KRAF: A Flexible Advertising Framework using Knowledge Graph-Enriched Multi-Agent Reinforcement Learning

Jose A. Ayala-Romero
Huawei Ireland
Research Center
Dublin, Ireland
jayalaromero@gmail.com

Péter Mernyei
Huawei Ireland
Research Center
Dublin, Ireland
pmernyei@gmail.com

Bichen Shi
Huawei Ireland
Research Center
Dublin, Ireland
bichen.shi@huawei-partners.com

Diego Mazón
Huawei Ireland
Research Center
Dublin, Ireland
dg.mazon@gmail.com

ABSTRACT

Bidding optimization is one of the most important problems in online advertising. Auto-bidding tools are designed to address this problem and are offered by most advertising platforms for advertisers to allocate their budgets. In this work, we present a **K**nowledge Graph-enriched Multi-Agent **R**einforcement Learning **A**dvertising **F**ramework (KRAF). It combines Knowledge Graph (KG) techniques with a Multi-Agent Reinforcement Learning (MARL) algorithm for bidding optimization with the goal of maximizing advertisers' return on ad spend (ROAS) and user-ad interactions, which correlates to the ad platform revenue. In addition, this proposal is flexible enough to support different levels of user privacy and the advent of new advertising markets with more heterogeneous data. In contrast to most of the current advertising platforms that are based on click-through rate models using a fixed input format and rely on user tracking, KRAF integrates the heterogeneous available data (e.g., contextual features, interest-based attributes, information about ads) as graph nodes to generate their dense representation (embeddings). Then, our MARL algorithm leverages the embeddings of the entities to learn efficient budget allocation strategies. To that end, we propose a novel coordination strategy based on a mean-field style to coordinate the learning agents and avoid the curse of dimensionality when the number of agents grows. Our proposal is evaluated on three real-world datasets to assess its performance and the contribution of each of its components, outperforming several baseline methods in terms of ROAS and number of ad clicks.

CCS CONCEPTS

• **Information systems** → **Computational advertising**.

KEYWORDS

Online Advertising, Bid Optimization, Multi-Agent Reinforcement Learning, Knowledge Graph

ACM Reference Format:
Jose A. Ayala-Romero, Péter Mernyei, Bichen Shi, and Diego Mazón. 2022. KRAF: A Flexible Advertising Framework using Knowledge Graph-Enriched Multi-Agent Reinforcement Learning. 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, 10 pages. https://doi.org/10.1145/3511808.3557373

1 INTRODUCTION

Online advertising has been gaining more importance in modern advertising markets during the last decade. It allows advertisers to increase the exposure of their products, improve user targeting and increase the platform revenue. Its market is increasing every year generating hundreds of billions of dollars per year [24]. Moreover, the online advertising industry is changing at a rapid pace due to the new privacy regulations and the advent of new advertising markets. This implies that the access to user data will be more and more restricted (e.g., 3rd party cookies are expected to eventually disappear [2]) limiting user profiling and targeting. Additionally, new advertising markets could bring in additional information and challenges to the advertising platform, such as in Autonomous Advertising, where companies offer free rides in autonomous cars in exchange for showing ads during the ride [7, 11], the geolocation and the route should be taken into consideration for ads servicing. For these reasons, the new generation of advertising platforms should adapt to heterogeneous data and market-dependent information. However, most of the current advertising platforms rely on click-through rate (CTR) models that use sequential user-ad interactions of a fixed format as inputs and rely on user tracking.

Furthermore, advertisers need to allocate their budget over time by sequentially selecting a bid value for each ad impression slot. While other works formulate the problem for a single bidder considering the market price stationary [3, 6, 35, 42], the auction mechanism is actually a multi-agent system by nature. That is, the outcome of each advertiser is highly dependent on the actions of all the involved bidders. This renders a complex coupled problem in which the change in the bidding strategy of one of the agents will affect others' strategies and vice versa. Finally, some additional aspects make this problem very challenging to solve. First, reward signals based on user response (e.g., click, conversion) are very sparse, preventing learning approaches to find efficient solutions. Second, the budget of the advertisers is usually limited and it is desirable to spend it throughout the whole day. Running out of budgets very early or having an unspent budget at the end of the period are not desirable situations and are indicating an ineffective use of the budget. Third, each advertiser has a different budget and performance. Prioritizing some of them can increase the benefit of the platform in the short term (e.g., advertisers with higher CTR

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557373

or budget), but it can be disadvantageous in the long term as the advertisers may decide to stop using the advertising platform due to their poor performance.

