Reinforcement Learning

Clickbait

Comparing different RL algorithms for ad-selection

Angad
Bharat
Jiya
Pratham

Problem

Comparison of different RL models in optimising ad-selection using more than just CTR

Using various **metrics** to analyze the performance of models such as **Epsilon-Greedy**, **Deep-Q**, **Softmax**, etc on real world online advertising data.

RL Formulation of Problem

01

Data Set

02

States

03

Actions

RL Formulation of Problem

01

Data Set

Size of the complete data: 38GB

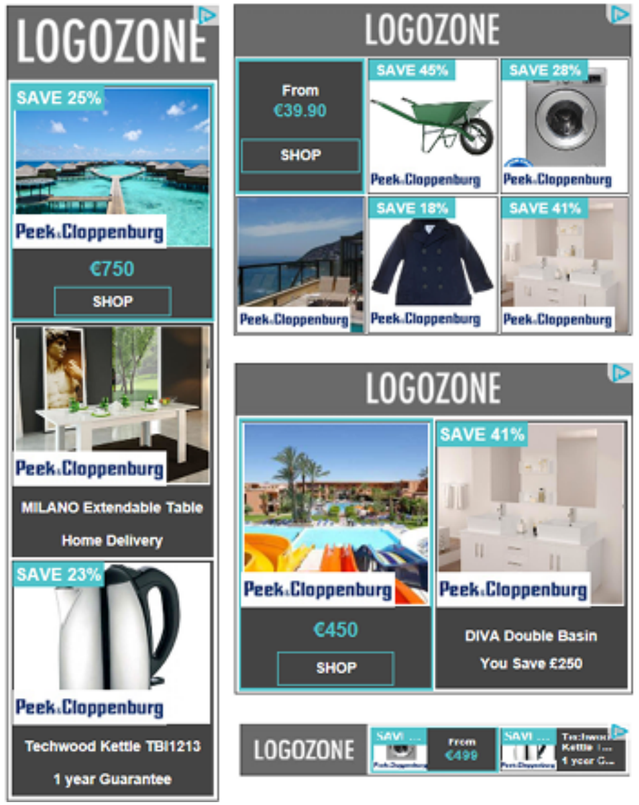
Plan to train models on a smaller sample of data provided by the same authors

The task is to choose the products to display in the ad knowing the banner type in order to maximize the number of clicks

For each user impression,

- **User context**: c
- **Number of slots in the banner type**: lc
- **Candidate pool of products p**: Pc

Logging policy pi, probabilistically selects products to construct a banner, using a Plackett-Luce ranking model.



An example of a banner with 4 ad slots

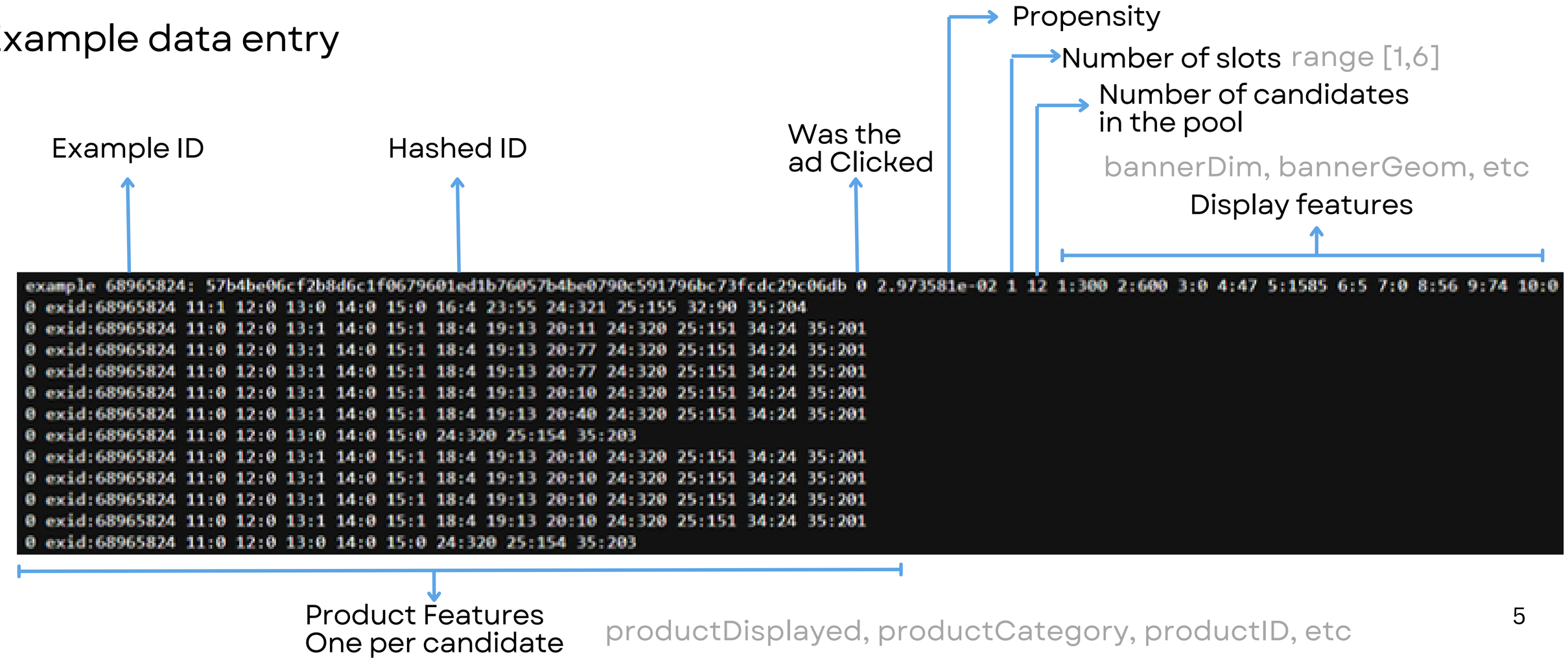
Subsampled non-clicked examples (keeping only 10%) to reduce dataset size.

