation (WWW'2018)
- Recommendations in Multiple Scenarios
  - DeepChain - Whole-Chain Recommendations (CIKM'2020)
  - MA-RDPG - Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning (WWW'2018)
  - RAM - Jointly Learning to Recommend and Advertise (KDD'2020)
  - DEAR - Deep Reinforcement Learning for Online Advertising in Recommender Systems (AAAI'2021)
- Online Environment Simulator
  - UserSim - User Simulation via Supervised Generative Adversarial Network (WWW'2021)
- Surveys
  - Deep Reinforcement Learning for Search, Recommendation, and Online Advertising: A Survey (SIGWEB'2019)
  - Reinforcement Learning based Recommender Systems: A Survey (Arxiv'2021)

Reinforcement Learning for Advertisements

- Goal: maximizing the advertising impression revenue from advertisers
  - Assigning the right ads to the right users at the right place



- Reinforcement learning for advertisements
  - Continuously updating the advertising strategies & maximizing the long-term revenue

Reinforcement Learning for Advertisements

- Challenges:
  - Different teams, goals and models → suboptimal overall performance

Advertising Revenue

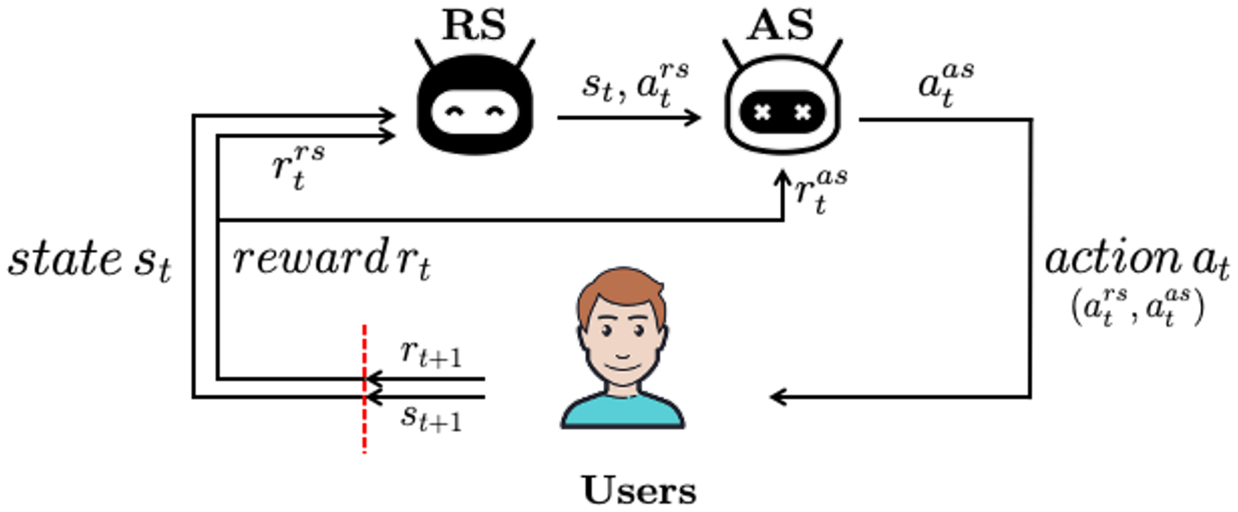
VS

User Experience

- Goal:
  - Jointly optimizing advertising revenue and user experience
  - KDD’2020, AAAI’2021

Reinforcement Learning Framework

- Two-level Deep Q-networks:
  - first-level: recommender system (RS)
  - second-level: advertising system (AS)

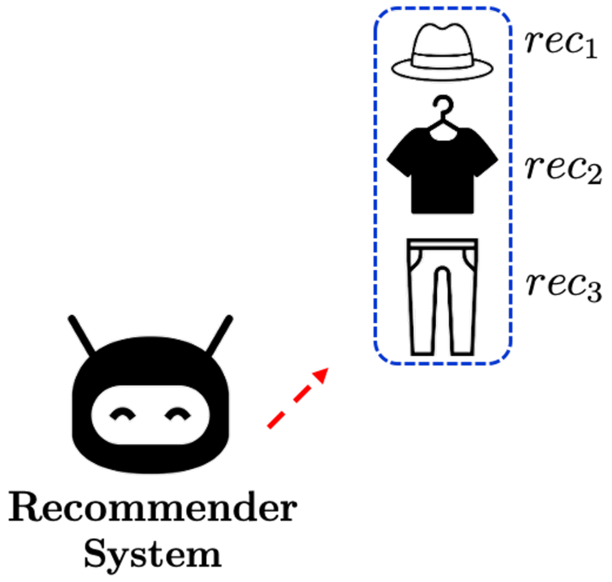


  - State: rec/ads browsing history
  - Action: $a_t = (a_t^{rs}, a_t^{as})$
  - Reward: $r_t(s_t, a_t^{rs})\ \ and\ \ r_t(s_t, a_t^{as})$
  - Transition: $s_t\ \ to\ \ s_{t+1}$