To overcome these challenges, we propose a **K**nowledge Graph-enriched Multi-Agent **R**einforcement Learning **A**dvertising **F**ramework (KRAF). This novel advertising framework can efficiently use heterogeneous data of different natures and adapt to diverse privacy settings. We tackle the problem of the coordination among advertisers by proposing a multi-agent learning algorithm with a novel coordination mechanism. Besides, we address the sparse system feedback and the constrained budget allocation problems by learning farsighted learning strategies using shaped reward signals.

Specifically, we integrate all the available information such as past user-ad interactions (if available), information about the user or ad slot (e.g., contextual information, interest groups from Interest-Based Advertising, etc), and information about ads (e.g., type of product, brand, etc.) into a Knowledge Graph (KG), e.g., Freebase [1], Microsoft Satori [25], etc. Based on that, we learn a dense representation of users and ads (embeddings) as well as a user-ad affinity metric and use them for decision-making. For that purpose, we apply a hybrid approach that combines Knowledge Graph Embedding (KGE) models and graph convolution networks (GCN) [33]. The KGE models parameterize entities and relations as vector representations preserving the structural properties of the graph and the GCN propagates recursively the information of the nodes. Hence, the embedding of each node is updated based on the embeddings of its neighbors, thus capturing high-order connectivity in the graph.

Then, we rely on Multi-Agent Reinforcement Learning (MARL) to learn farsighted bidding strategies for advertisers. We propose a novel coordination strategy for our MARL solution based on a mean-field style, using custom coordination signals designed specifically for this problem. The coordination strategy also avoids the curse of dimensionality with the growth of the number of agents. In order to avoid the problem of sparsity in the rewards signal, we use a shaped reward signal to control the spent budget during the day providing fairness among advertisers.

As a result, we propose a very flexible advertising framework that efficiently allocates the advertisers' budget maximizing their ROAS and taking into account the total platform revenue. The novel combination of KG techniques and a custom MARL algorithm allows KRAF to adapt to several advertising settings with different data availability and user privacy.

2 RELATED WORK

Learning user preferences using KG. KG techniques have been recently incorporated into recommender systems to improve their performance [10]. The strength of KG techniques relies on its capacity to agglutinate heterogeneous information from different domains. In the context of making recommendations, KGs can include feedback from users (ratings, clicks, dwell, times, etc.), content information of the items (e.g., item attributes like brands or categories the item belongs to), and user side information (e.g., device type). KG-based recommendation systems can be categorized into path-based and embedding methods, depending on how they leverage KGs. In path-based methods [27, 29, 44], long-range connectivity is used to connect the item and the target user via KG entities. The user preference is predicted based on these paths using different techniques. For example, the authors in [29] enhance the user representation by memorizing the item representation along with the user-item paths. Nevertheless, the recommendation accuracy with these methods is highly dependent on the quality of paths, which are hard to design. In embedding-based methods [12, 30, 41], a Knowledge Graph Embedding (KGE) algorithm is used to extract the embedding of each entity and relation in the graph. Then, the entity embeddings are used to better represent items for recommendation. For example, the authors in [30] propose a deep end-to-end framework that learns high-order interactions between items using knowledge graph embedding and cross & compress units. Other works apply graph neural networks in order to exploit the KG structure. Thus, they distill embeddings based on the node's neighborhood information and effectively apply them for recommendation. For example, the authors in [33] combine KGE with graph attention networks in order to model the high-order connectivity in an end-to-end fashion. However, all of these methods rely essentially on one-step prediction tasks based on instantaneous feedback, neglecting the long-term utility.

To overcome this issue, other works combine KG techniques with reinforcement learning (RL) in the context of recommender systems [4, 32, 46]. For instance, in [46] a graph convolutional network is used to learn the state representation to enhance the performance of an RL recommendation policy.

In contrast to previous works, we face a more challenging problem with a twofold objective: Firstly, similar to recommender systems, we aim to deliver high-quality ads matching users' interests. Secondly, we aim to find an optimal policy to allocate the budget of all advertisers over time to maximize their ROAS. In this context, we combine KG techniques with MARL algorithms to address this double objective. To the best of our knowledge, this is the first time that KG techniques are used in advertising.

Bid optimization. Bid optimization is one of the most studied problems in advertising. The goal is to optimize the bid value of advertisers aiming at maximizing some KPIs under budget constraints [31]. Some works formulate it as an optimization problem [23, 42]. The authors in [42] model the optimal bidding strategy as a non-linear function of the predicted CTR (pCTR). However, these methods are not able to incorporate budget constraints.

