ResAct: Reinforcing Long-term Engagement in Sequential Recommendation with Residual Actor

Wanqi Xue*, Qingpeng Cai, Ruohan Zhan, Dong Zheng, Peng Jiang, Kun Gai, Bo An

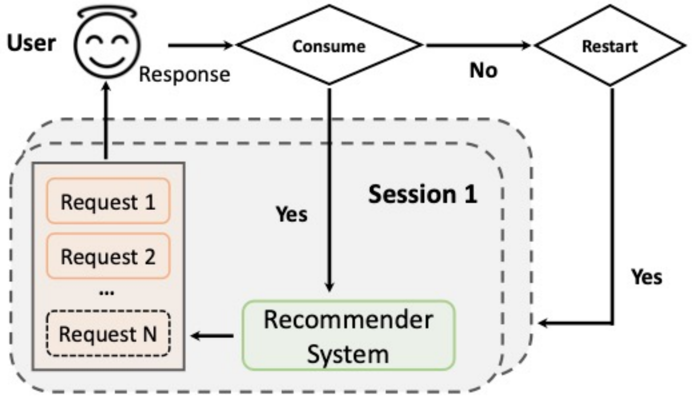
School of Computer Science and Engineering, NTU, Singapore

Kuaishou Technology

* This work was done during an internship at Kuaishou

Long-term engagement in Sequential Recommendation

- Sequential recommendation
  - Session-Request structure
  - Keep feeding items to users
  - Users decide when to start and quit a session



- Long-term user engagement
  - Increasing users' visiting frequency: low return time between sessions
  - Increasing the length of sessions: more items can be consumed in each session
- Reinforcement learning: a promising direction
  - Focusing on maximizing cumulative reward from a long-term perspective

KUAISHOU

Sequential Recommendation as an MDP

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

- Describe sequential recommendation as a Markov Decision Process
  - Defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$
  - $\mathcal{S} = \bar{\mathcal{S}}_h \times \mathcal{S}_l$ is the state space
    - Bi-level structure in sequential recommendation
    - Decompose the state space to session-level (high-level) and request-level (low-level) features
  - Rewards relevant to long-term user engagement
    - Time duration between two sessions $r(\delta^u) = \left(\lfloor \frac{\min(\delta^u_{avg}, \delta_{75\%})}{\delta^u} \rfloor\right).clip(0,5)$
    - Session Length $r(\eta^u) = \left(\lfloor \frac{\eta^u}{\eta^u_{avg} \times 0.8} \rfloor\right).clip(0,5)$
  - Optimization objective

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

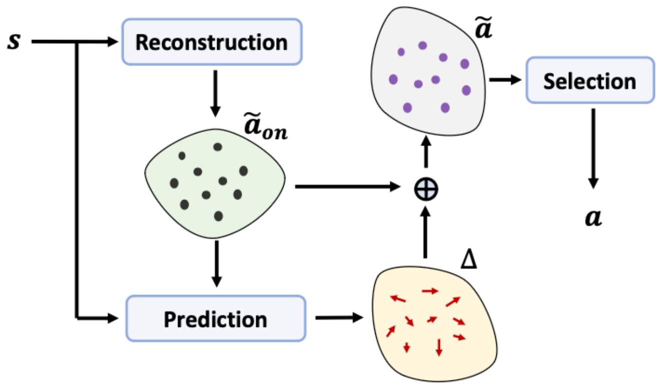
$$Q^\pi(s_t, a_t) = \mathbb{E}_{(s_{t'}, a_{t'}) \sim \pi} \left[r(s_t, a_t) + \sum_{t'=t+1}^{\infty} \gamma^{(t'-t)} \cdot r(s_{t'}, a_{t'}) \right]$$

Challenges

- Relating changes in long-term engagement to a single recommendation
- For reinforcement learning methods
  - Policy optimization in huge policy space
    - The evaluation of state-action values (Q values)
    - Probing previously unexplored areas will hurt users experiences
    - Sparse rewards
    - State representations will not contain much information about long-term engagement

Workflow of ResAct

- Motivation
  - Learn a recommendation policy which is **broadly consistent to, but better than**, the online-serving policy
    - Sufficient data near the learning policy so that the state-action values can be well estimated
    - Safe recommendation
  - Relate state representations to long-term reward signals
- Direct learning such policy is difficult: adding an action residual
- Workflow
  - Reconstruction $\{\tilde{a}^i_{on} \sim \tilde{\pi}_{on}(a|s)\}^n_{i=1}$
  - Prediction $\tilde{a}^i = \tilde{a}^i_{on} + \Delta(s, \tilde{a}^i_{on})$
  - Selection $\arg\max_{\tilde{a}} Q^{\hat{\pi}}(s, \tilde{a})$ for $\tilde{a} \in \{\tilde{a}^i\}^n_{i=0}$.



