

ResAct: Reinforcing long-term engagement in sequential recommendation with residual actor

(Xue et al., 2022)



DEPARTAMENTO DE CIENCIA
DE LA COMPUTACIÓN

Javier Campos
Gabriel Campos
Alexandra Arriagada

Contexto

Un sistema de recomendación secuencial es aquel que mantiene un flujo continuo de recomendaciones hasta que el usuario termina la sesión o el servicio.



TikTok

Contexto

Long Term Engagement (LTE):

¿Cómo obtenerlo?

1. Con un bajo tiempo de retorno entre sesiones.
2. Un largo tiempo de duración de la sesión.

*“Long-term engagement is preferred over immediate engagement in sequential recommendation as it directly affects **product operational metrics** such as daily active users (DAUs) and dwell time.”*

Problema



Mejorar el engagement a largo plazo de una recomendación secuencial, es un **área menos investigada** comparada con el engagement inmediato



Relacionar los cambios en el engagement a largo plazo del usuario con una sola recomendación es un **problema difícil**

Trabajos relacionados

Recomendación secuencial

- 1 Adaptive, Personalized Diversity for Visual Discovery (Teo et al., 2016)
- 2 Reinforcement learning to optimize long-term user engagement in recommender systems (Zou et al., 2019)

RL en sistemas recomendadores

- 1 Deep reinforcement learning in large discrete action spaces (Dulac-Arnold et al., 2015)
- 2 Topk off-policy correction for a reinforce recommender system (Chen et al., 2019)

Trabajos relacionados: Limitantes

Recomendación secuencial

Se basan en **grandes supuestos empíricos** como que la diversidad de recomendaciones provocará una mejora en la retención de usuarios.



Según Zhao et al. (2020) no existe consenso científico en cuanto a cómo medir correctamente la diversidad

