Hybrid Transfer in Deep Reinforcement Learning for Ads Allocation

Ze Wang∗ Meituan Beijing, China wangze18@meituan.com
Guogang Liao∗ Meituan Beijing, China liaoguogang@meituan.com
Xiaowen Shi‡ Meituan Beijing, China shixiaowen03@meituan.com
Xiaoxu Wu Meituan Beijing, China wuxiaoxu04@meituan.com
Chuheng Zhang‡ IIIS, Tsinghua University Beijing, China zhangchuheng123@live.com
Bingqi Zhu Meituan Beijing, China zhubingqi@meituan.com
Yongkang Wang Meituan Beijing, China wangyongkang03@meituan.com
Xingxing Wang Meituan Beijing, China wangxingxing04@meituan.com
Dong Wang Meituan Beijing, China wangdong07@meituan.com

ABSTRACT
Ads allocation, which involves allocating ads and organic items to limited slots in feed with the purpose of maximizing platform revenue, has become a research hotspot. Notice that, platforms (e.g., e-commerce platforms, video platforms, food delivery platforms and so on) usually have multiple entrances for different categories and some entrances have few visits. Data from these entrances has low coverage, which makes it difficult for the agent to learn. To address this challenge, we propose Similarity-based Hybrid Transfer for Ads Allocation (SHTAA), which effectively transfers samples as well as knowledge from data-rich entrance to data-poor entrance. Specifically, we define an uncertainty-aware similarity for MDP to estimate the similarity of MDP for different entrances. Based on this similarity, we design a hybrid transfer method, including instance transfer and strategy transfer, to efficiently transfer samples and knowledge from one entrance to another. Both offline and online experiments on Meituan food delivery platform demonstrate that the proposed method could achieve better performance for data-poor entrance and increase the revenue for the platform.

KEYWORDS
Ads Allocation, Reinforcement Learning, Transfer Learning

ACM Reference Format:
Ze Wang, Guogang Liao, Xiaowen Shi, Xiaoxu Wu, Chuheng Zhang, Bingqi Zhu, Yongkang Wang, Xingxing Wang, and Dong Wang. 2022. Hybrid Transfer in Deep Reinforcement Learning for Ads Allocation. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM ’22), October 17–21, 2022, Atlanta, GA, USA. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557611

1 INTRODUCTION
Ads and organic items are mixed together and displayed to users in e-commerce feed nowadays [4, 7, 19]. How to allocate limited ads slots reasonably and effectively to maximize platform revenue has attracted growing attention [10, 17, 20]. Several recent strategies for ads allocation model the problem as Markov Decision Process (MDP) [14] and solve it using reinforcement learning (RL) [3, 8, 20–22]. For instance, Xie et al. [18] propose a hierarchical RL-based framework to first decide the type of the item to present and then determine the specific item for each slot. Liao et al. [8] propose CrossDQN which takes the crossed items according to the action as input and allocates the slots in one screen at a time.

However, these excellent RL-based algorithms face one major challenge when applied in platforms(e.g., e-commerce platforms, video platforms, food delivery platforms and so on). As shown in Figure 1, the Meituan food delivery platform has multiple entrances for different categories (e.g., Homepage, Food, Desert and so on), facilitating users to access different categories of content. Some entrances have few visits, resulting in a low data coverage. This further makes it difficult for the agent to learn. Therefore, it is desirable for a learning algorithm to transfer samples and leverage knowledge acquired in data-rich entrance to improve performance of the agent for ads allocation in data-poor entrance. Motivated by this phenomenon, we incorporate Transfer learning (TL) technology into RL to solve this problem.
we design a hybrid transfer framework, which consists of instance which is predicted by pre-trained NSR model. And we define a

1The code and data example are publicly accessible at https://github.com/princewen/SHTAA

For instance, Tirinzoni et al. [5,9,15,16,23] present a algorithm called IWFQI for transferring samples in batch RL that uses importance weighting to automatically account for the difference in source and target distributions. But IWFQI does not fully exploit possible similarities between tasks. Liu et al. [9] quantify the environmental dynamics of an MDP by the N-step return (NSR) values and present a knowledge transfer method called NSR-based value function transfer. However, they ignore the uncertainty of the NSR model itself when transferring, since the uncertainty of NSR can be measured by the uncertainty of function

