A Comparative Study of Offline Reinforcement Learning Methods for Movie Recommendation

Changyuan Yu (cy2812), Wenhan Bai (wb2445), Chengbo Huang (ch4019), Xingru Lu (xl3602)
Columbia University
ELEN E6885 – Fall 2025

***Abstract*—Offline recommendation learning poses unique challenges due to sparse user feedback and the lack of interactive exploration. This project presents a comparative study of four representative reinforcement learning approaches on the MovieLens-1M dataset: Linear Upper Confidence Bound (LinUCB) for contextual bandits, Conservative Regression Reinforcement Learning (CRR) for advantage-weighted imitation, and two trajectory-aware deep offline RL methods—Implicit Q-Learning (IQL) and Conservative Q-Learning (CQL).**

LinUCB provides a strong contextual baseline with clear CTR gains under offline replay. CRR further improves Top-K HitRate through its shared-embedding actor–critic design and advantage-weighted behavior cloning. Incorporating SASRec-encoded user histories, IQL and CQL achieve the highest personalized ranking results, consistently outperforming the random baseline. IQL benefits from expectile-based value estimation and advantage-weighted regression, while CQL offers robust performance via pessimistic Q-value regularization.

Overall, our results show that recommendation quality improves as models adopt richer structure—from contextual features, to conservative imitation, and ultimately to sequential value learning. These findings offer practical guidance for selecting offline RL methods and point toward promising hybrid approaches that combine the strengths of existing techniques.

I. Introduction

Offline recommendation learning is challenging because it relies only on sparse historical user feedback, without any chance to gather new interactions [1]. As a result, algorithms must learn effective policies entirely from static logs, making it harder to understand user preferences and avoid mistakes [2].

In this project, we present a comparative study of four representative reinforcement learning approaches applied to the MovieLens-1M dataset [3]: Linear Upper Confidence Bound (LinUCB) for contextual bandits, Conservative Regression Reinforcement Learning (CRR) for advantage-weighted imitation, and two trajectory-aware deep offline RL methods—Implicit Q-Learning (IQL) and Conservative Q-Learning (CQL). By integrating SASRec-encoded user histories [4] with IQL and CQL, we further examine how sequential value learning compares to simpler contextual baselines.

Overall, our experiments show that recommendation quality steadily improves as the models incorporate richer structure—from simple contextual features, to conservative imitation, and finally to full sequential value learning. This progression provides practical guidance on choosing the most suitable offline RL method based on specific system requirements.

II. Background and Related Work

We first focus on the Linear UCB (LinUCB) family of algorithms [5], which models movie recommendation as a contextual bandit problem. At each discrete time step, the agent observes a context vector for each arm (movie) and selects an action to maximize the expected cumulative click-through rate (CTR). To handle data sparsity and feature interactions, variants such as Disjoint LinUCB and Hybrid LinUCB are utilized, estimating parameters either independently across arms or sharing them to capture global user demographics.

Moving beyond single-step bandits, we examine Conservative Regression Reinforcement Learning (CRR) [6] to address the limitations of offline learning where transition dynamics are absent or implicit. Our implementation utilizes a shared-embedding actor-critic model where the policy update modifies behavior cloning by weighting actions according to an advantage estimate. This allows the system to follow logged actions only when they demonstrate superior performance compared to the average alternative, effectively handling the distribution shift inherent in offline RL.

To capture the temporal structure of user preferences, we adopt two deep offline RL methods: Implicit Q-Learning (IQL) [7] and Conservative Q-Learning (CQL) [8]. IQL avoids explicit optimization over unsupported actions by using expectile-based value estimation and advantage-weighted regression, which reduces the risk of overestimating out-of-distribution (OOD) actions. Conversely, CQL directly regularizes the Q-function to be pessimistic on OOD actions, penalizing values for actions that are likely under the candidate policy but rarely observed in the dataset.

Our work distinguishes itself by combining these reinforcement learning paradigms with SASRec-encoded interaction histories [4]. While prior approaches often treat interactions as isolated events, our integration of self-attentive sequential encoders with IQL and CQL allows the model to capture sequential preference evolution, resulting in personalized rankings that surpass random baselines in both NDCG and HitRate evaluations.