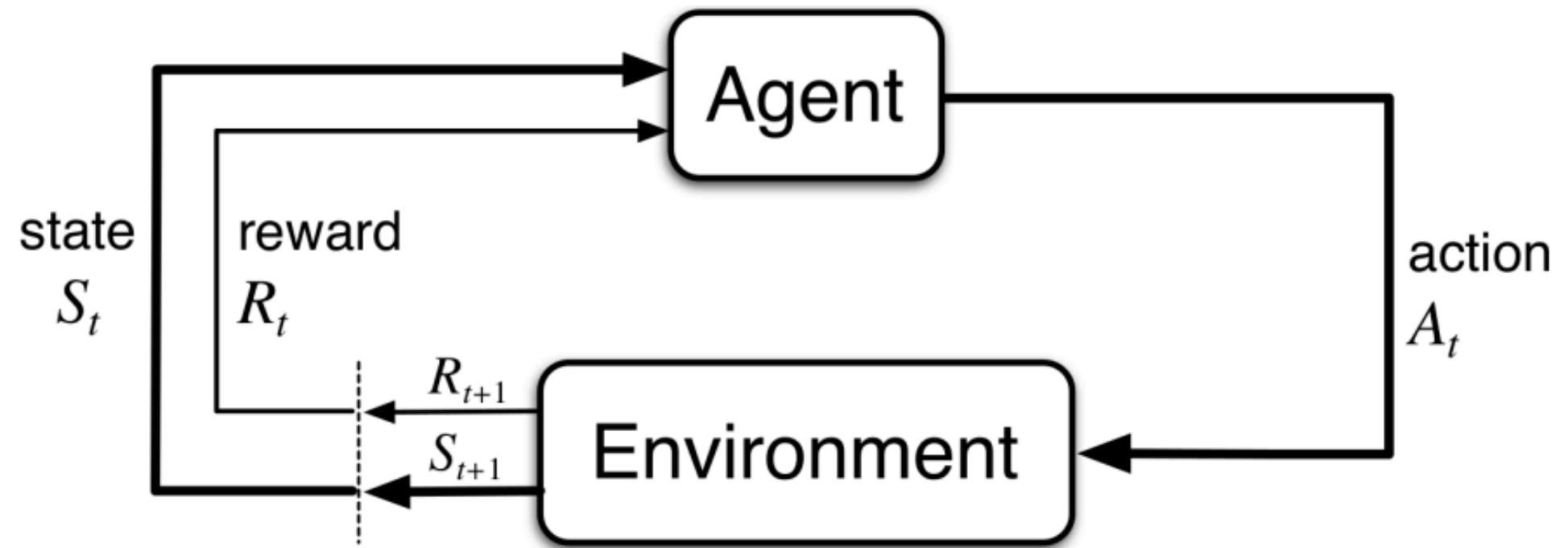
RL en sistemas recomendadores

Los agentes de RL deben aprender **en todo el espacio**, lo cual es complicado debido a que las interacciones en línea son **costosas**, además del **gran tamaño** de estados y acciones.

Solución

Reinforcement Learning (RL)

Se centra en aprender políticas que maximicen la recompensa acumulada desde una perspectiva a largo plazo (Sutton & Barto, 2018). Nos ofrece un framework prometedor para optimizar long-term engagement.



Fuente: curso Aprendizaje Reforzado, R. Toro 2024

Solución: Desafíos

Reinforcement Learning (RL)

La evolución del “user-stickiness” toma un periodo largo, dificultando la evaluación del Q-value.

La exploración requiere de experimentos en vivo, lo que puede afectar la experiencia del usuario.

Las recompensas asociadas a LTE son sparse.

Las representaciones estados podrían tener poca relación con LTE.

Propuesta

Residual Actor (ResAct)

Aprender una política cercana, pero mejor que la política on-line.

Se descompone el problema en dos sub-tareas:

- Reconstruir la política de recomendación on-line
- Mejorarla con las predicciones de un Residual Actor

Objetivos

1

Recopilar suficientes datos cerca de la política aprendida así poder estimar adecuadamente los valores de estado-acción.

2

Evitar la interacción en línea, reduciendo costos y riesgos asociados.

RL Set-up

MDP

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$$

Q-value

$$Q^\pi(s_t, a_t) = \mathbb{E}_{(s_{t'}, a_{t'}) \sim \pi} \left[r(s_t, a_t) + \sum_{t'=t+1}^{\infty} \gamma^{(t'-t)} \cdot r(s_{t'}, a_{t'}) \right]$$

Objetivo

$$\max_{\pi} \mathcal{J}(\pi) = \mathbb{E}_{s_t \sim d_t^\pi(\cdot), a_t \sim \pi(\cdot | s_t)} [Q^\pi(s_t, a_t)]$$

Setup

$\max_{\pi} \mathcal{J}(\pi)$ es difícil

Separamos el problema en 2:

$$\hat{a} = a_{on} + \Delta(s, a_{on})$$

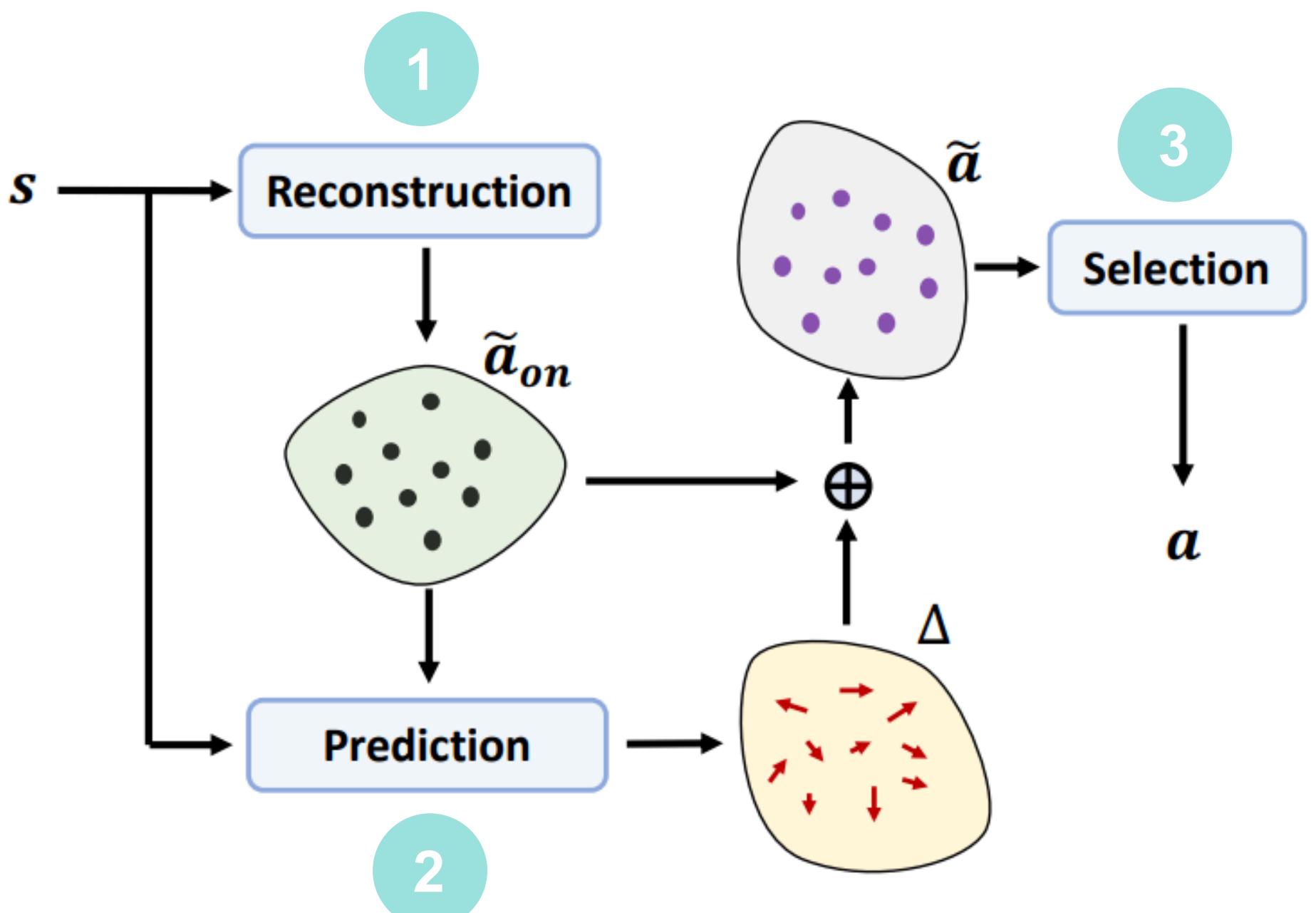
$\pi_{on}(a|s)$ Política online

$\hat{\pi}(a|s)$ Política objetivo



Es de esperar que:

$$\mathcal{J}(\hat{\pi}) \geq \mathcal{J}(\pi_{on})$$

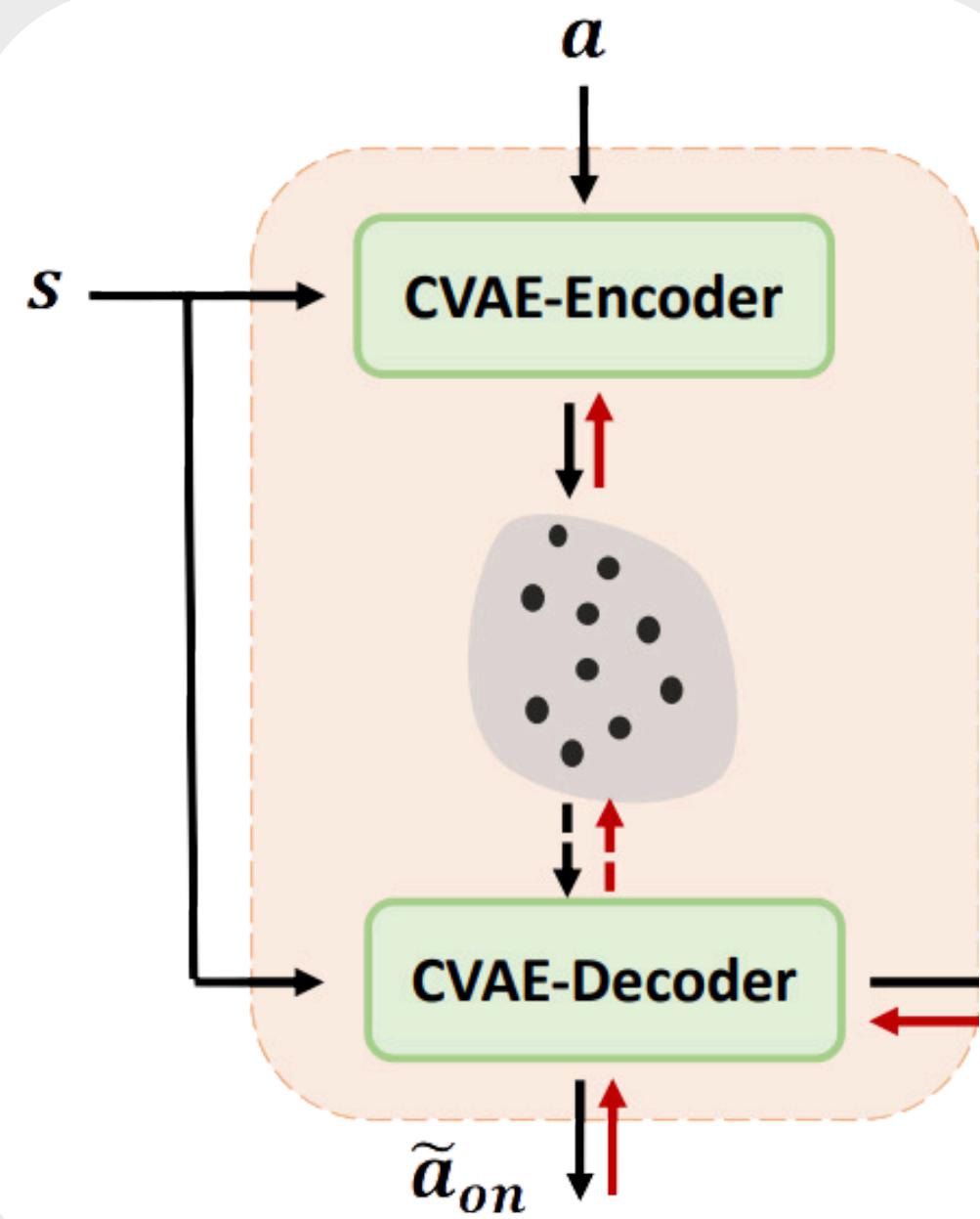


- 1 Reconstruction
- 2 Prediction
- 3 Selection

Algoritmo

1

Reconstruction



CVAE: Conditional Variational Auto Encoder

$$E(\cdot|s, a_{on}; \theta_e)$$

$$c \sim \mathcal{C}(s, a_{on})$$

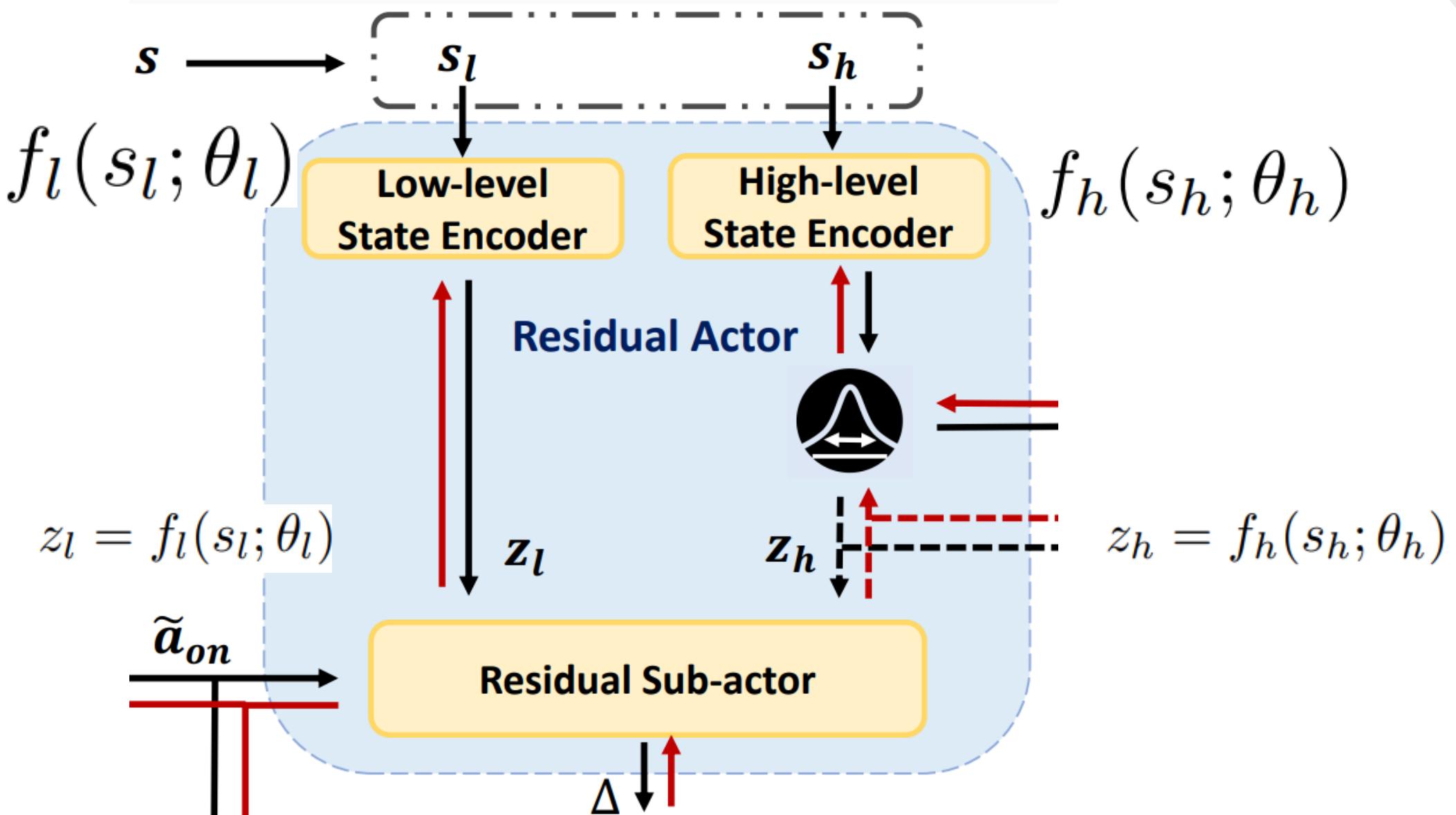
$$D(a|s, c; \theta_d)$$

Reconstruction Loss: MSE + KL regularizer

$$L_{\theta_e, \theta_d}^{Rec} = \mathbb{E}_{s, a_{on}, c} [(D(a|s, c; \theta_d) - a_{on})^2 + KL(\mathcal{C}(s, a_{on}; \theta_e) || \mathcal{N}(0, 1))]$$

2

Prediction



ResAct: Feature
Extraction+ sub-actor

$$\begin{aligned}
 f(\Delta | s, a; \theta_f) \\
 = \\
 \{f_h, f_l, f_a\}
 \end{aligned}$$

$$\Delta = f_a(z, a; \theta_a) ; z = \text{Concat}(z_h, z_l)$$

Optimizando la política

$$\hat{\pi}(a|s, c) = D(\tilde{a}_{on}|s, c; \theta_d) + f(\Delta|s, \tilde{a}_{on}; \theta_f)$$

Se busca optimizar $\{\theta_d, \theta_f\}$ de modo que $\mathcal{J}(\hat{\pi})$ sea máximo

Esto requiere calcular state-action values, es decir,
necesitamos un **crítico**.

$$Q^{\hat{\pi}}(s, a)$$

Crítico

Se aproxima mediante el método Clipped Double Q-Learning:

$$Q_1(s, a; \theta_{q_1}) \text{ y } Q_2(s, a; \theta_{q_2})$$

Se optimizan los parámetros usando Temporal Difference:

$$L_{\theta_{q_j}}^{TD} = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1})} [(Q_j(s_t, a_t; \theta_{q_j}) - y)^2], j = \{1, 2\};$$

$$y = r_t + \gamma \min \left[Q'_1(s_{t+1}, \hat{\pi}'(a_{t+1}|s_{t+1}); \theta'_{q_1}), Q'_2(s_{t+1}, \hat{\pi}'(a_{t+1}|s_{t+1}); \theta'_{q_2}) \right]$$