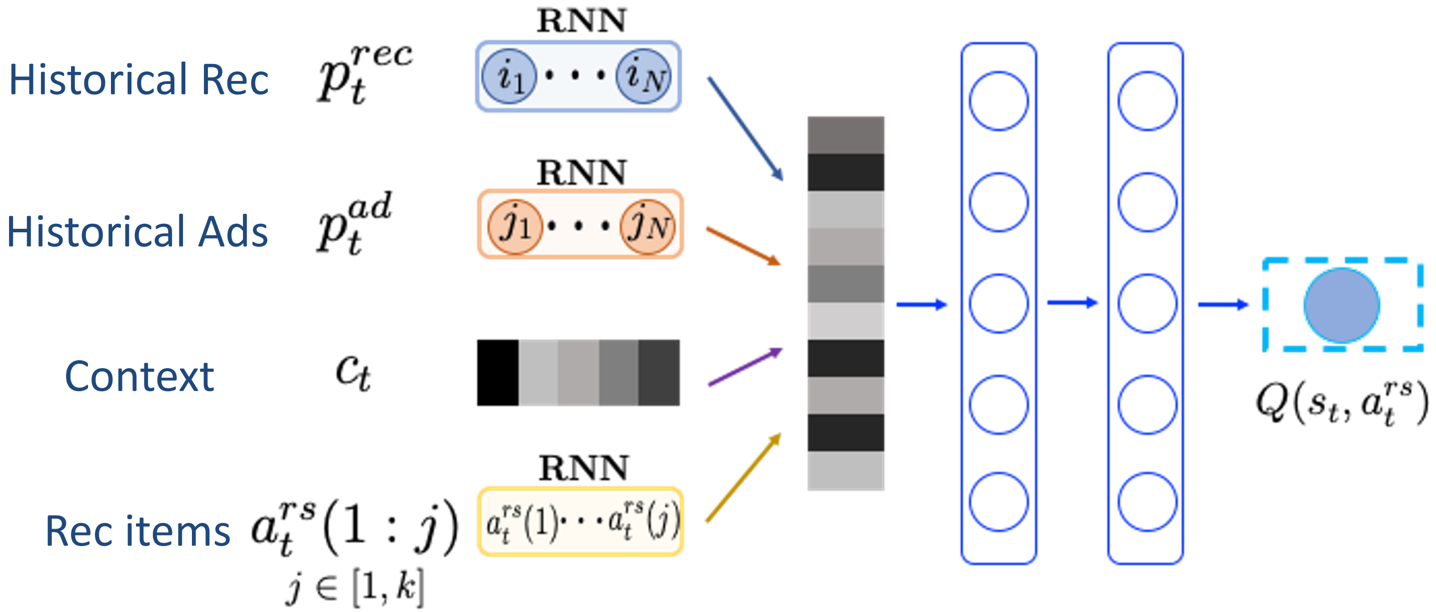
Recommender System

- Goal: long-term user experience or engagement

- Challenge: combinatorial action space



Cascading DQN for RS



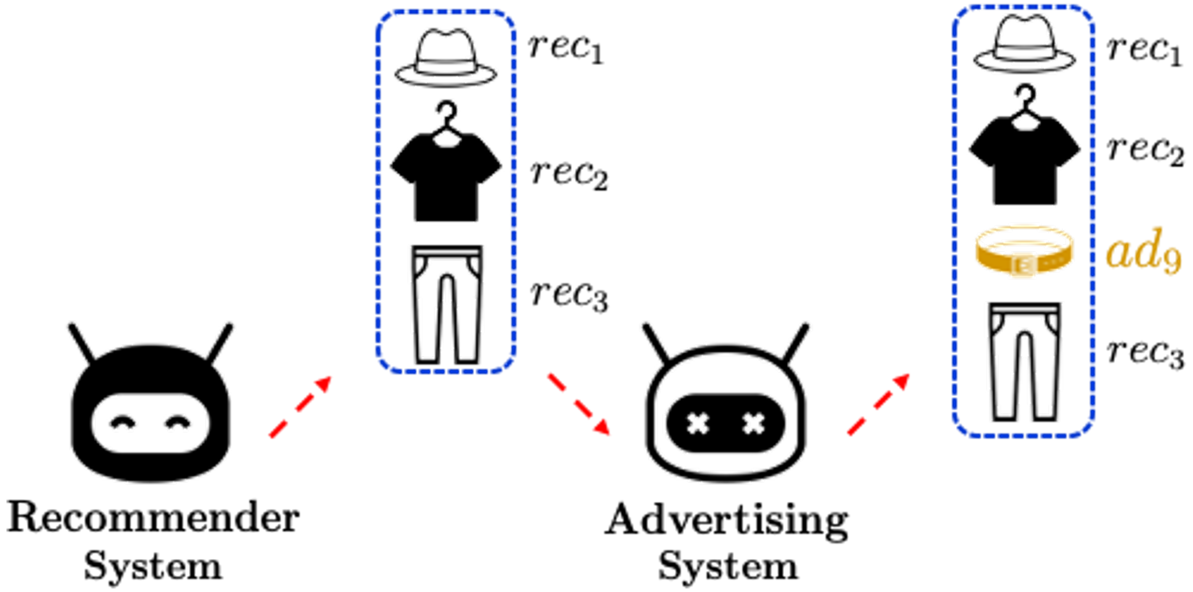
$$O\binom{N}{k} \rightarrow O(kN)$$

N: number of candidate items
k: length of rec-list

Advertising System

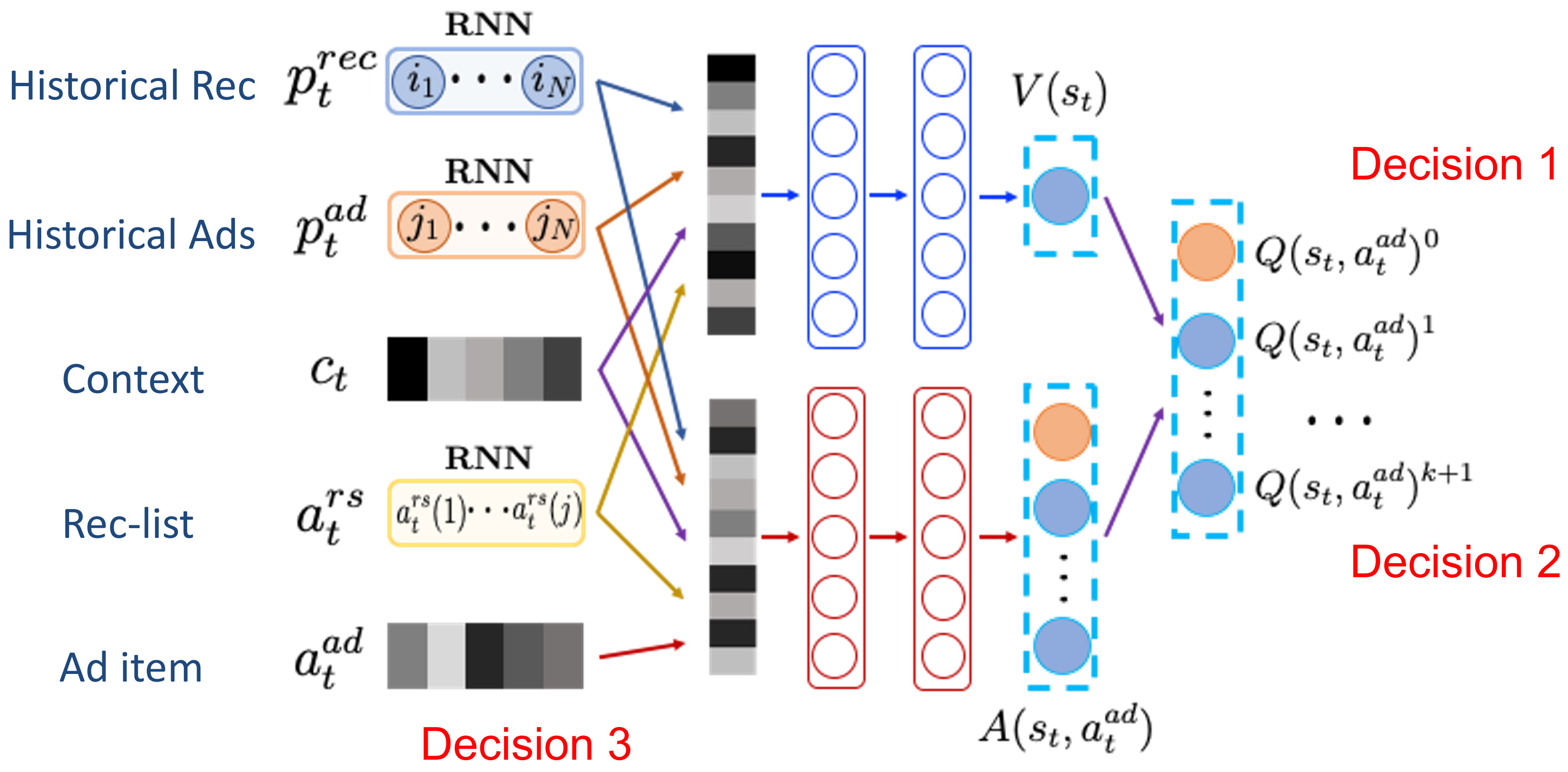
- Goal:
  - maximize the advertising revenue
  - minimize the negative influence of ads on user experience

- Decisions:
  - interpolate an ad?
  - the optimal location
  - the optimal ad



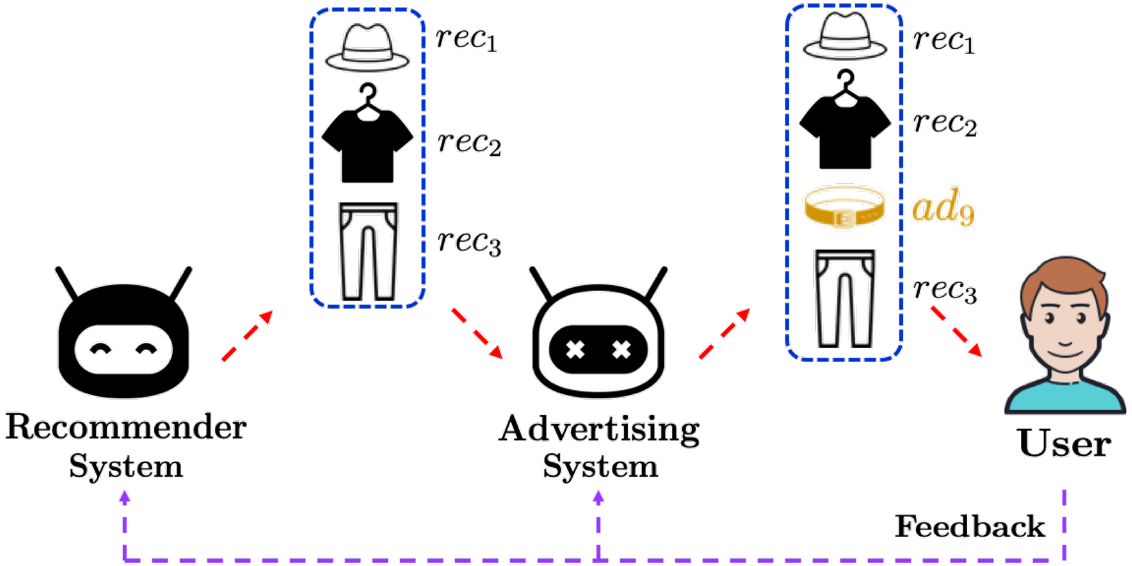
Novel DQN for AS

- Three decisions:
  1. interpolate an ad?
  2. the optimal location
  3. the optimal ad