III. Algorithm Development & Experimentlal Results

A. UCB

We focus on the Linear UCB (LinUCB) family of algorithms [5], modeling movie recommendation as a contextual

bandit problem. At each discrete time step t, the agent observes a context vector $\mathbf{x}_{t,a}$ for each arm (movie) a in the current candidate set $\mathcal{A}_t$, and selects an action to maximize the expected cumulative click-through rate (CTR).

1) Algorithm Formulation: We utilize a 48-dimensional context vector composed of 30 user features and 18 movie genre features. We implement and compare two primary variants of the algorithm:

a) Disjoint LinUCB: This variant assumes that parameters are independent across arms. It estimates a separate coefficient vector $\boldsymbol{\theta}_a^*$ for each arm a. The expected reward is modeled linearly as $E[r_{t,a} \mid \mathbf{x}_{t,a}] = \mathbf{x}_{t,a}^\top \boldsymbol{\theta}_a^*$. The parameters are estimated via Ridge Regression (equivalent to ℓ_2-regularized least squares). At each trial t, the agent selects the arm from the candidate set $\mathcal{A}_t$ that maximizes the upper confidence bound:

$$a_t = \underset{a \in \mathcal{A}_t}{\operatorname{argmax}} \left(\mathbf{x}_{t,a}^\top \hat{\boldsymbol{\theta}}_a + \alpha \sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}} \right), \qquad (1)$$

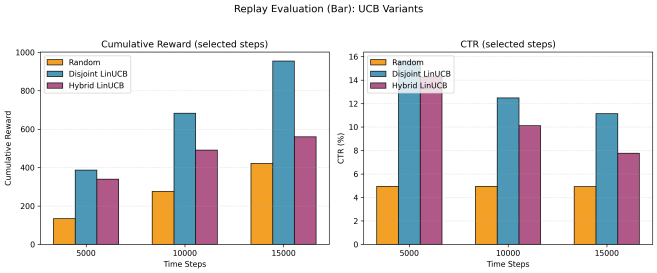
where $\mathbf{A}_a$ is the covariance matrix for arm a (initialized to the identity matrix), and α is a hyperparameter controlling the exploration-exploitation trade-off.

b) Hybrid LinUCB: To better handle data sparsity, the Hybrid variant incorporates shared parameters $\boldsymbol{\beta}^*$ common to all arms, alongside arm-specific parameters $\boldsymbol{\theta}_a^*$. The reward model is defined as $E[r_{t,a}] = \mathbf{z}_{t,a}^\top \boldsymbol{\beta}^* + \mathbf{x}_{t,a}^\top \boldsymbol{\theta}_a^*$, where $\mathbf{z}_{t,a}$ denotes features with shared effects (e.g., user demographics). Crucially, this variant retains the UCB exploration mechanism but calculates the confidence interval by accounting for the joint uncertainty of both the shared coefficients and the arm-specific parameters [5].

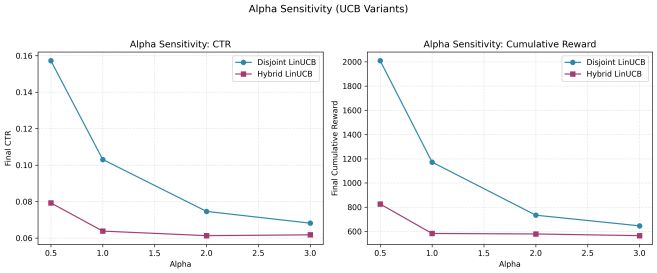
2) Experimental Setup and Results: Performance assessment employed the unbiased offline replay protocol established by Li et al. [5] applied to temporally ordered interaction logs. At every discrete time-step t, we constructed a candidate arm set $\mathcal{A}_t$ ($K = 20$) comprising the ground-truth logged movie augmented with 19 negative samples (non-interacted items). A binary reward signal $r = 1$ was assigned exclusively when the algorithm's choice aligned with the logged action and the associated user rating exceeded the threshold (≥ 4). To provide a comparative lower bound, a Random policy sampling uniformly from $\mathcal{A}_t$ was implemented, with cumulative metrics logged every 5,000 iterations.