Reconstructing Online Behaviors

- Imitating the behaviors of online serving policy

$$\tilde{\pi}_{on}(a|s) \approx \pi_{on}(a|s)$$

- A naïve approach $\mathbb{E}_{s,a_{on}\sim\pi_{on}(a|s)}\left[(D(a|s;\theta_d)-a_{on})^2\right]$.
  - It can only generate one candidate
- Inspired by VAE
  - Encoder and Decoder

$$E(\cdot|s,a_{on};\theta_e) \qquad D(a|s,c;\theta_d)$$

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

  - It can generate multiple action candidates

$$\tilde{a}^i_{on} = D(a|s,c^i;\theta_d) \qquad \{c^i \sim \mathcal{N}(0,1)\}^n_{i=0}.$$

KUAISHOU

Learning to Predict the Action Residual

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

- Improving upon the reconstructed candidates with a residual model

$$f(\Delta|s,a;\theta_f) \qquad \theta_f=\{\theta_h,\theta_l,\theta_a\}$$

$$z_h=f_h(s_h;\theta_h), z_l=f_l(s_l;\theta_l); z=Concat(z_h,z_l); \Delta=f_a(z,a;\theta_a).$$

- Generate an improved action with a deterministic policy

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

- Optimization through policy gradient

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

$$L^{TD}_{\theta_{q_j}}=\mathbb{E}_{(s_t,a_t,r_t,s_{t+1})}\left[(Q_j(s_t,a_t;\theta_{q_j})-y)^2\right], j=\{1,2\};$$
$$y=r_t+\gamma\min\left[Q_1^{'}(s_{t+1},\hat{\pi}^{'}(a_{t+1}|s_{t+1});\theta_{q_1}^{'}),Q_2^{'}(s_{t+1},\hat{\pi}^{'}(a_{t+1}|s_{t+1});\theta_{q_2}^{'})\right].$$

Learning to Predict the Action Residual

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

$$\theta_f \leftarrow \theta_f + \nabla_{\theta_f}\mathcal{J}(\hat{\pi}), \theta_f = \{\theta_h, \theta_l, \theta_a\}.$$

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

- Action selection
  - Reuse the learned state-action function (the critic)

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

Facilitating Feature Extraction

- Relating state features to long-term rewards
  - Expressiveness: maximize the mutual information between state features and rewards

$$I_{\theta_h}(z_h; r) \;=\; \iint p_{\theta_h}(z_h) p(r|z_h) \log \frac{p(r|z_h)}{p(r)} \mathrm{d}z_h$$

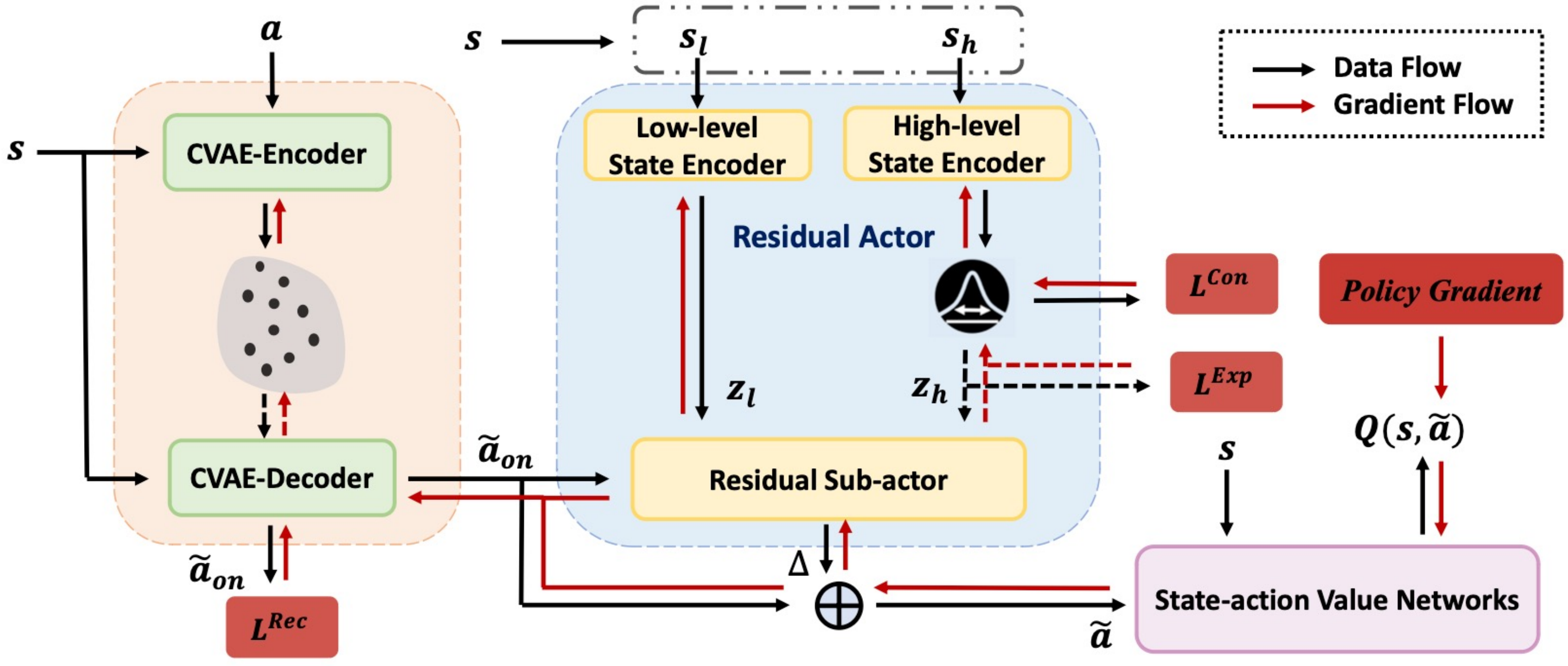
$$L^{Exp}_{\theta_h,\theta_o} = \mathbb{E}_{s,z_h \sim p_{\theta_h}(z_h|s_h)} \left[\mathcal{H}(p(r|s) || o(r|z_h;\theta_o))\right]$$