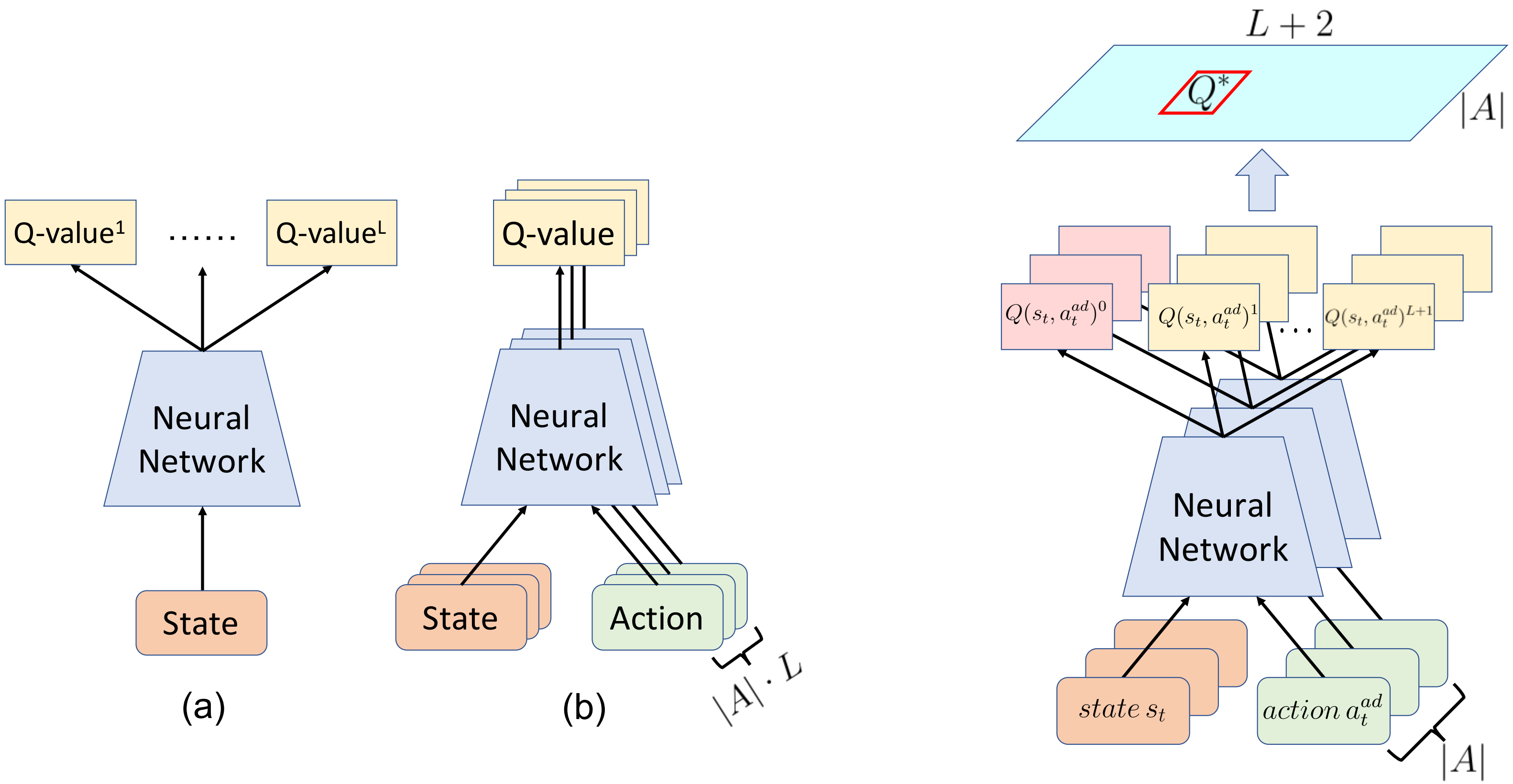
Systems Update

- Target User:
  - browses the mixed rec-ads list
  - provides her/his feedback



Advantage

- The first individual DQN architecture that can simultaneously evaluate the Q-values of multiple levels' related actions



Experiments

- Metrics:
  - user dwelling time
  - number of videos browsed
  - advertising revenue

Tiktok short video dataset

Object	Quantity
# session	1,000,000
# user	188,409
# normal video	17,820,066
# ad video	10,806,778
rec-list with ad	55.23%

Overall performace

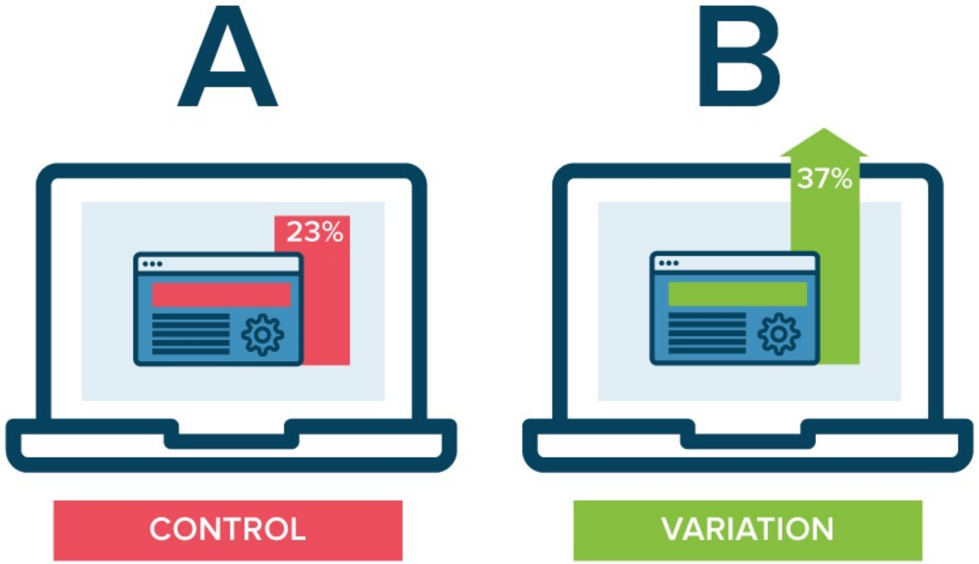
Metrics	Values	Algorithms					
		W&D	DFM	GRU	DRQN	RAM-l	RAM-n
R^{rs}	value	17.61	17.95	18.56	18.99	**19.61**	19.49
	improv.(%)	11.35	9.25	5.66	3.26	-	0.61
	p-value	0.000	0.000	0.000	0.000	-	0.006
R^{as}	value	8.79	8.90	9.29	9.37	**9.76**	9.68
	improv.(%)	11.03	9.66	5.06	4.16	-	0.83
	p-value	0.000	0.000	0.000	0.000	-	0.009
R^{rev}	value	1.07	1.13	1.23	1.34	1.49	**1.56**
	improv.(%)	45.81	38.05	26.83	16.42	4.70	-
	p-value	0.000	0.000	0.000	0.000	0.001	-

Outline

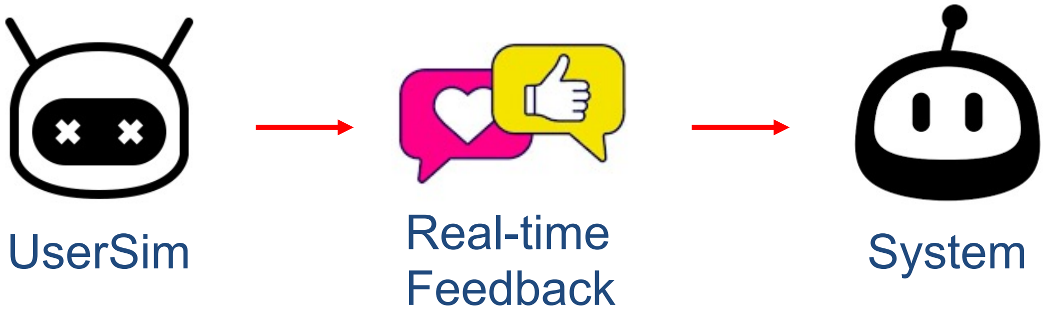
- Recommendations in Single Scenario
  - DeepPage - Deep Reinforcement Learning for Page-wise Recommendations (RecSys'2018)
  - DEERS - Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning (KDD'2018)
  - DRN - A Deep Reinforcement Learning Framework for News Recommendation (WWW'2018)
- Recommendations in Multiple Scenarios
  - DeepChain - Whole-Chain Recommendations (CIKM'2020)
  - MA-RDPG - Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning (WWW'2018)
  - RAM - Jointly Learning to Recommend and Advertise (KDD'2020)
  - DEAR - Deep Reinforcement Learning for Online Advertising in Recommender Systems (AAAI'2021)
- Online Environment Simulator
  - UserSim - User Simulation via Supervised Generative Adversarial Network (WWW'2021)
- Surveys
  - Deep Reinforcement Learning for Search, Recommendation, and Online Advertising: A Survey (SIGWEB'2019)
  - Reinforcement Learning based Recommender Systems: A Survey (Arxiv'2021)