The empirical trajectories in Fig. 1 substantiate three primary insights:

Disjoint vs. Hybrid Performance: While both contextual bandit formulations significantly eclipse the stochastic Random baseline (CTR $\approx 5\%$), Disjoint LinUCB demonstrates superior efficacy over its Hybrid counterpart (e.g., $\approx 12.5\%$ vs. $\approx 10\%$ CTR). This distinct performance margin implies that preference manifolds within the MovieLens domain are predominantly item-specific rather than globally shared. Although Hybrid LinUCB theoretically leverages collaborative features through shared parameters $\boldsymbol{\beta}$, this imposition appears to act as an over-regularizer, restricting the model's capacity to model granular, user-specific affinities—a flexibility inherent to the fully independent parameterization of the Disjoint architecture.



(a) Cumulative Reward and CTR performance.



(b) Sensitivity analysis of exploration parameter α.

Fig. 1: Performance evaluation of LinUCB variants.

Impact of Exploration Parameter (α): A sensitivity analysis across $\alpha \in \{0.5, 1.0, 2.0, 3.0\}$, detailed in Fig. 1b, reveals an inverse correlation between exploration magnitude and realized reward, optimizing at $\alpha = 0.5$. This penalty on aggressive exploration is an artifact of the offline replay estimator's structural limitations: the evaluator acknowledges only exact matches with historical logs. Consequently, inflated upper confidence bounds drive the policy toward unexplored regions which, regardless of their latent utility, yield null rewards (mismatches) in the absence of counterfactual ground truth.

Temporal Dynamics: Longitudinal analysis exposes a secular degradation in Disjoint LinUCB's CTR, drifting from $\approx 16\%$ to $\approx 11\%$ over the simulation horizon. This attenuation is likely attributable to dual mechanisms. Primarily, the inherent non-stationarity of user intent over the dataset's temporal span challenges the rigidity of static linear assumptions. Furthermore, the decline points to artifacts of policy divergence inherent to offline methodology: as the target policy evolves to maximize estimated rewards, its distribution increasingly deviates from the stochastic logging policy, thereby artificially depressing the probability of intersecting with logged hits.

B. Conservative Regression Reinforcement Learning (CRR)

We implement an offline reinforcement learning algorithm based on Conservative Regression Reinforcement Learning (CRR) and apply it to the MovieLens-1M dataset. Each user–item interaction is treated as a contextual bandit sample (s, a, r), where the state s is the user ID, the action a is the item ID, and the reward $r \in \{0, 1\}$ is derived from the explicit MovieLens rating (we assign $r = 1$ for ratings ≥ 4). The task

simplifies to learning a recommendation policy because the interactions between users do not follow a specific order in the offline data.

The structure of our implementation is inspired by the open-source EasyRL4Rec project [9], which also applies offline RL algorithms to recommender systems. While the original repository focuses on deeper sequential models and multi-step state tracking, our work adopts its core idea of using CRR for ranking but implements a simplified and fully self-contained Matrix Factorization (MF) actor–critic framework tailored for MovieLens-1M.

Our implementation uses a shared-embedding actor–critic model. Both the actor and the critic rely on the same matrix-factorization representation [10]: user embeddings u_s and item embeddings v_a. The critic estimates the value of selecting item a for user s via the inner product

$$Q(s,a) = u_s^\top v_a, \tag{2}$$

which is a standard structure in collaborative filtering. Because this is a bandit environment, the critic is trained with the regression loss

$$\mathcal{L}_{\text{critic}} = (Q(s,a) - r)^2, \tag{3}$$

without temporal-difference bootstrapping.

The CRR policy update modifies behavior cloning by weighting actions according to an advantage estimate. Since no transition dynamics exist, we use a mean-value baseline given by the average critic value across all items:

$$A(s,a) = Q(s,a) - \frac{1}{|\mathcal{A}|}\sum_{a'} Q(s,a'). \tag{4}$$

These advantages are normalized within each batch and transformed into weights using an exponential rule:

$$w(s,a) = \exp(\beta A(s,a)), \qquad w \le 20, \tag{5}$$

where β controls the strength of the preference for high-advantage actions. The weighting system of this policy enables it to follow logged actions only when these actions demonstrate superior performance compared to the average alternative solution for handling the typical distribution shift that affects offline RL. The final actor loss combines the CRR objective with a small behavior cloning term for stability:

$$\mathcal{L}_{\text{actor}} = -w(s,a)\log\pi(a|s) + 0.1\,\text{CE}(\pi(a|s), a) \tag{6}$$

We evaluate the learned policy using Top-K HitRate, which checks whether the true interacted item appears in the top-K recommendations. Figure 2 and Figure 3 show the performance over 10 training epochs.