We impose a prior distribution

and allocate for each screen sequentially. So the ads allocation problem for different entrances can be formulated as an MDP, using a tuple \((S, \mathcal{A}, r, P, \gamma)\), which specifies the state space \(S\), the action space \(\mathcal{A}\), the reward \(r\), the state transition probability \(P\) and the discount factor \(\gamma\). Since this paper mainly focuses on how to transfer knowledge from data-rich entrance to data-poor entrance to improve the performance of the agent for ads allocation, we mainly follow the recent work - CrossDQN [8] in the detailed definition of elements (i.e., \(S, \mathcal{A}, r, P, \gamma\)) on ads allocation, to simplify our content and reduce the cost of understanding for readers. Besides, we define the N-step return (NSR) [9] after current state \(s_t\) as follows:

\[
\begin{align*}
    r_t^N = r_t + r_{t+1} + \cdots + r_{t+N-1} \\
    \end{align*}
\]

We denote the dataset for source task as \(D_S = \{ (s, a, r^N, s') \}\), and the dataset for target task as \(D_T = \{ (s, a, r^N, s') \}\). The objective is to find an ads allocation policy for the target task based on \(D_S\) and \(D_T\) to maximize the cumulative reward.

3 METHODOLOGY

In this section, we will introduce SHTAA in detail. Two main ideas are: i) using predicted uncertain-aware NSR to measure MDP similarity, ii) proposing a hybrid transfer approach (consisting of instance transfer and strategy transfer) for selective transfer and avoidance of negative transfer.

3.1 Uncertainty-Aware MDP Similarity

Liu et al. [9] demonstrate the effectiveness of using NSR to quantify the environmental dynamics of an MDP and transfer based on it. But they ignore the uncertainty of the NSR model itself. In this paper, we propose an uncertainty-aware MDP similarity concept in which we combine high-dimensional representation capability of deep learning model and distribution modeling capability of Gaussian process (GP) [12] to construct the NSR model.

Specifically, we assume a GP prior distribution over \(f\), i.e., \(f \sim \mathcal{GP}(m(s, a), k((s, a), (s, a)'))\), where \(m(s, a) = E[f(s, a)]\) is the mean function and \(k(s, a), (s, a)')\) is the covariance kernel function. Formally, we define \(f : (S, \mathcal{A}) \rightarrow \mathbb{R}\) as the prediction model of NSR. Under a Bayesian perspective of function learning, the uncertainty of NSR can be measured by the uncertainty of function \(f\). We impose a prior distribution \(p(f)\) to the function \(f\) and (approximately) infer the posterior distribution \(p(f|D)\) over functions as the learned model:

\[
    p(f|D) = \frac{p(f)p(D|f)}{p(D)}
\]

We aim to estimate the function \(f^* = f((s, a)_i)\) as well as its uncertainty on a test sample \((s, a)_i\), given the observed samples \(D^b = \{ (s_i, a_i, r^N) \}_{i=1}^b\) at batch \(B\). Let \((S, A)_b \in \mathbb{R}^{bd}\) denote the matrix of the observed feature vectors, \(f_b = f((S, A)_b) \in \mathbb{R}^b\) be the vector of the corresponding function values, and \(y_b = [r^N_1, \cdots, r^N_b]^\top \in \{0,1\}^b\) denote the vector of user feedback. Then the likelihood of \(r^N\), \(f_b\) and \(f_i\) is:

\[
    p(y_b, f_b, f_i) = p(y_b|f_b)p(f_b, f_i) = p(f_b, f_i) \prod_{i=1}^b p(y_i|f_i),
\]
3.2 Hybrid Transfer

The hybrid transfer method, which is shown in Figure 2, consists of instance transfer and strategy transfer. Next we will introduce each part separately.