Real-time Feedback

- The most practical and precise way is online A/B test

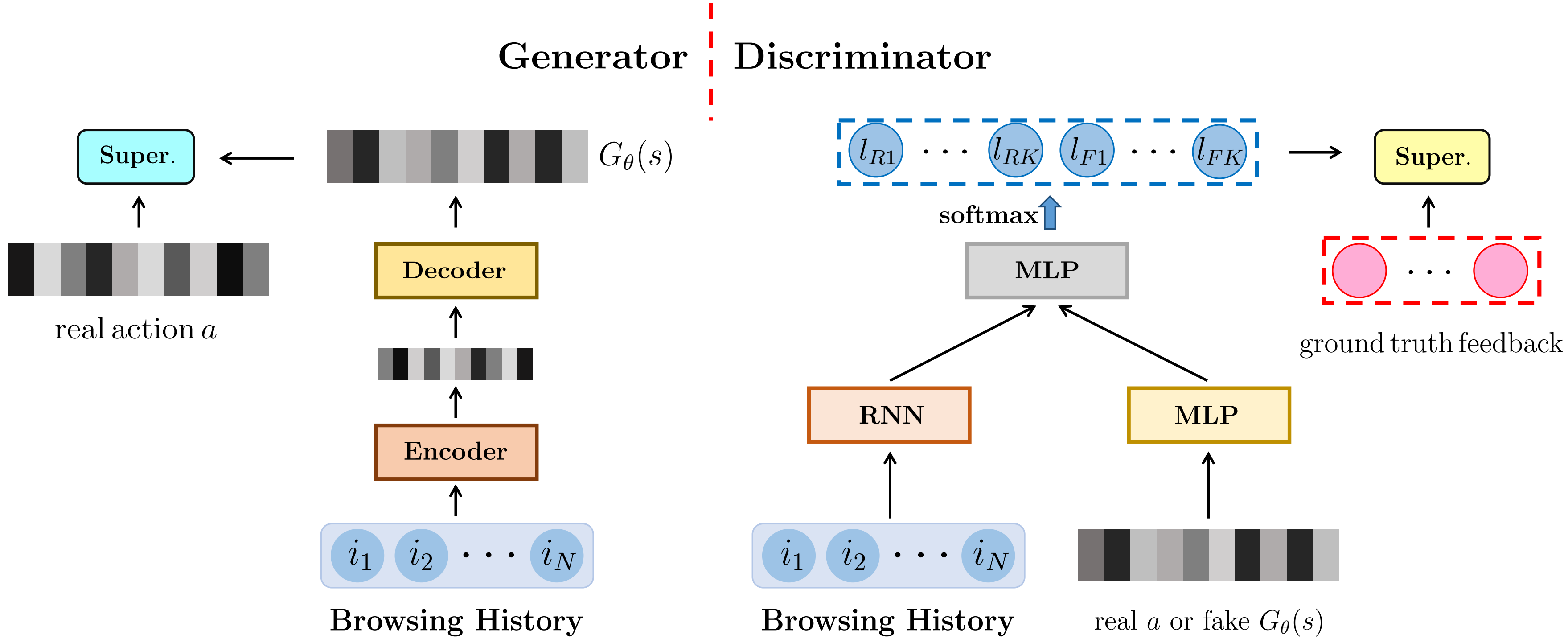


- Online A/B test is inefficient and expensive
  - Taking several weeks to collect sufficient data
  - Numerous engineering efforts
  - Bad user experience



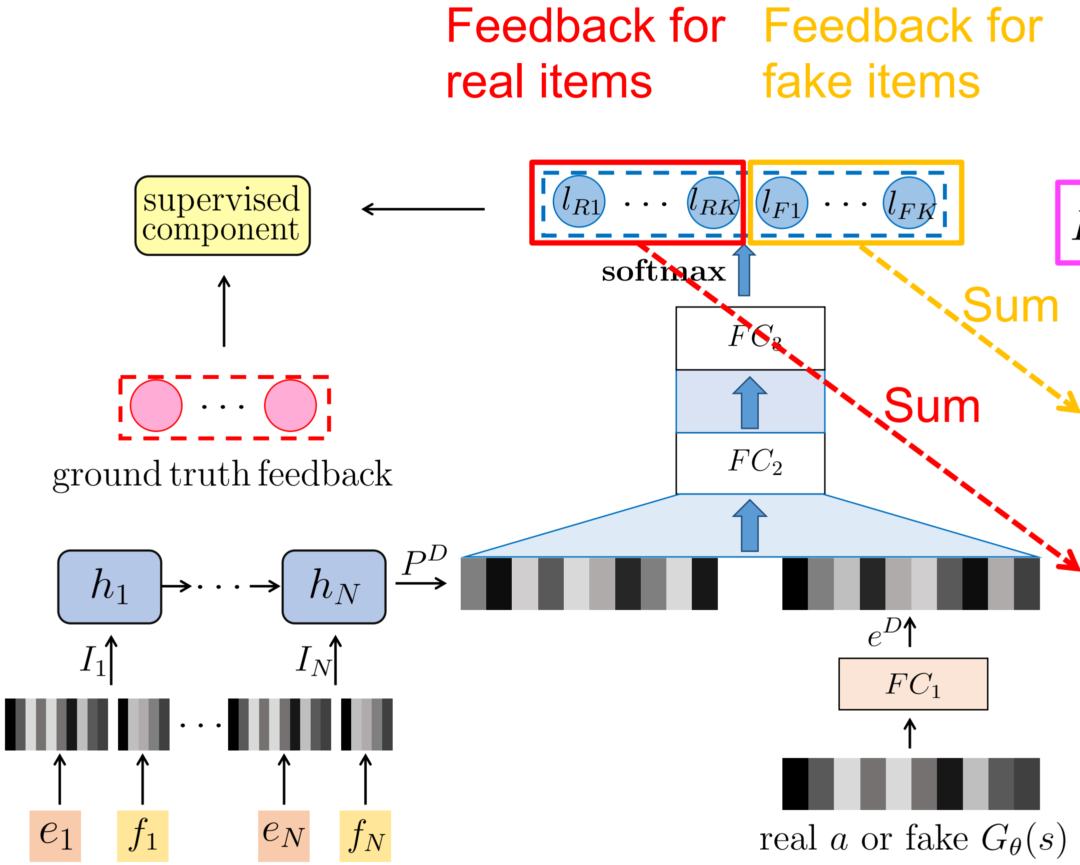
Overview

- Simulating users' real-time feedback is challenging
  - Underlying distribution of item sequences is extremely complex
  - Data available to each user is rather limited



Optimization

- Discriminator



$$\boxed{L_D^{unsup}} = -\{\mathbb{E}_{s,a\sim p_{data}} \log D_\phi(s,a) + \mathbb{E}_{s\sim p_{data}} \log D_\phi(s, G_\theta(s))\}$$

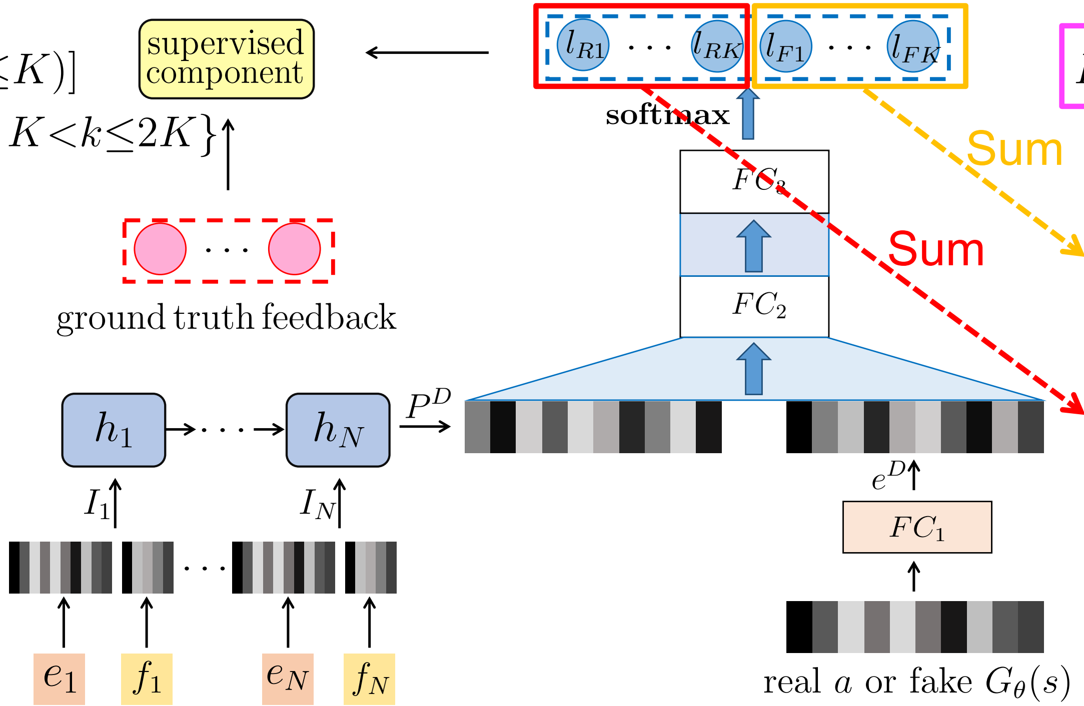
$$D_\phi(s, G_\theta(s)) = \sum_{k=K+1}^{2\times K} p_{model}(l_k | s, G_\theta(s))$$

$$D_\phi(s,a) = \sum_{k=1}^{K} p_{model}(l_k | s, a)$$

Optimization

- Discriminator

$$\boxed{L_D^{sup}} = -\{\mathbb{E}_{s,a,r\sim p_{data}}[\log p_{model}(l_k|s,a,k\le K)] + \lambda\cdot\mathbb{E}_{s,r\sim p_{data}}[\log p_{model}(l_k|s,G_\theta(s)),K<k\le 2K\}$$



$$\boxed{L_D^{unsup}} = -\{\mathbb{E}_{s,a\sim p_{data}}\log D_\phi(s,a) + \mathbb{E}_{s\sim p_{data}}\log D_\phi(s,G_\theta(s))\}$$

$$D_\phi(s,G_\theta(s)) = \sum_{k=K+1}^{2\times K} p_{model}(l_k|s,G_\theta(s))$$