The results show steady improvement across all Top-K metrics, indicating that CRR learns a conservative but effective policy from offline user–item data. This highlights the effectiveness of combining shared embeddings with advantage-weighted policy learning for large-scale offline recommendation.

C. Implicit Q-learning (IQL)

Contextual bandit approaches such as CRR ignore the temporal structure in MovieLens-1M, where a recommendation

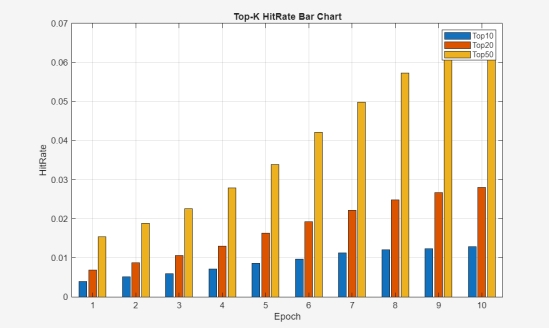


Fig. 2: CRR bar.

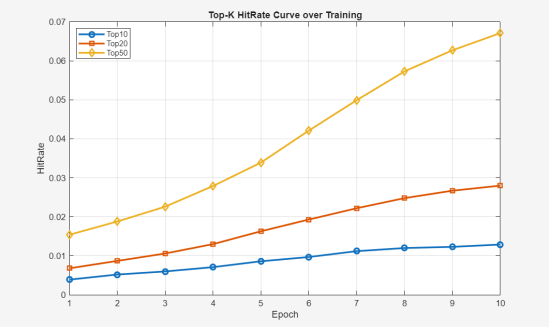


Fig. 3: CRR hitrate.

can influence future engagement. Moreover, the large combinatorial state–action space contains many rarely-observed (s,a) pairs, making offline RL vulnerable to overestimating OOD actions. These issues motivate a method that: (i) captures sequential preference evolution, and (ii) avoids explicit optimization over unsupported actions. We therefore adopt Implicit Q-Learning (IQL).

a) MDP Formulation: We model the offline recommendation task as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where:

- **State $\mathcal{S}$:** $s_t = f_{\text{enc}}(u, \text{history}_{<t})$ combines static user features (gender, age, occupation, zipcode) and SASRec-encoded interaction history [4], matching our `StateEncoder` implementation.
- **Action $\mathcal{A}$:** Recommending a movie ID from the catalog:
$$\mathcal{A} = \{1, 2, \ldots, M\}, \quad M = \#\textit{of movies}.$$
- **Reward r:** Explicit rating score from the log:
$$r(s_t, a_t) \in \{1, 2, 3, 4, 5\},$$
- **Transition P:** Offline transition dynamics from logged trajectories:
$$P(s_{t+1} \mid s_t, a_t) = P_{\mathcal{D}}(\cdot),$$
with terminal state at the end of each user sequence.
- **Discount γ:** $\gamma = 0.99$.

b) Training Procedure: The training loop performs three distinct optimization steps per batch of transitions $\mathcal{B} = \{(s, a, r, s', d)\}$.

1) **V Network Update (Expectile Regression).** The V Network is updated first using τ-expectile regression,

aiming to estimate the $\tau_{\text{expectile}}$-th expectile of the Q-value distribution, where $\tau_{\text{expectile}}$ is typically > 0.5 (default 0.7) to promote optimism.

- Objective: Minimize the expectile loss between the predicted state value $V(s)$ and the observed Q-value $Q(s,a)$, where $Q(s,a)$ is extracted from the current Q Network.
- Loss Function:

$$\mathcal{L}_V = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[L_{\tau_{\text{exp}}}(Q(s,a) - V(s))\right] \quad (7)$$

2) **Q Network Update (TD Regression).** The Q Network is updated using a standard Temporal Difference (TD) approach.