RL Formulation of Problem

01

Data Set

Example data entry

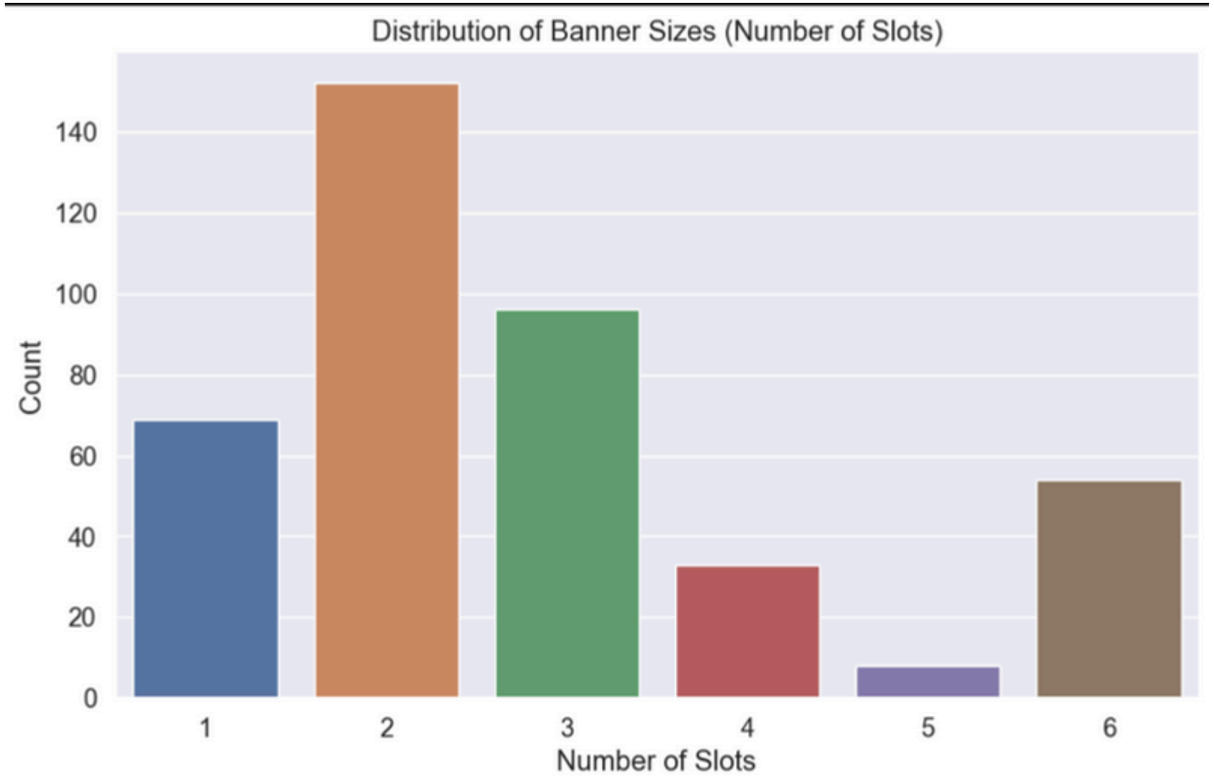


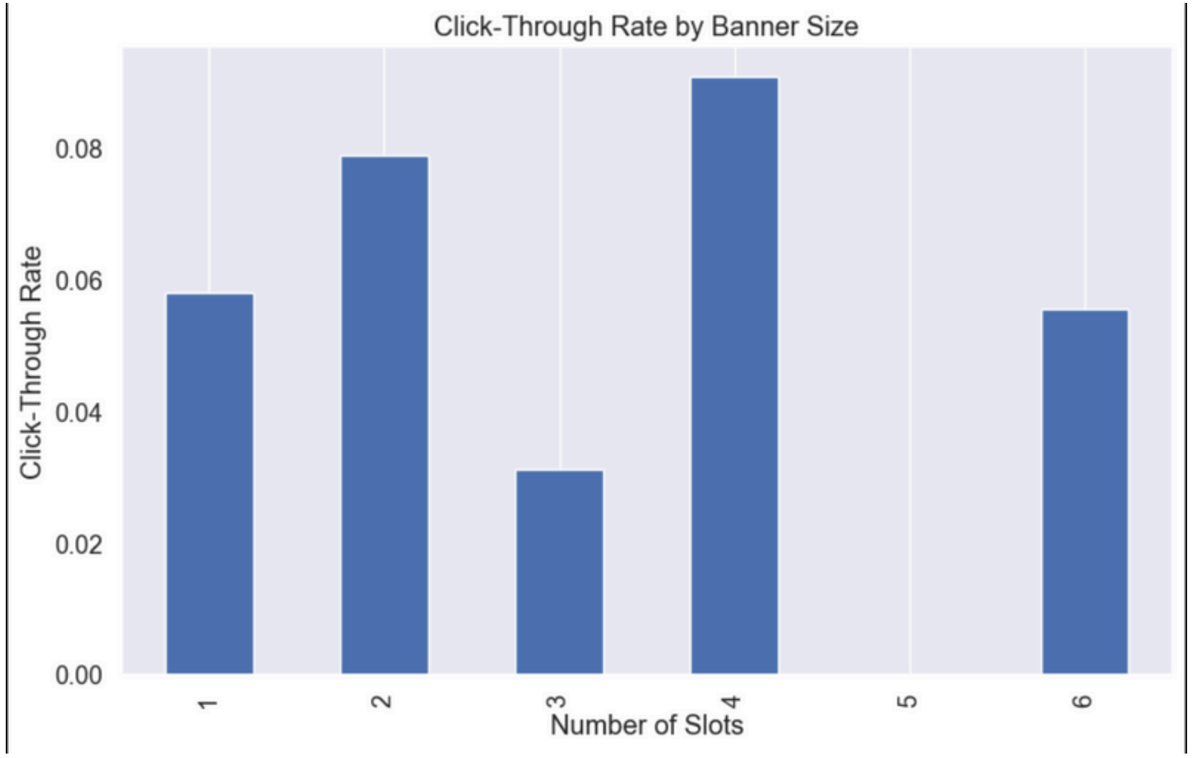
RL Formulation of Problem

01

Data Set

EDA





RL Formulation of Problem

02, 03

States, Actions, Rewards

States(S)

- **User Context**: Information about the user at the time of impression.
- **Banner Type**: Characteristics such as the number of ad slots.
- **Candidate Pool (P)**: Set of available products for each impression.