### 3.2.1 Instance Transfer

The key challenge in instance transfer is to avoid the negative transfer. In this paper, we propose an instance transfer method based on the uncertainty-aware MDP similarity. Specifically, given a sample \((s, a, r, r^N_s, w)\) \(\in D_S\) and a weight threshold \(\tau\), the local environmental dynamics of \((s, a)\) in the source task is regarded similar to that in the target task if \(w \geq \tau\). In this case, the sample for the source task can be added directly into the mixed dataset. And when \(w < \tau\), which means that the local environmental dynamics related to \((s, a)\) in the source task and target task are different, this sample will be filtered out.\(^2\) In this way, the selective instance transfer based on uncertainty-aware MDP similarity can effectively avoid negative transfer.

### 3.2.2 Strategy Transfer

The pre-trained agent for the source task (hereinafter referred to as agent\(_S\)) can guide the learning of the agent for the target task (hereinafter referred to as agent\(_T\)). Specifically, we first constrain the action space of agent\(_T\)’s target value function based on the output Q-value of agent\(_S\). The RL loss for agent\(_T\) is calculated as follows:

\[
\text{Loss}_{RL} = (r + \gamma \max_{a' \in A_T} Q_T(s', a') - Q_T(s, a))^2,
\]

where \(A_T\) is a set of \(\beta\) actions corresponding to the top-\(\beta\) highest \(Q_S(s', a)\) and \(\beta\) is determined by \(w\). By restricting the action space, the agent\(_T\) are forced to behave similar to agent\(_S\).

Second, the output Q-value of agent\(_T\) can also guide the learning process of agent\(_T\). We take \(Q_S(s, a)\) as the learning target of agent\(_T\):

\[
\text{Loss}_{STL} = (Q_S(s, a) - Q_T(s, a))^2.
\]

The similarity-based weight is used to adjust the extent of strategy transfer, w.r.t. the loss:

\[
\text{Loss} = \frac{1}{|B|} \sum_{(s, a, r, r^N_s, w) \in B} \left( w \cdot \text{Loss}_{STL} + \text{Loss}_{RL} \right).
\]

### 3.3 Offline Training

We follow the offline RL paradigm, and the process of offline training is shown in Algorithm 1. We first pre-train the NSR models and agent\(_S\). Then we train the agent\(_T\) through SHTAA.

**Algorithm 1 Offline training of SHTAA**

1. Source dataset \(D_S\), target dataset \(D_T\)
2. pre-train
3. Train the NSR Model for source task on \(D_S\)
4. Train the NSR Model for target task on \(D_T\)
5. Train the agent\(_S\) on \(D_S\)
6. train
7. Calculate the similarity-based weight \(w\) for each sample
8. Filter samples in \(D_S\) based on weight \(w\) and threshold \(\tau\)
9. Merge filtered \(D_S\) and \(D_T\) as \(D = \{(s, a, r, r^N_s, w)\}\)
10. Initialize agent\(_T\) with random weights
11. repeat
12. Sample a batch \(B\) of \((s, a, r, r^N_s, w)\) from \(D\)
13. Update network parameters by minimizing Loss in (10) until Convergence

\(^2\)We have tried weighted transfer in instance transfer, the performances are close.
3.4 Online Serving
In the online serving system, the agent for ad allocation in target entrance selects the action with the highest reward based on current state and converts the action to ads slots set for the output. When user pulls down, state is updated and the above process is repeated.

4 EXPERIMENTS
4.1 Experimental Settings
4.1.1 Dataset. We collect the dataset by running an exploratory policy on Meituan food delivery platform during January 2022. The dataset contains 12,411,532 requests from 1,919,201 users in the source entrance and 813,271 requests from 214,327 users in the target entrance. Each request contains several transitions.

4.1.2 Evaluation Metrics. We evaluate different methods with ads revenue $R^{ad}$ and service fee $R^{fee}$. Follow the definition in [8].

4.1.3 Parameters Settings. The hidden layer sizes of the NSR models is (128, 64, 32). The structure and parameters of the agents follow the work in CrossDQN [8]. The learning rate is $10^{-3}$, the optimizer is Adam [6] and the batch size is 8,192. $\tau$ is 0.7 and $N$ is 3.