- TD Target: The target value uses the V-value of the next state s', which is approximated by the maximum Q-value from the **Target Q Network** Q_{target}.

$$Y = r + \gamma(1-d)\cdot \max_{a'} Q_{\text{target}}(s', a') \quad (8)$$

- Loss Function: Mean Squared Error (MSE) between the predicted $Q(s,a)$ and the TD target Y.

$$\mathcal{L}_Q = \mathbb{E}_{(s,a,r,s',d)\sim\mathcal{D}}\left[(Q(s,a) - Y)^2\right] \quad (9)$$

3) **Policy Network Update (Advantage-Weighted Regression).** The policy is updated using Advantage-Weighted Regression (AWR), which is a key component of IQL to avoid explicit minimization over the action space.

- Advantage Calculation: The advantage $\hat{A}(s,a)$ is computed using the newly updated Q and V networks:

$$\hat{A}(s,a) = Q(s,a) - V(s) \quad (10)$$

- Loss Function: The policy loss minimizes the weighted negative log-likelihood of the actions in the offline dataset. The weights are determined by the exponentiated advantage, encouraging the policy to mimic high-advantage actions.

$$\mathcal{L}_\pi = -\mathbb{E}_{(s,a)\sim\mathcal{D}}\big[\min\left(\exp(\beta_{\text{awr}}\hat{A}(s,a)), c_{\text{clip}}\right) \cdot \log \pi(a \mid s)\big] \quad (11)$$

where β_{awr} (default 3.0) is the inverse temperature and c_{clip} (default 20.0) limits the maximum weight to ensure stability.

4) **Target Network Synchronization.** After each update step, the Target Q Network ($\mathbf{Q}_{\text{target}}$) is softly updated towards the current Q Network using Polyak averaging with a rate τ_{target} (default 0.005):

$$\theta_{\text{target}} \leftarrow \tau_{\text{target}}\cdot\theta_Q + (1-\tau_{\text{target}})\cdot\theta_{\text{target}} \quad (12)$$

Overall Performance: As shown in Fig. 4, IQL achieves large gains over Random: for example, NDCG@50 improves from 0.0039 to 0.0185, and HitRate@50 from 0.014 to 0.075. The advantage increases as K increases.

Benefit of Offline Trajectory Learning: Improvements stem from: (1) conservative expectile value learning reducing OOD overestimation, and (2) policy refinement via advantage weighting that emphasizes high-quality logged actions.

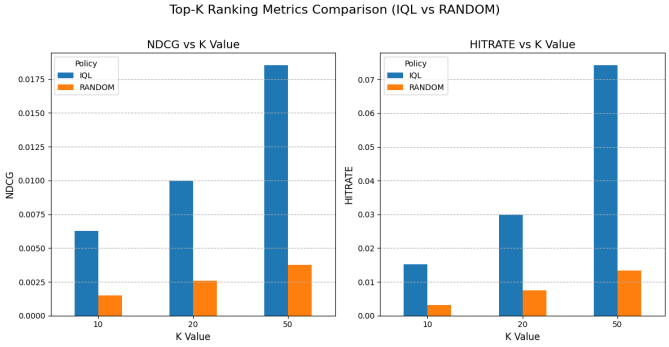


Fig. 4: Top-K ranking performance on MovieLens-1M. IQL consistently outperforms a random policy across all evaluation settings.

Personalization Ability: The SASRec-based encoder [4] enables personalized ranking rather than global item popularity. Larger K provides more opportunities to include multiple user-relevant items.

D. Conservative Q-Learning (CQL)

While IQL addresses OOD overestimation indirectly via expectile value learning and advantage-weighted policy regression, it still relies on a standard TD objective for Q-values. This can leave room for over-optimistic estimates on actions that are weakly supported by the offline dataset. Conservative Q-Learning (CQL) directly regularizes the Q-function to be *pessimistic* on out-of-distribution actions, thereby tightening the gap between estimated and realizable returns in purely offline settings.