Actions (A)

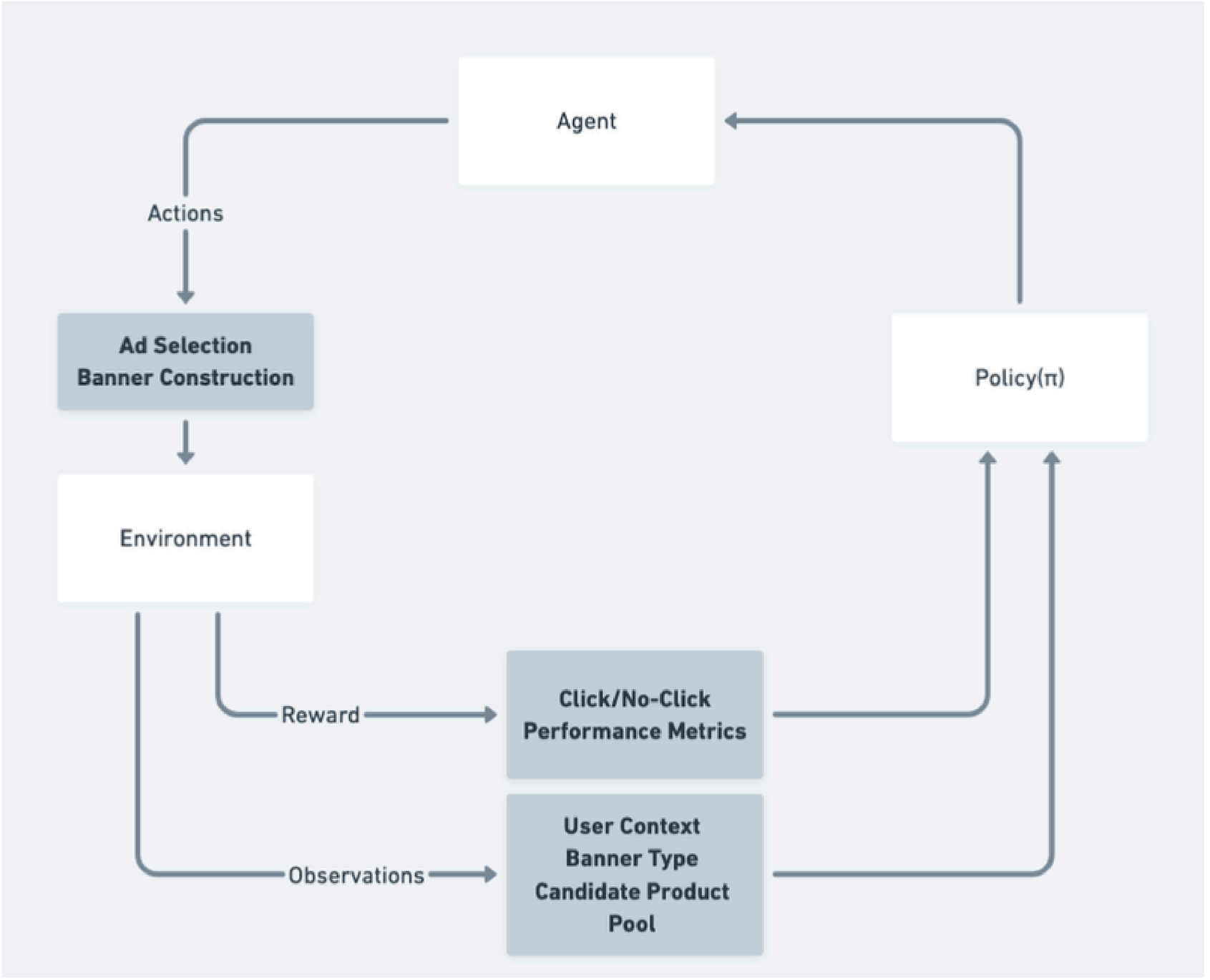
- **Ad Selection**: Choosing which ad(s) to display in each banner slot.
- **Banner Construction**: Assembling a banner by selecting multiple ads from the candidate pool.

Reward (R)

- **Feedback** from user interaction (click = 1, no-click = 0)
- **Performance Metrics**: Evaluated via methods like Inverse Propensity Scoring (IPS) and Self-Normalized IPS (SNIPS).

Policy (π):

- A **probabilistic** mechanism used to construct banners by randomly selecting products.
- Ensures diverse data collection for future model improvements.