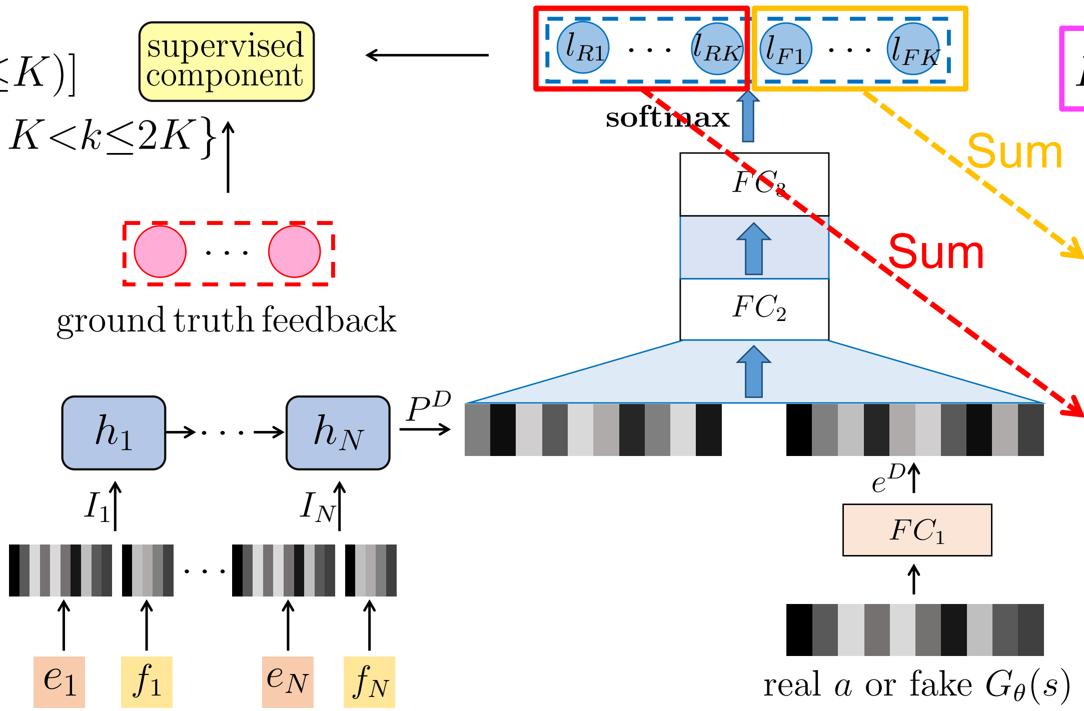
$$D_\phi(s,a) = \sum_{k=1}^{K} p_{model}(l_k|s,a)$$

Optimization

- Discriminator

$$L_D = \boxed{L_D^{unsup}} + \alpha \cdot \boxed{L_D^{sup}}$$

$$\boxed{L_D^{sup}} = -\{\mathbb{E}_{s,a,r\sim p_{data}}[\log p_{model}(l_k|s,a,k\leq K)] + \lambda \cdot \mathbb{E}_{s,r\sim p_{data}}[\log p_{model}(l_k|s,G_\theta(s)), K<k\leq 2K\}$$



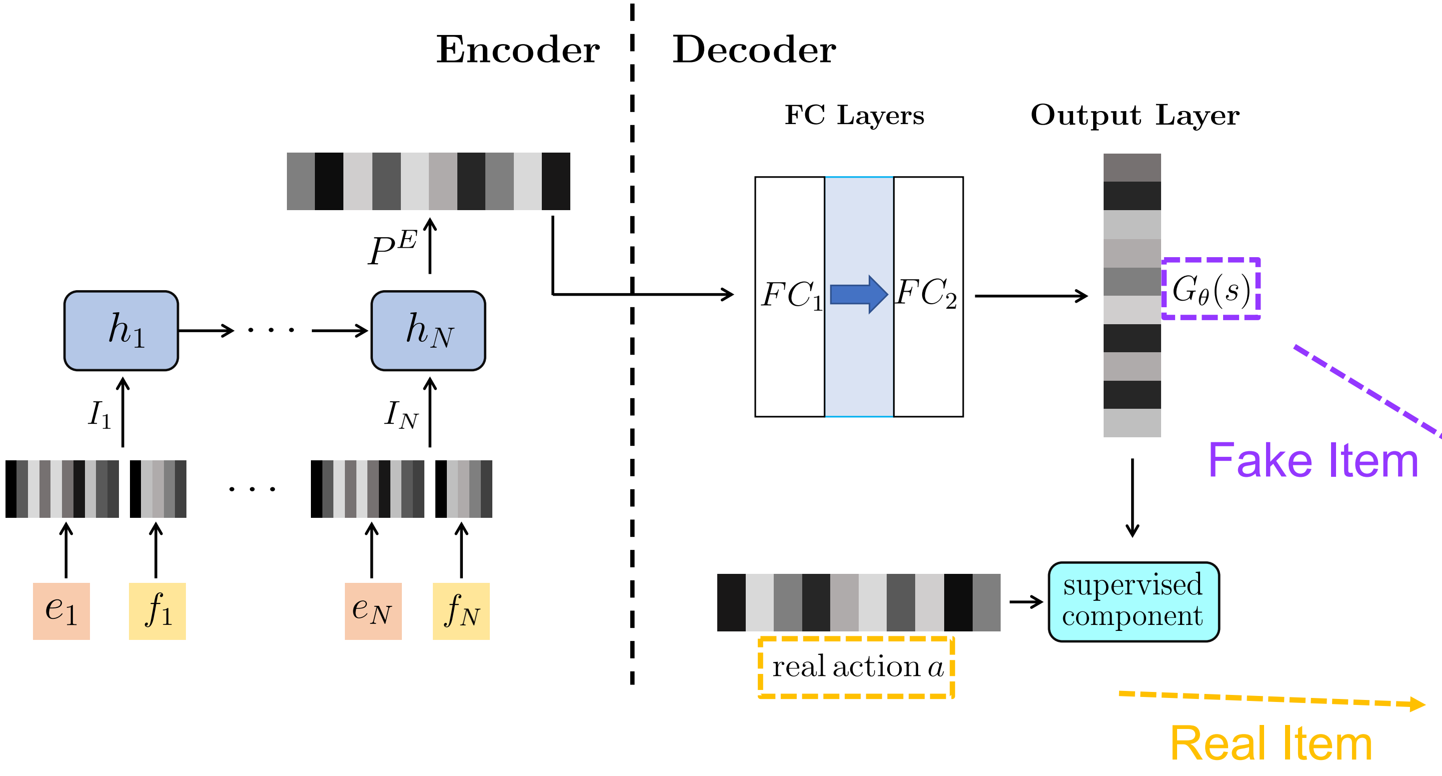
$$\boxed{L_D^{unsup}} = -\{\mathbb{E}_{s,a\sim p_{data}} \log D_\phi(s,a) + \mathbb{E}_{s\sim p_{data}} \log D_\phi(s,G_\theta(s))\}$$

$$D_\phi(s,G_\theta(s)) = \sum_{k=K+1}^{2\times K} p_{model}(l_k|s,G_\theta(s))$$