  - Conciseness: minimizing mutual information between state features and row states to reduce redundant information

$$\begin{aligned} L^{Con}_{\theta_h} &= \int p(s_h) \left[\int p_{\theta_h}(z_h|s_h) \log \frac{p_{\theta_h}(z_h|s_h)}{m(z_h)} \mathrm{d}z_h\right] \mathrm{d}s_h \\ &= \mathbb{E}_s \left[KL(p_{\theta_h}(z_h|s_h) || m(z_h))\right]. \end{aligned}$$

Overview of ResAct

- ➢Inference: data flow
- ➢Training: gradient flow



Experimental Setup

- Dataset
  - MovieLensL-1m
  - RecL-25m

Table 1: Statistics of *RecL-25m*.

	Users	Sessions	Requests
	99,899	6,126,583	25,921,753

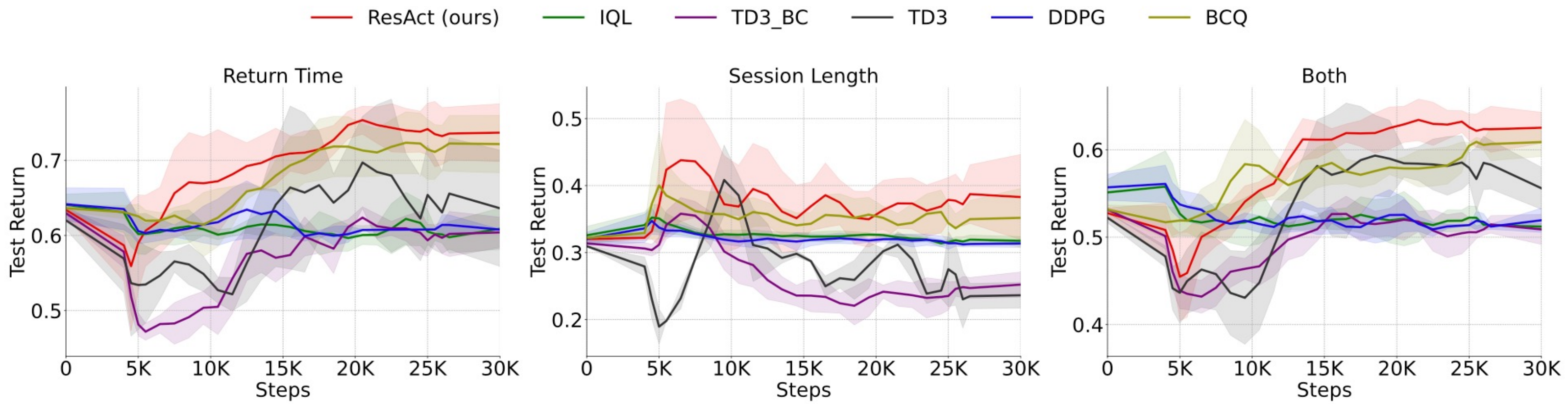
	Avg return time (h)	Avg session length	Avg # of sessions
Mean	-	4.0449	61.3277
75%	11.2794	4.8792	85
25%	4.3264	2.1358	30

- Evaluation metric: Normalized Capped Importance Sampling

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

Overall Performance

- Learning curves



- Performance comparison

Table 2: Performance comparison on MovieLensL-1m. The "±" indicates 95% confidence intervals.

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

Table 3: Performance comparison on RecL-25m in various tasks. The "±" indicates 95% confidence intervals.

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