Agent
Actions
Ad Selection
Banner Construction
Policy(π)
Environment
Reward
Click/No-Click
Performance Metrics
Observations
User Context
Banner Type
Candidate Product
Pool

Literature Survey

Problem and methods used currently

- DEAR: Deep Reinforcement Learning for online advertisig impression in Recommender Systems
- Large-scale Validation of Counterfactual Learning Methods: A Test-Bed

DEAR: Deep Reinforcement Learning for Online Advertising Impression in Recommender Systems

- Proposes a **Deep Q-network (DQN)-based reinforcement** learning framework for ad placement in recommender systems.
- **Key Features:**
  - Determines **three key decisions jointly:**
    - Whether to insert an ad in a recommendation list.
    - Which ad to select for display?
    - Where to place the ad in the list.
- **Novel DQN architecture** handles multiple decisions simultaneously, reducing computational complexity.
- Balances **maximizing ad revenue** and **minimizing negative user experience**.
- Evaluated on **real-world Douyin Short Video** dataset with **improvements over baselines like Wide & Deep Learning and GRU-based models**
- Findings: The **DEAR framework outperforms conventional models** by optimizing long-term user engagement and ad revenue tradeoffs.

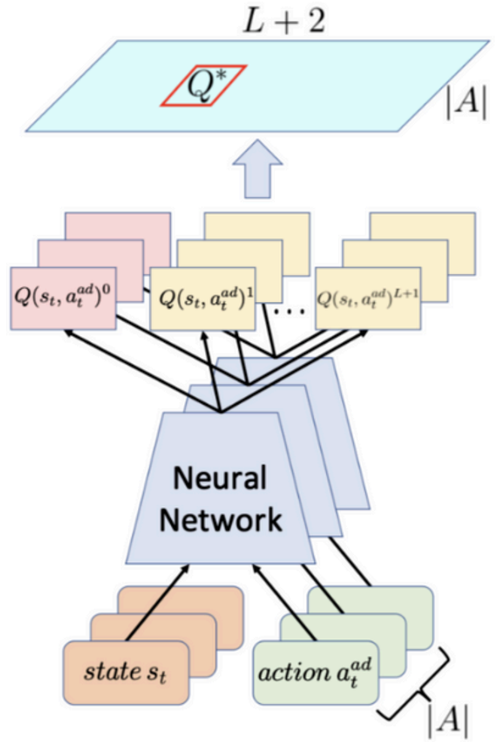


Fig. 1. The Novel DQN architecture for online advertising.

Zhao, X., Gu, C., Zhang, H., Yang, X., Liu, X., Tang, J., & Liu, H. (2021). DEAR: Deep Reinforcement Learning for Online Advertising Impression in Recommender Systems.

Large-scale Validation of Counterfactual Learning Methods: A Test-Bed

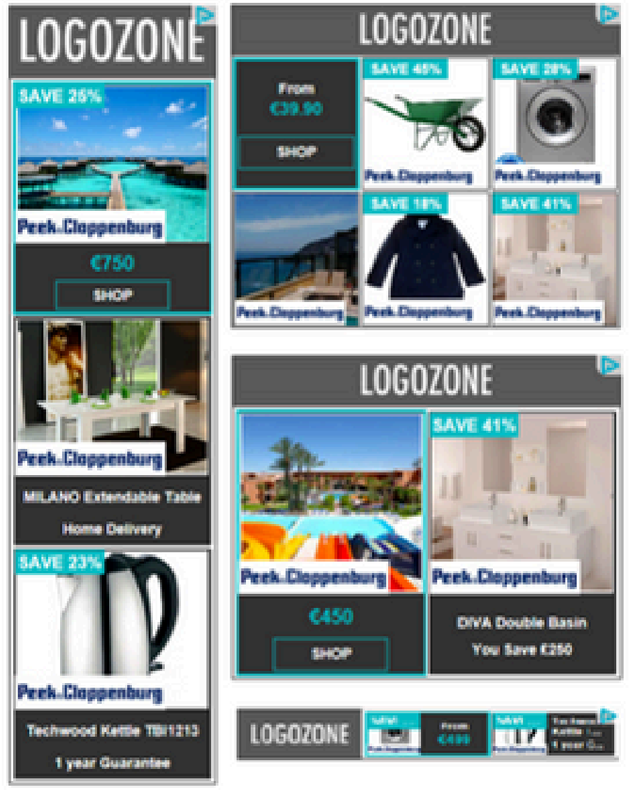
Objective: Uses **off-policy learning to optimize banner ad placement**

Focuses on batch learning from **bandit feedback (BLBF) for ad selection** using logged user interactions.

Key Methods:

- **Doubly robust optimization (DRO):** Combines regression with inverse propensity scoring (IPS) to improve policy learning.
- **POEM (Policy Optimization with Estimated Marginals):** Reduces variance in policy optimization for better ad selection.
- **Supervised Learning Baselines** (Regression, IPS) for CTR prediction and ad ranking.