$$D_\phi(s,a) = \sum_{k=1}^{K} p_{model}(l_k|s,a)$$

Optimization

- Generator

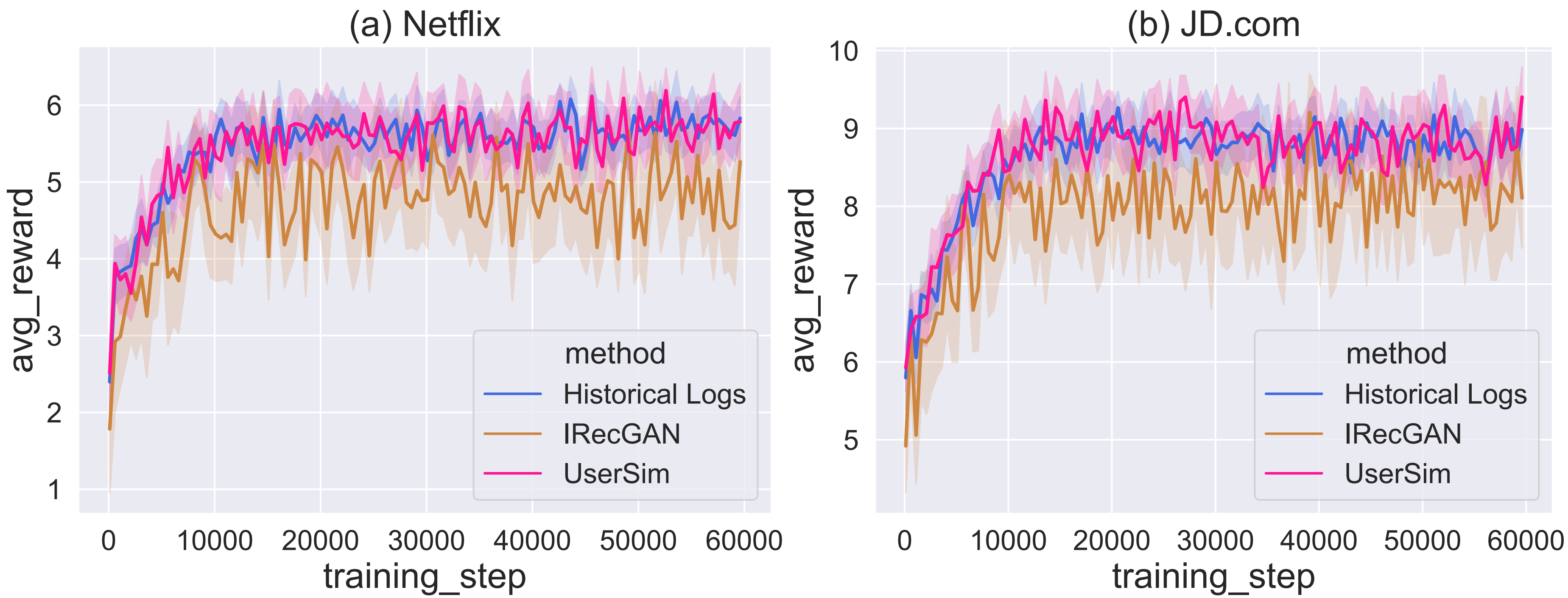


$$\boxed{L_G^{unsup}} = \mathbb{E}_{s\sim p_{data}}[\log D_\phi(s, G_\theta(s))]$$

$$L_G = \boxed{L_G^{unsup}} + \beta \cdot \boxed{L_G^{sup}}$$

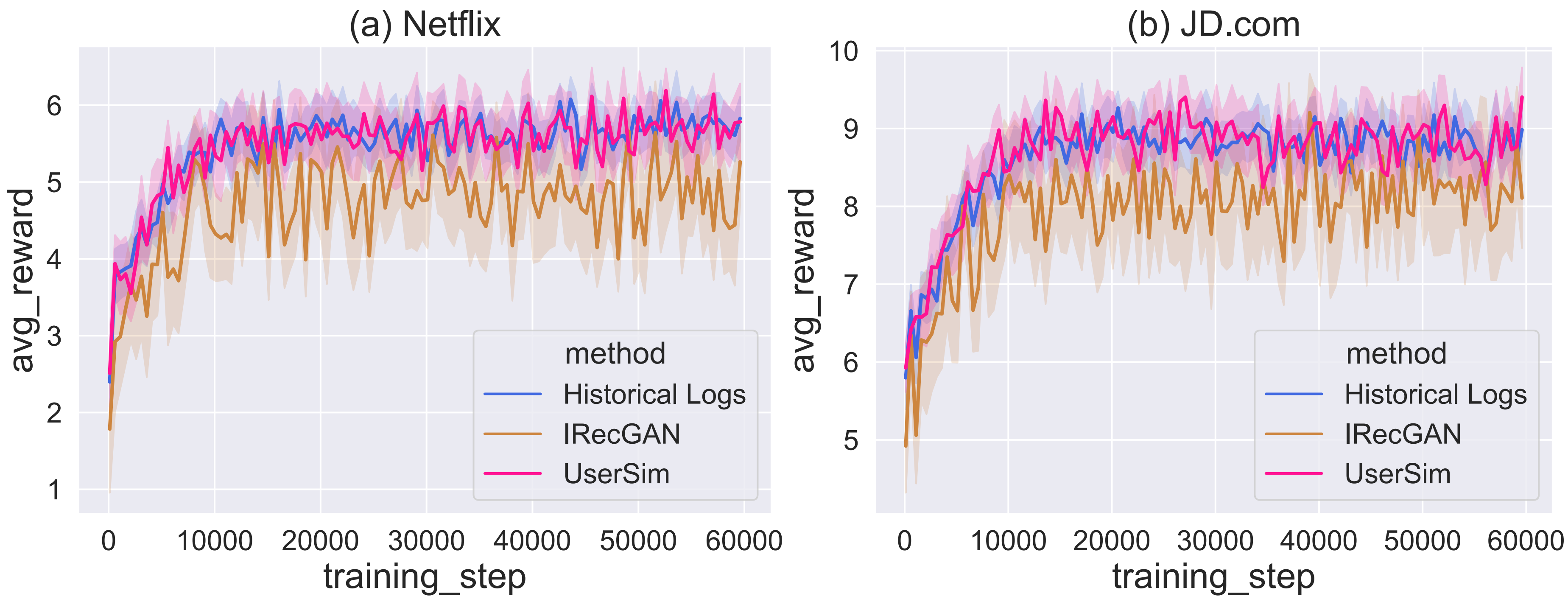
$$\boxed{L_G^{sup}} = \mathbb{E}_{s,a\sim p_{data}}\|\boxed{a} - \boxed{G_\theta(s)}\|_2^2$$

RL-based Recommender Training



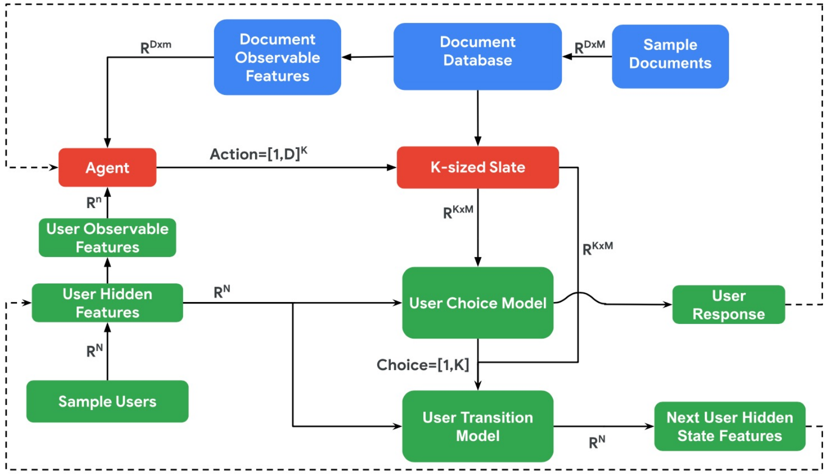
- Metric: average reward of a session
- Baselines: Historical Logs, IRecGAN

RL-based Recommender Training

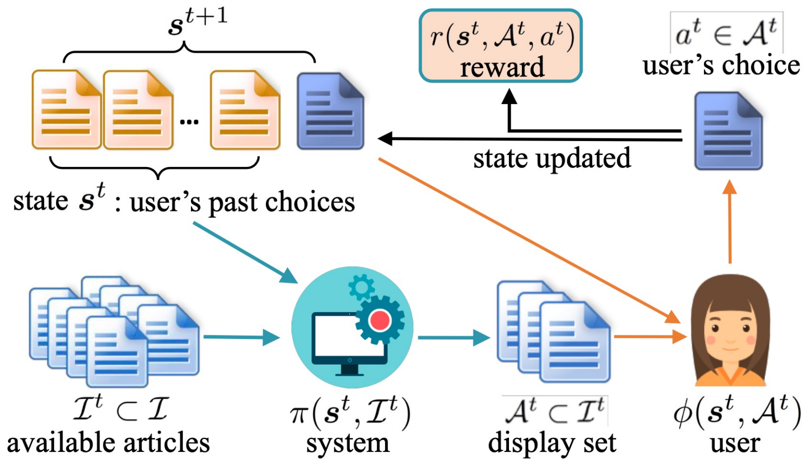


- Metric: average reward of a session
- Baselines: Historical Logs, IRecGAN
- UserSim converges to the similar avg_reward with the one upon historical data
- UserSim performs much more stably than the one trained based upon IRecGAN