DPG

Para optimizar la política, se calculan los gradientes:

$$\nabla_{\theta_f} \mathcal{J}(\hat{\pi}) = \mathbb{E}_{s,c} \left[\nabla_a Q^{\hat{\pi}}(s, a) \Big|_{a=\hat{\pi}(a|s,c)} \nabla_{\theta_f} f(\Delta|s, a; \theta_f) \Big|_{a=D(a|s,c;\theta_d)} \right]$$

$$\nabla_{\theta_d} \mathcal{J}(\hat{\pi}) = \mathbb{E}_{s,c} \left[\nabla_a Q^{\hat{\pi}}(s, a) \Big|_{a=\hat{\pi}(a|s,c)} \nabla_{\theta_d} D(a|s, c; \theta_d) \right].$$

Se actualiza el decoder y el actor según:

$$\theta_f \leftarrow \theta_f + \nabla_{\theta_f} \mathcal{J}(\hat{\pi})$$

$$\theta_d \leftarrow \theta_d + \nabla_{\theta_d} \mathcal{J}(\hat{\pi}) - \nabla_{\theta_d} L_{\theta_e, \theta_d}^{Rec}$$

Selection

Se puede usar al crítico entrenado para seleccionar y calcular finalmente la política óptima

$$\hat{\pi}(a|s) = \hat{\pi}(a|s, c^*);$$

$$c^* = \arg \max_c Q_1(s, \hat{\pi}(a|s, c); \theta_{q_1}), c \in \{c^i \sim \mathcal{N}(0, 1)\}_{i=0}^n$$

Mejorando las representaciones de estados

“Good state representations always ease the learning of models”
(Nielsen, 2015)

Queremos que z_h sea una buena representación del long term engagement, para eso es intuitivo:

1. Maximizar la información mutua con rewards long term **Expressiveness**
2. Minimizar la entropía de la representación **Conciseness**

Se propone un encoder estocástico:

$$(\mu_h, \sigma_h) = f_h(s_h; \theta_h)$$
$$z_h \sim \mathcal{N}(\mu_h, \sigma_h)$$

Expressiveness

Maximizar cota inferior de información mutua entre la representación y la recompensa a largo plazo:

$$I_{\theta_h}(z_h; r) \geq \iint p_{\theta_h}(z_h)p(r|z_h) \log o(r|z_h; \theta_o) dz_h dr + H(r)$$

Minimiza la función de pérdida:

$$L_{\theta_h, \theta_o}^{Exp} = \mathbb{E}_{s, z_h \sim p_{\theta_h}(z_h|s_h)} [\mathcal{H}(p(r|s)||o(r|z_h; \theta_o))]$$

Conciseness

Minimizar cota superior de información mutua entre la representación y el estado de alto nivel:

$$I_{\theta_h}(z_h; s_h) \leq \iint p(s_h) p_{\theta_h}(z_h|s_h) \log \frac{p_{\theta_h}(z_h|s_h)}{m(z_h)} ds_h dz_h$$

Minimiza la función de pérdida:

$$L_{\theta_h}^{Con} = \mathbb{E}_s [KL(p_{\theta_h}(z_h|s_h) || m(z_h))]$$

Datasets

MovieLens-1m

RecL-25m

Métrica de Evaluación

Normalized Capped Importance Sampling

$$\tilde{J}^{NCIS}(\pi) = \frac{1}{|\mathcal{T}|} \sum_{\xi \in \mathcal{T}} \left[\frac{\sum_{(s,a,r) \in \xi} \tilde{\rho}_{\pi,\pi_\beta}(s,a)r}{\sum_{(s,a,r) \in \xi} \tilde{\rho}_{\pi,\pi_\beta}(s,a)} \right], \quad \tilde{\rho}_{\pi,\pi_\beta}(s,a) = \min \left(c, \frac{\phi_{\pi(s)}(a)}{\phi_{\pi_\beta(s)}(a)} \right)$$