Findings: Counterfactual learning methods significantly outperform traditional supervised learning in optimizing ad layouts



Four examples of ads used in display advertising: a vertical ad, a grid, and two horizontal ads (mock-ups).

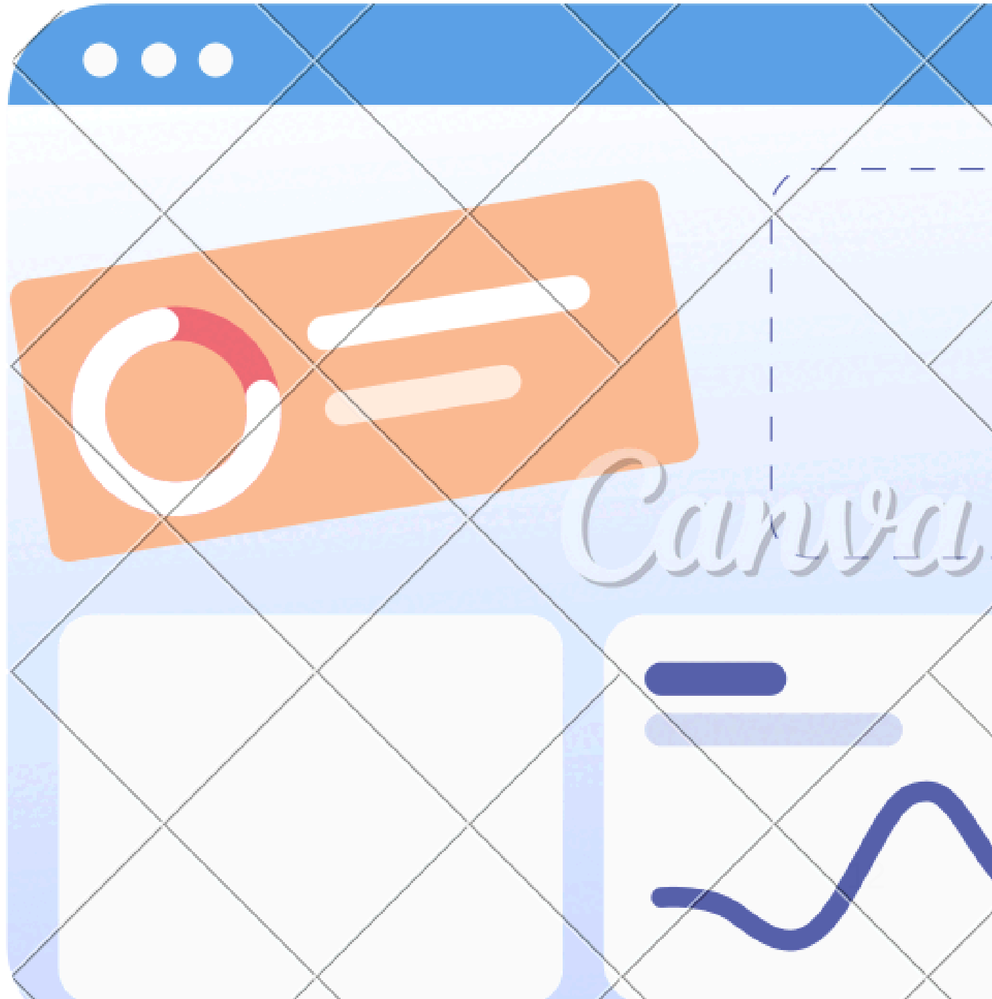
Lefortier, D., Swaminathan, A., Gu, X., Joachims, T., & De Rijke, M. (2016). Large-scale validation of counterfactual learning methods: a Test-Bed.

Our Plan

The models we plan to implement:

- **Epsilon Greedy**
- UpperConfidence Bound (UCB)
- SoftMax (Boltzmann Exploration)
- Deep Q Network
- Deep Deterministic Policy Gradient (DDPG)

Implementation of the **remaining models** and an **extensive analysis** on their performance is what we plan to do for the rest of the semester



Epsilon Greedy

Initial Outputs

```
[22:28:37] Computed Metrics over 19891 impressions:
IPS estimate: 0.007930607259468297
SNIPS estimate: 0.6423254170799739
[22:28:37] Offline metric computation completed.
```