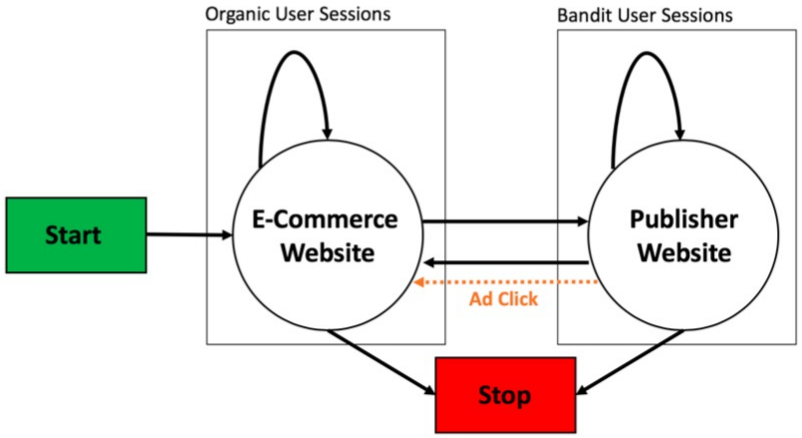
Other Simulators



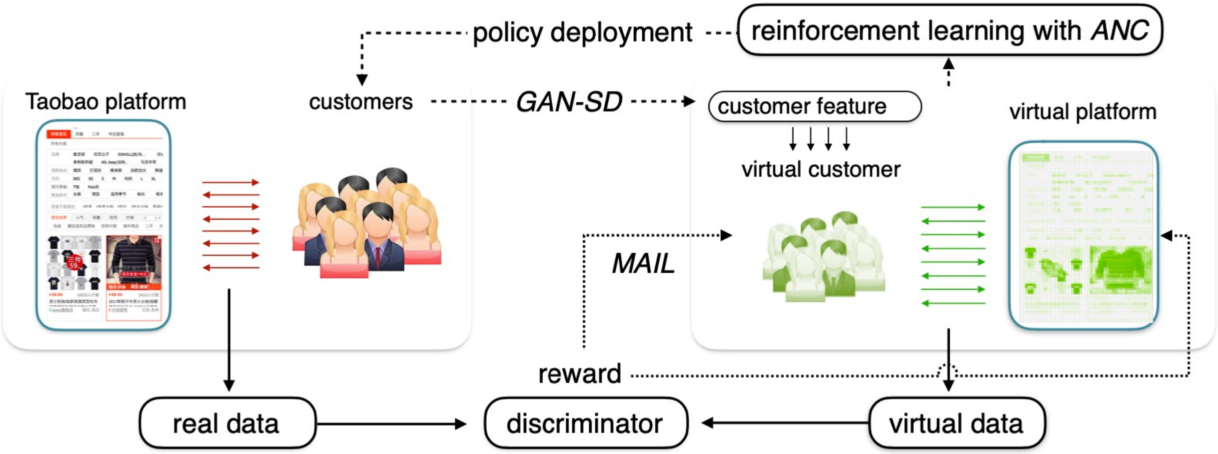
RecSim @ Google



GAN-PW @ Alibaba



RecoGym @ Criteo



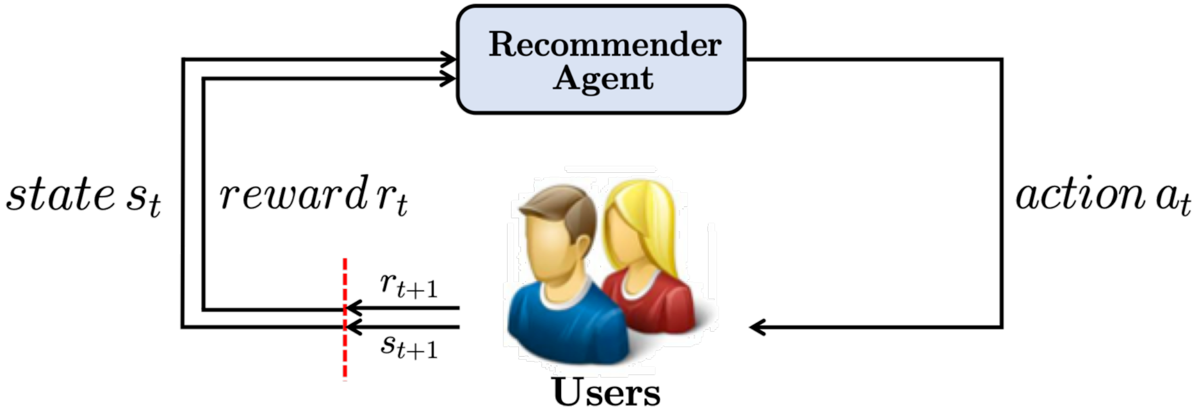
Virtual-Taobao @ Alibaba

Outline

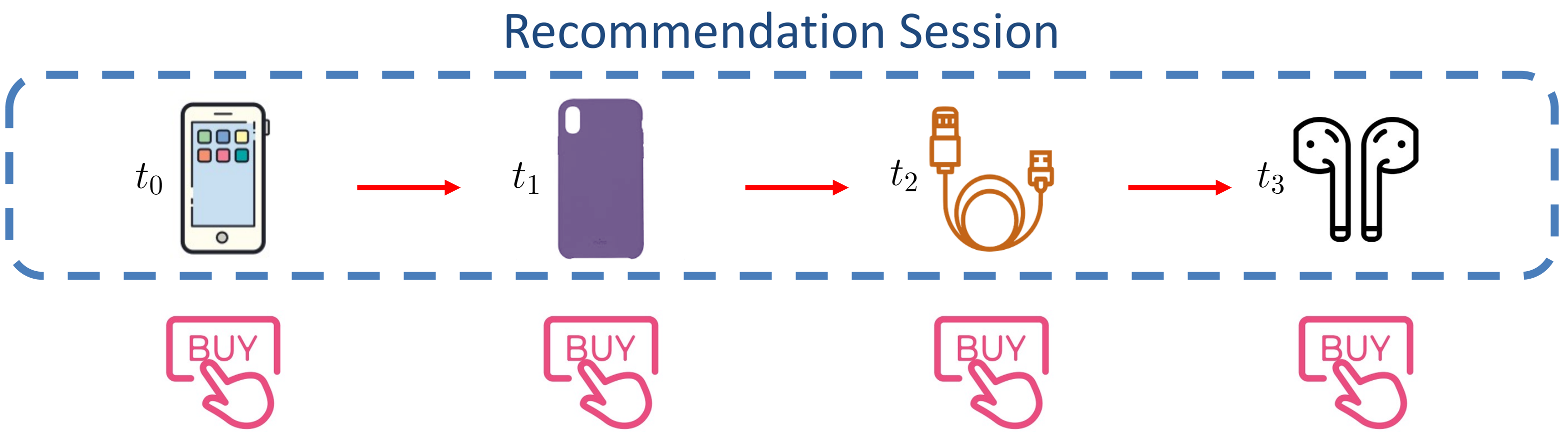
- Recommendations in Single Scenario
  - DeepPage - Deep Reinforcement Learning for Page-wise Recommendations (RecSys’2018)
  - DEERS - Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning (KDD’2018)
  - DRN - A Deep Reinforcement Learning Framework for News Recommendation (WWW’2018)
- Recommendations in Multiple Scenarios
  - DeepChain - Whole-Chain Recommendations (CIKM’2020)
  - MA-RDPG - Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning (WWW’2018)
  - RAM - Jointly Learning to Recommend and Advertise (KDD’2020)
  - DEAR - Deep Reinforcement Learning for Online Advertising in Recommender Systems (AAAI’2021)
- Online Environment Simulator
  - UserSim - User Simulation via Supervised Generative Adversarial Network (WWW’2021)
- Surveys
  - Deep Reinforcement Learning for Search, Recommendation, and Online Advertising: A Survey (SIGWEB’2019)
  - Reinforcement Learning based Recommender Systems: A Survey (Arxiv’2021)