We keep the same MDP formulation $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ and state representation $s_t = f_{\text{enc}}(u, \text{history}_{<t})$ as in the IQL section; the difference lies solely in how we fit Q and the induced policy.

a) Conservative Q Objective: Given a batch of transitions $\mathcal{B} = \{(s,a,r,s',d)\}$ drawn from the logged dataset $\mathcal{D}$, CQL augments the standard TD regression loss with a conservative regularizer. Intuitively, the method penalizes Q-values for actions that are likely under candidate policies but rarely observed in $\mathcal{D}$, and encourages Q-values on dataset actions to remain high.

For discrete actions, a common instantiation of the CQL loss is:

$$\mathcal{L}_{\text{CQL}} = \mathbb{E}_{(s,a,r,s',d)\sim\mathcal{D}}\Big[(Q(s,a) - Y)^2\Big] + \alpha\cdot\mathcal{R}_{\text{cons}}(s), \quad (13)$$

where the TD target Y is defined as

$$Y = r + \gamma(1-d)\max_{a'} Q_{\text{target}}(s', a'), \quad (14)$$

and the conservative regularizer compares the Q-values under a candidate action distribution and the logged distribution:

$$\mathcal{R}_{\text{cons}}(s) = \Big(\mathbb{E}_{a\sim\pi(\cdot|s)}[Q(s,a)] - \mathbb{E}_{a\sim\mathcal{D}(\cdot|s)}[Q(s,a)]\Big). \quad (15)$$

In our discrete MovieLens setting, we approximate $\mathbb{E}_{a\sim\pi(\cdot|s)}[Q(s,a)]$ using the softmax policy induced by the current Q-network,

$$\pi(a|s) \propto \exp(Q(s,a)), \tag{16}$$

and compute the data expectation by evaluating $Q(s,a)$ on the logged action a for each transition. The coefficient $\alpha > 0$ controls the strength of conservatism: larger α pushes Q-values down for actions outside the support of the dataset, at the cost of underestimating returns if set too aggressively.

b) Policy Extraction: Unlike IQL, CQL does not require an auxiliary value network $V(s)$. We maintain a separate policy network $\pi_\theta(a|s)$ in our implementation for consistency with the IQL pipeline, but conceptually one can simply interpret a softmax over Q-values as the target policy. In practice, we update π_θ to track the Q-function by maximizing the expected Q-value under the policy:

$$\mathcal{L}_\pi^{\text{CQL}} = -\mathbb{E}_{s\sim\mathcal{D}}\Big[\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[Q(s,a)]\Big], \tag{17}$$

or, in an equivalent discrete form,

$$\mathcal{L}_\pi^{\text{CQL}} = -\mathbb{E}_{s\sim\mathcal{D}}\Big[\sum_a \pi_\theta(a|s)\, Q(s,a)\Big]. \tag{18}$$

This encourages the actor to place probability mass on actions with higher conservative Q-values, while the CQL regularizer ensures that these Q-values remain pessimistic on unsupported actions.

c) Target Networks and Optimization: Analogous to IQL, we employ a target Q-network Q_{target} to stabilize TD learning. After each gradient step on Q, we apply Polyak averaging:

$$\theta_{\text{target}} \leftarrow \tau_{\text{target}}\, \theta_Q + (1-\tau_{\text{target}})\, \theta_{\text{target}}, \tag{19}$$

with a small τ_{target} (e.g., 0.005). We optimize $\mathcal{L}_{\text{CQL}}$ and $\mathcal{L}_\pi^{\text{CQL}}$ jointly over minibatches drawn from the same MovieLens trajectories as in the IQL experiments, sharing the identical state encoder and reward definition.

d) Random vs IQL vs CQL: Experimental Results: We compare three policies on the held-out MovieLens-1M split:

- **Random**: a non-personalized baseline that assigns i.i.d. uniform scores to all candidate movies.
- **IQL**: the implicit Q-learning policy described above, trained with expectile value regression and advantage-weighted regression.
- **CQL**: our conservative Q-learning policy trained with the CQL-augmented TD loss and policy improvement objective.

All policies are evaluated using the same Top-K ranking metrics, NDCG@K and HitRate@K for $K \in \{10, 20, 50\}$, on the test trajectories.

Overall Performance: As visualized in Fig. 5, both offline RL methods substantially outperform the Random baseline across all evaluated K. This indicates that the learned policies successfully exploit the long-term structure of user trajectories rather than merely memorizing individual interactions. The relative performance between IQL and CQL reflects their different regularization philosophies: IQL leans on optimistic value estimates filtered through advantage weighting, whereas CQL explicitly penalizes overestimation on unsupported actions.