A) Metrics Calc for Epsilon Greedy

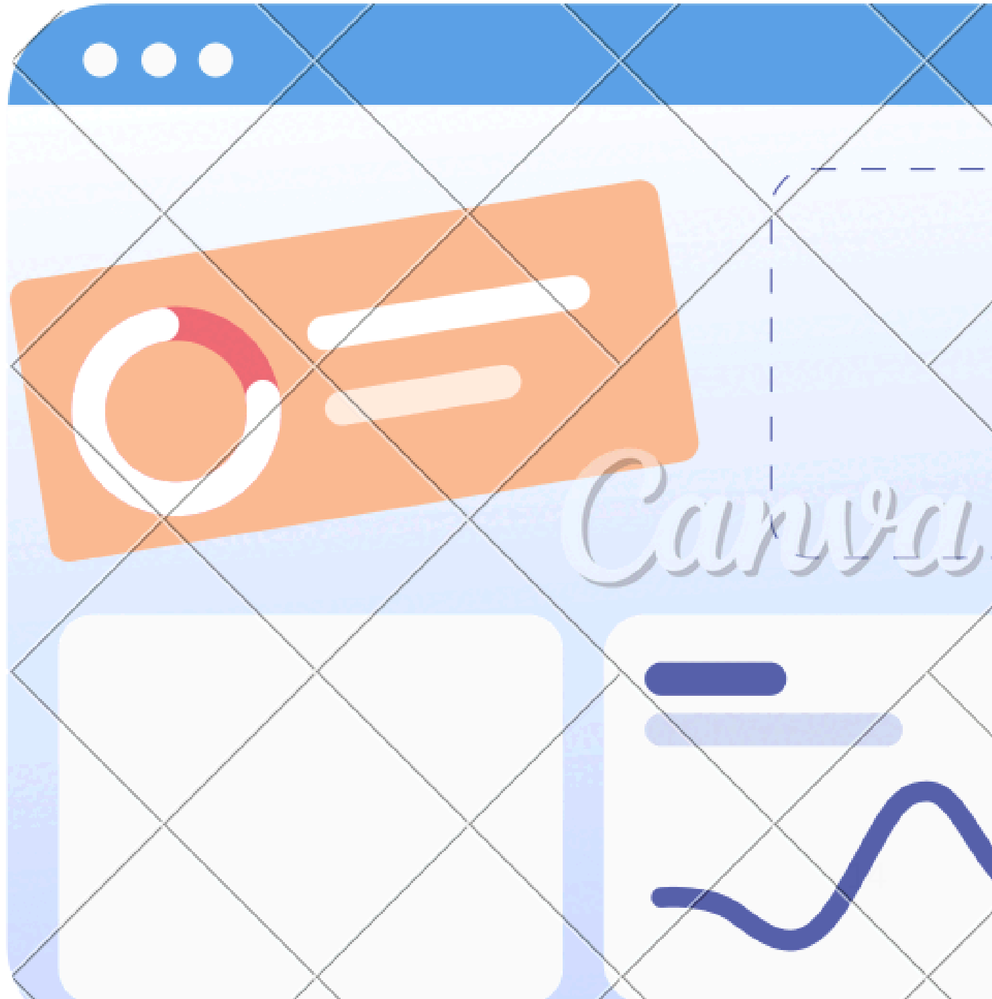
```
[22:34:45] Starting epoch 1/3...
[22:34:48] Epoch 1: Processed and updated model with 10000 impressions.
[22:34:50] Epoch 1: Final batch processed. Total impressions this epoch: 19891.
[22:34:50] Epoch 1 complete.
```

B) Included batches and epochs to handle complete Train Data

Evaluation Metrics

We plan to use two metrics:

- **IPS** (Inverse Propensity Scoring)
- **SNIPS** (Self-Normalized IPS)



1. IPS (Inverse Propensity Scoring) is the average weighted reward.

$$\hat{R}(\pi) = \frac{1}{N} \sum_{i=1}^{N} \delta_i \frac{\pi(y_i|x_i)}{q_i}$$

2. SNIPS (Self-Normalized IPS) is the total weighted reward divided by the total weight.

$$\hat{C}(\pi) = \frac{1}{N} \sum_{i=1}^{N} \frac{\pi(y_i|x_i)}{q_i}$$

$$\hat{R}_{snips}(\pi) = \frac{\hat{R}(\pi)}{\hat{C}(\pi)}$$

3. How different values of hyperparameters affect the outcome in each proposed methodology?

Thank You