4.2 Offline Experiment
In this section, we validate our method on offline data and evaluate the performance using an offline estimator. Through extended engineering, the offline estimator models the user preference and aligns well with the online service.

4.2.1 Baselines & Ablations. We study four baselines and three ablated variants to verify the effectiveness of SHTAA.
- DEAR [21] is an advanced DQN architecture to jointly determine three related tasks for ads allocation. Here we train it on $D_T$.
- CrossDQN [8] is an advanced method for ads allocation. Here we take it as the structure of agent$_T$ and train agent$_T$ on $D_T$.
- Cross DQN (w/ $D_S$) transfers all samples in source dataset into the training of agent$_T$ based on the previous baseline.
- IWFQI [16] is a algorithm for transferring samples in batch RL that uses importance weighting to automatically account for the difference in the source and target distributions.
- NSR-CrossDQN. Liu et al. [9] propose the NSR-based value function transfer method. Here we implement this transfer method on $D_S$ and $D_T$ based on CrossDQN.
- SHTAA (w/o UA-Sim) does not use uncertainty-aware MDP similarity and uses the MDP similarity concept defined in the NSR-based value function transfer method [9] instead.
- SHTAA (w/o AC) does not use action constraint in SHTAA.
- SHTAA (w/o Loss$_{TL}$) does not use Loss$_{TL}$ in SHTAA.

4.2.2 Performance Comparison. We present the experimental results in Table 1 and have the following findings: i) The performance of SHTAA is superior to all baselines, which mainly justifies that SHTAA can selectively transfer the samples and knowledge from source task to target task. ii) Compared with IWFQI, the superior performance of our method mainly justifies the effectiveness of our strategy transfer. iii) Compared with NSR-CrossDQN, the superior performance of our method mainly justifies the effectiveness of our uncertainty-aware MDP similarity concept and action constraint.

| model                  | $R^{ad}$          | $R^{fee}$         |
|------------------------|-------------------|-------------------|
| DEAR                   | 0.2389 (±0.0008)  | 0.2471 (±0.0013)  |
| CrossDQN               | 0.2450 (±0.0004)  | 0.2492 (±0.0007)  |
| CrossDQN (w/ $D_S$)    | 0.2461 (±0.0005)  | 0.2497 (±0.0009)  |
| IWFQI                  | 0.2463 (±0.0004)  | 0.2505 (±0.0007)  |
| NSR-CrossDQN           | 0.2471 (±0.0009)  | 0.2518 (±0.0009)  |
| SHTAA                  | 0.2562 (±0.0003)  | 0.2596 (±0.0005)  |
| - w/o UA-Sim           | 0.2465 (±0.0004)  | 0.2521 (±0.0001)  |
| - w/o AC               | 0.2503 (±0.0001)  | 0.2561 (±0.0002)  |
| - w/o Loss$_{TL}$      | 0.2527 (±0.0006)  | 0.2572 (±0.0006)  |
| Improvement            | 3.68%             | 3.09%             |

4.2.3 Ablation Study. To verify the impact of our designs, we study three ablated variants of our method and have the following findings: i) The performance of SHTAA is superior to CrossDQN, which verifies the effectiveness of all our designs. ii) The performance of SHTAA is superior to SHTAA (w/o UA-Sim), which mainly verifies the effectiveness of uncertainty-aware MDP similarity. iii) The performance gap between SHTAA and SHTAA (w/o AC) indicates the effectiveness of action constraint. iv) The performance gap between SHTAA and SHTAA (w/o Loss$_{TL}$) indicates the guiding significance of agent$_S$.

4.2.4 Hyperparameter Analysis. We analyze the sensitivity of these two hyperparameters: $N$ and $\tau$. The optimal hyperparameter values are obtained by grid search and we have the following findings: i) $N$ is the step number of NSR. Due to the discount rate $\gamma$, the rewards sampled at larger time steps must contribute less to the NSR. Therefore, if an appropriate step number $N$ is chosen, the corresponding NSR may contain most of the most relevant information from the local environmental dynamics. ii) $\tau$ is the weight threshold for instance transfer. A smaller $\tau$ would lead to more negative transfer while a larger $\tau$ would lead to less efficient transfer learning.