Resultados

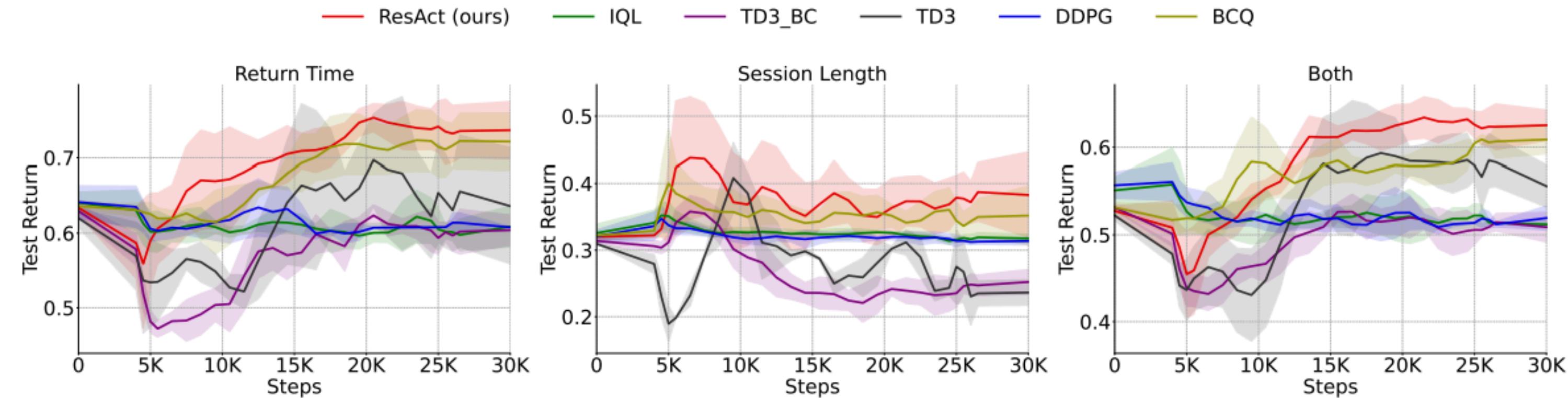


Figure 5: Learning curves of RL-based methods on *RecL-25m*, averaged over 5 runs.

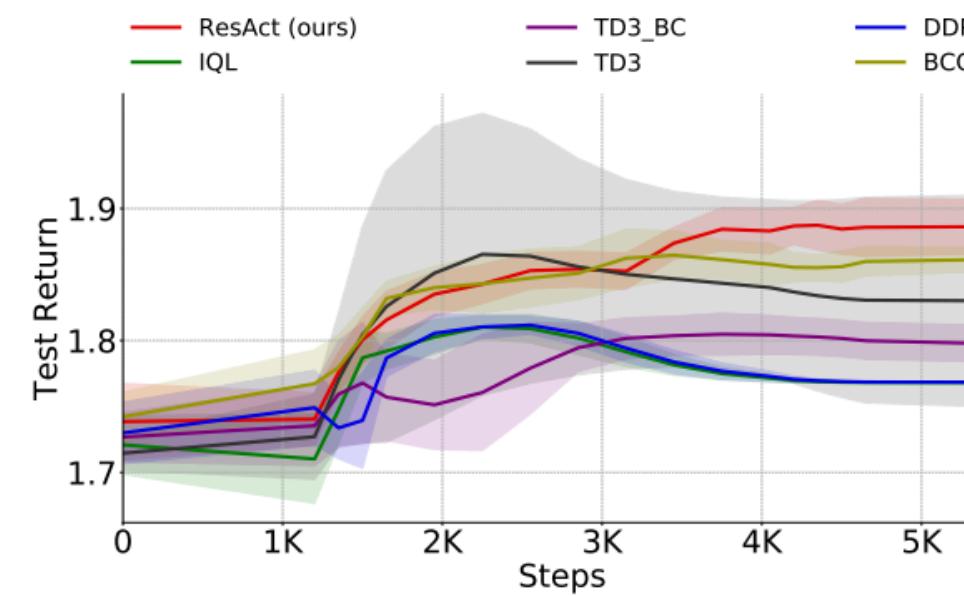


Figure 4: Learning curves of RL-based methods on *MovieLensL-1m*.