Conclusion

- Continuously updating the recommendation strategies during the interactions



- Maximizing the long-term reward from users

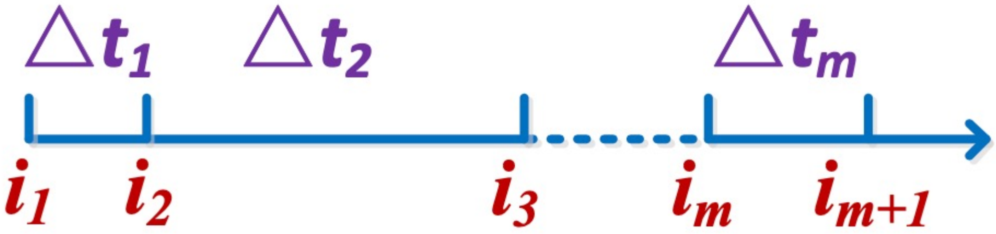


Future Directions

- Incorporating more types of user-item interactions into recommendations

Shopping Cart | Repeat Purchase | Favorites | Dialog System | Dwelling Time

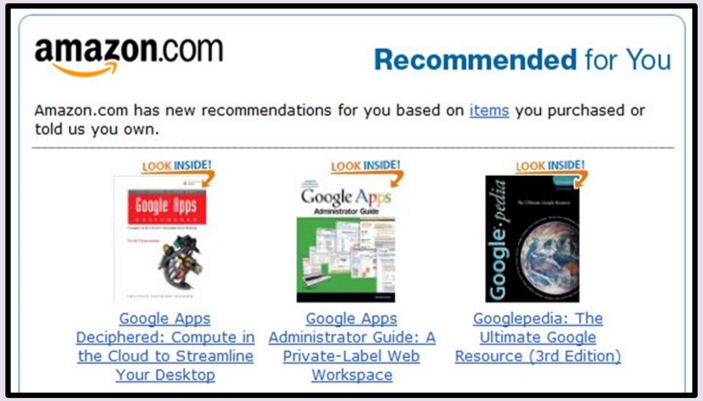
- Considering continuous time information for recommendations



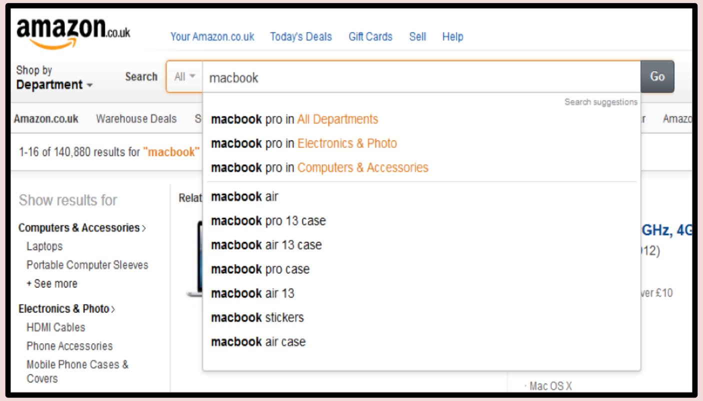
Reinforcement Learning for Search Engine

- Goal: finding and ranking a set of items based on a user query

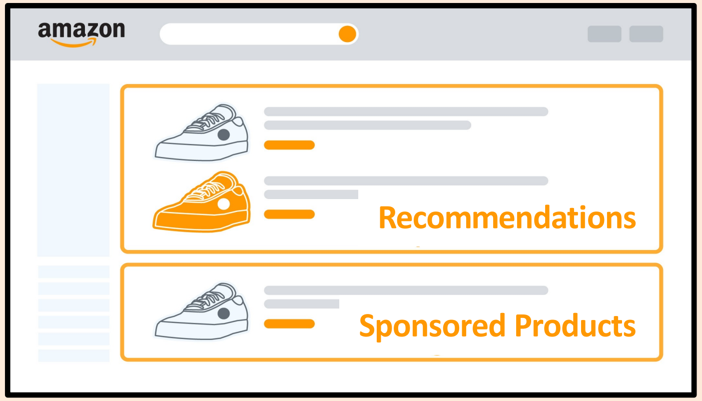
Recommendations



Search Engine



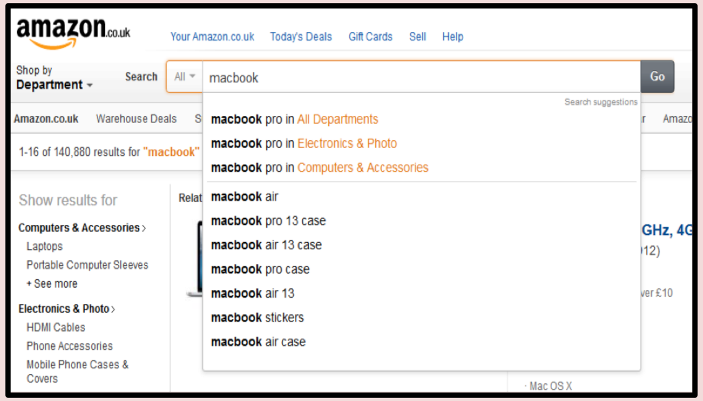
Advertisements



Reinforcement Learning for Search Engine

- Goal: finding and ranking a set of items based on a user query
  - **Query understanding:** jointly learning the tokenization, spelling correction, query rewriting and entity recognition, etc

Search Engine

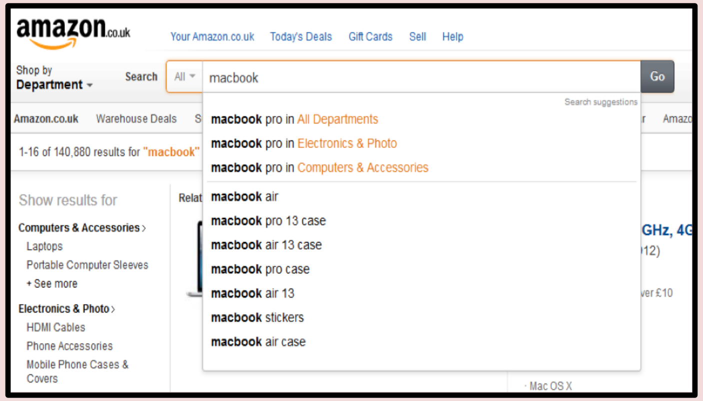


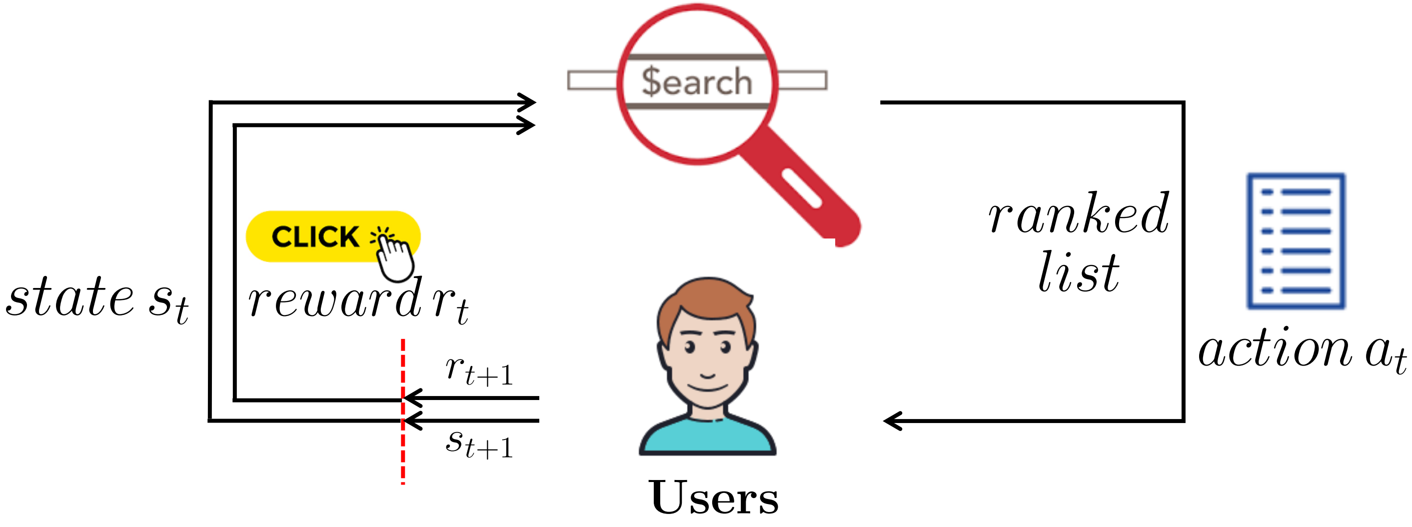
raw query → tokenization → spelling correction → query rewriting → entity recognition → structured query

Reinforcement Learning for Search Engine

- Goal: finding and ranking a set of items based on a user query
  - **Query understanding:** jointly learning the tokenization, spelling correction, query rewriting and entity recognition, etc
  - **Ranking:** directly optimizing user's feedback, such as user clicks & dwelling time

Search Engine





Reinforcement Learning for Search Engine

- Goal: finding and ranking a set of items based on a user query
  - **Query understanding:** jointly learning the tokenization, spelling correction, query rewriting and entity recognition, etc
  - **Ranking:** directly optimizing user's feedback, such as user clicks & stay time
  - **Session search:** user's behaviors of search results in the prior iteration will influence user's behaviors in the next search iteration