4.3 Online Results
We deploy the agent trained by SHTAA on Meituan food delivery platform’s data-poor entrance through online A/B test. The ads revenue and service fees increase by 3.72% and 3.31%, which demonstrates that our method greatly increases the platform revenue.

5 CONCLUSIONS
In this paper, we propose Similarity-based Hybrid Transfer for Ads Allocation (SHTAA) to effectively transfer the samples and knowledge from data-rich source entrance to other data-poor entrance. Both offline experiments and online A/B test have demonstrated the superior performance and efficiency of our method.

ACKNOWLEDGMENTS
We thank the anonymous reviewers for their suggestions and comments. We also thank Fan Yang, Hui Niu for helpful discussions.
REFERENCES

[1] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. 2018. Distributional reinforcement learning with quantile regression. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32.

[2] Chao Du, Zhifeng Gao, Shuo Yuan, Lining Gao, Ziyan Li, Yifan Zeng, Xiaoliang Zhu, Jian Xu, Kun Gai, and Kuang-Chih Lee. 2021. Exploration in Online Advertising Systems with Deep Uncertainty-Aware Learning. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2792–2801.

[3] Jun Feng, H. Li, Minjie Huang, Shichen Liu, Wenwu Ou, Zhirong Wang, and Xiaoyan Zhu. 2018. Learning to Collaborate: Multi-Scenario Ranking via Multi-Agent Reinforcement Learning. Proceedings of the 2018 World Wide Web Conference (2018).

[4] A. Ghose and Sha Yang. 2009. An Empirical Analysis of Search Engine Advertising: Sponsored Search in Electronic Markets. Manag. Sci. 55 (2009), 1605–1622.

[5] Anastasios Giannopoulos, Sotirios Spantidelas, Nikolaos Kapsalis, Panagiotis Karkazis, and Panagiotis Trakadas. 2021. Deep reinforcement learning for energy-efficient multi-channel transmissions in 5G cognitive Hetnets: Centralized, decentralized and transfer learning based solutions. IEEE Access 9 (2021), 129358–129374.

[6] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

[7] Xiang Li, Chao Wang, Bin Tong, Jiwei Tan, Xiaoyi Zeng, and Tao Zhuang. 2020. DeepTime-Aware Item Evolution Network for Click-Through Rate Prediction. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 785–794.

[8] Guogang Liao, Ze Wang, Xiaoxu Wu, Xiaowen Shi, Chuheng Zhang, Yongkang Wang, Xinxing Wang, and Dong Wang. 2021. Cross DQN: Cross Deep Q Network for Ads Allocation in Feed. arXiv preprint arXiv:2109.04353 (2021).

[9] Ruobing Xie, Shaoliang Zhang, Bui Wang, Fong Xia, and Leyu Lin. 2021. Hierarchical Reinforcement Learning for Integrated Recommendation. In Proceedings of AAAI.

[10] Carl Edward Rasmussen. 2003. Gaussian processes in machine learning. In Summer school on machine learning. Springer, 63–71.

[11] Hugh Salimbeni and Marc Deisenroth. 2017. Doubly stochastic variational inference for deep Gaussian processes. Advances in neural information processing systems 30 (2017).

[12] Yuguang Yan, Zhiyuan Xu, Bironjdh Tiwana, and Shaunak Chatterjee. 2020. Ads Allocation in Feed via Constrained Optimization. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 3386–3394.

[13] Xiangyu Zhao, Changsheng Gu, Haoshenlan Zhang, Xiawang Yang, Xiaobing Liu, Hui Liu, and Jiliang Tang. 2021. Jointly learning to recommend and advertise. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 3319–3327.

[14] Zhuangdi Zhu, Kaiming Lin, and Jiayu Zhou. 2020. Transfer learning in deep reinforcement learning: A survey. arXiv preprint arXiv:2009.07888 (2020).