Resultados

Table 2: Performance comparison on MovieLensL-1m. The “ \pm ” indicates 95% confidence intervals.

	Return
DDPG	1.7429 \pm 0.0545
TD3	1.7363 \pm 0.0546
TD3_BC	1.7135 \pm 0.0541
BCQ	1.7898 \pm 0.0320
IQL	1.7360 \pm 0.0546
IL	1.7485 \pm 0.0310
IL_CVAE	1.7344 \pm 0.0316
ResAct (Ours)	1.8123 \pm 0.0319

Table 3: Performance comparison on RecL-25m in various tasks. The “ \pm ” indicates 95% confidence intervals.

	Return	Time	Session Length	Both
DDPG	0.6375 \pm 0.0059	0.3290 \pm 0.0056	0.5908 \pm 0.0092	
TD3	0.6756 \pm 0.0133	0.4015 \pm 0.0073	0.5498 \pm 0.0103	
TD3_BC	0.6436 \pm 0.0059	0.3671 \pm 0.0037	0.5563 \pm 0.0050	
BCQ	0.6837 \pm 0.0061	0.3836 \pm 0.0033	0.5915 \pm 0.0049	
IQL	0.6296 \pm 0.0094	0.3430 \pm 0.0057	0.5579 \pm 0.0067	
IL	0.6404 \pm 0.0058	0.3186 \pm 0.0032	0.5345 \pm 0.0048	
IL_CVAE	0.6410 \pm 0.0058	0.3178 \pm 0.0031	0.5346 \pm 0.0047	
ResAct (Ours)	0.7980 \pm 0.0067	0.5433 \pm 0.0045	0.6675 \pm 0.0053	



Conclusiones

- ResAct se diseña para mejorar el LTE en recomendación secuencial, **aumentando el tiempo y la frecuencia de las sesiones.**
- ResAct aprende una política cercana, pero mejor que la política en uso.
- ResAct funciona reconstruyendo la política en uso y luego mejorandola al agregar una acción residual.

- Se utilizan regularizadores de teoría de la información para hacer que las representaciones de estado sean más expresivas y concisas.
- Los resultados experimentales **muestran la superioridad de ResAct frente a métodos previos** tanto en el dataset de benchmarking MovieLensL-1m y el dataset real RecL-25m

Comentarios

Inevitablemente causará problemas de adicción

No se compara experimentalmente con
GRU4Rec

Baselines no tan relacionados al problema

No se compara con el trabajo de Chen (2019)

No existe un benchmark adecuado para este
problema

Referencias

Choon Hui Teo, Houssam Nassif, Daniel Hill, Sriram Srinivasan, Mitchell Goodman, Vijai Mohan, and SVN Vishwanathan. Adaptive, personalized diversity for visual discovery. In Proceedings of the 10th ACM Conference on Recommender Systems, pp. 35–38, 2016.

Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, and Dawei Yin. Reinforcement learning to optimize long-term user engagement in recommender systems. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2810–2818, 2019.

Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. Deep reinforcement learning in large discrete action spaces. arXiv preprint arXiv:1512.07679, 2015.

Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. Topk off-policy correction for a reinforce recommender system. In Proceedings of the 12th ACM International Conference on Web Search and Data Mining, pp. 456–464, 2019.

Yifei Zhao, Yu-Hang Zhou, Mingdong Ou, Huan Xu, and Nan Li. Maximizing cumulative user engagement in sequential recommendation: An online optimization perspective. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2784–2792, 2020c.

Richard S Sutton and Andrew G Barto. Reinforcement Learning: An Introduction. MIT Press, 2018.

Wanqi Xue, Qingpeng Cai, Ruohan Zhan, Dong Zheng, Peng Jiang, Kun Gai, Bo An. ResAct: Reinforcing Long-Term Engagement in Sequential Recommendation with Residual Actor. ICLR, 2023.