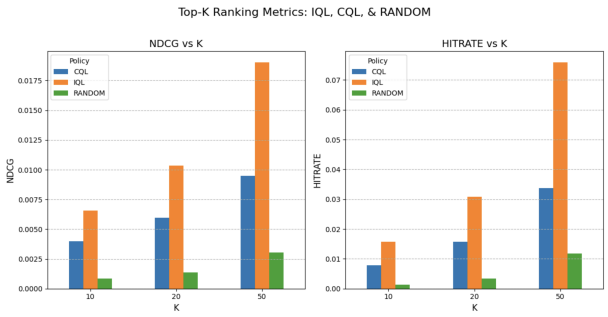


Fig. 5: Top-K ranking performance on MovieLens-1M for Random, IQL, and CQL.

Effect of Conservative Regularization: CQL's conservative penalty tends to shrink Q-values on actions that are rarely seen in the training data. In recommendation, this trades off exploratory recommendations against robustness: the policy is discouraged from recommending items that are not well-supported by the logs, which can be beneficial when the dataset is sparse or biased. In our experiments, this manifests as a policy that is competitive with IQL in NDCG@K and HitRate@K while exhibiting more stable training dynamics as hyperparameters (e.g., learning rate or batch size) are varied.

Personalization Across K: For both IQL and CQL, performance generally improves as K grows, since larger recommendation lists provide more opportunities to include user-relevant items. The SASRec-based state encoder is crucial here: it aggregates long-term consumption patterns, enabling the policies to go beyond global popularity and adapt rankings to each user's historical tastes. The gap between the learned policies and the Random baseline across K highlights the benefit of sequence-aware offline RL on MovieLens-1M.

IV. Conclusion

In this project, we conducted a systematic comparison of several representative learning paradigms for offline recommendation, including contextual bandits (LinUCB), conservative imitation-based offline RL (CRR), and two value-driven deep offline RL methods (IQL and CQL). Using the MovieLens-1M dataset as our benchmark, we found that each class of algorithms offers its own strengths within the offline recommendation landscape.

LinUCB serves as an efficient and strong baseline because it uses contextual relationships to make predictions without relying on sequential information. CRR builds on this by incorporating advantage-weighted behavior cloning, which improves Top- K HitRate while remaining stable under offline data limitations. The highest performance level emerges from IQL and CQL which use SASRec-based user-history encodings to track user sequence patterns. The system uses expectile-based value estimation together with advantage-weighted regression to achieve better results than CQL does.

The system implements explicit pessimism in CQL to prevent overestimation of unsupported actions which produces more reliable policies when the available data is limited.

Overall, our results show that recommendation accuracy improves as models incorporate more structure—from contextual modeling to conservative imitation and finally to trajectory-aware value learning. These findings provide guidance for selecting methods based on system needs: bandit approaches suit simpler settings, CRR offers stable imitation learning, and IQL/CQL provide stronger personalized ranking when sequential data are available. Future work may explore hybrid models that combine these strengths and evaluate them under diverse logging strategies and data conditions.

REFERENCES

[1] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[2] X. Zhao, L. Xia, L. Zhang, Z. Ding, D. Yin, and J. Tang, "Deep reinforcement learning for search, recommendation, and online advertising: A survey," *ACM SIGWEB Newsletter*, no. Spring, pp. 1–15, 2019.

[3] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems (TIIS)*, vol. 5, no. 4, pp. 1–19, 2015.

[4] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 197–206.

[5] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th International Conference on World Wide Web (WWW)*, 2010, pp. 661–670.

[6] Z. Wang, A. Novikov, K. Zolna, J. Merel, J. T. Springenberg, S. E. Reed, B. Shahriari, N. Siegel, C. Gulcehre, N. Heess *et al.*, "Critic regularized regression," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 7768–7778.

[7] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *International Conference on Learning Representations (ICLR)*, 2022.

[8] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative q-learning for offline reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1179–1191.

[9] "Easyrl4rec: Offline reinforcement learning for recommendation," https://github.com/SimonCuCu/EasyRL4Rec, 2024.

[10] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.