Other works propose more sophisticated bidding strategies that rely on reinforcement learning (RL). The authors in [3] model the problem as a Markov Decision Process (MDP) and learn sequentially how to allocate the budget. In [35], deep RL is used to optimize the bidding strategy based on a high-level semantic information state. All these works perform bid optimization of one single agent (advertiser), while the competitors are considered as a part of the environment, i.e., the market price is considered stationary. However, the single-agent RL formulation neglects that the auction mechanism is a multi-agent system by nature as the outcome of each of the agents depends on all the involved bidding agents.

To alleviate this problem, some works formulate the problem as a Markov Game and propose MARL strategies for bid optimization [5, 16, 37, 38]. The authors in [16] group advertisers into clusters and apply the Deep Deterministic Policy Gradient algorithm [19] at the cluster level. This approach is extended in [5], where the authors assume partially observable opponents. They aggregate the competitors in each auction as a virtual agent and introduce an opponent model for market price prediction. Finally, the authors in [37] propose a mixed cooperative-competitive framework to trade-off between advertisers' utility and platform revenue. To that end, they propose a reward credit assignment scheme to balance cooperation and competition. In all these works it is assumed that either the affinity between the user and the ad (e.g., pCTR) or the value of the ad is given.

In contrast, we propose a framework that uses additional information about ads (e.g., type of product or service, utility, etc.), user information (if available), and contextual information (related to the ad slot). All this information allows us to exploit potential higher-level links between users and ads that are taken into account for budget allocation decision-making. Moreover, our proposed framework provides a high level of flexibility allowing us to integrate any type of relevant information into the bidding process. This favors its integration in new advertising markets such as autonomous advertising [7]. Finally, our framework can easily integrate user privacy methods such as Interest-based advertising (IBA) [9].

3 PROBLEM FORMULATION

3.1 Unified Knowledge graph

User-ad interaction bipartite graph. The interactions between users and ads (e.g., click, purchase, visualization) are usually recorded in the advertising platforms. This information can be encoded into a user-ad bipartite interaction graph $\mathcal{G}_j$ composed of entity-interaction-entity defined as $\{(u, \mu_j, z)\,|u \in \mathcal{U}, \mu_j \in \mathcal{M}_j, z \in \mathcal{Z}\}$, where $\mathcal{U}$ is the set of users, $\mathcal{M}_j$ is the set of available interactions (e.g., click, purchase), and $\mathcal{Z}$ is the set of ads.

Knowledge Graph. Besides the interactions between users and ads, some additional information about these entities can be available. We can have ad attributes such as "smartphone", "Huawei (brand)", or "5G connectivity". We can also use information about the user via, for instance, contextual advertising. For example, if a user is reading an article about the performance of LeBron James in his last game in a digital sports newspaper, we can extract attributes such as "sport", "basketball", and "NBA". These attributes are related to the preferences of the user. Considering IBA, the attributes represent the user's topics of interest.

We assume that these attributes have an associated entity in a Knowledge Graph (KG), e.g, Freebase [1], Microsoft Satori [25]. We define the KB $\mathcal{G}_k$ as a set of triples subject-property-object. Formally, $\mathcal{G}_k = \{(h, \mu_k, \omega)\,|h, \omega \in \mathcal{E}_k, \mu_k \in \mathcal{M}_k\}$, where $\mathcal{E}_k$ is the set of entities and $\mathcal{M}_k$ is the set of possible relations from the head entity h to the tail entity ω. KGs encode objects, events, situations or abstract concepts structured as a graph. An example of a KG triple is $\{h =$ "Albert Einstein", $\mu_k =$ "born in", $\omega =$ "Germany"$\}$.

Unified Knowledge Graph (UKG). Assuming that every attribute has a corresponding entity in the KG, we can merge the user-ad interaction bipartite graph, its attributes, and the Knowledge Graph into a Unified Graph $\mathcal{G} = \{(h, \mu, \omega)\,|h, \omega \in \mathcal{E}, \mu \in \mathcal{M}\}$, where $\mathcal{E} = \mathcal{E}_k \cup \mathcal{U} \cup \mathcal{Z}$, and $\mathcal{M} = \mathcal{M}_k \cup \mathcal{M}_j$. By using the Unified Knowledge Graph we can exploit high-order connectivity among nodes (Fig. 1). For example, if user1 and user2 clicked on ad1 and user2 also clicked on ad2, we can think that user1 may like ad2 (similarity between users). This can be represented in the following graph: $u_1 \xrightarrow{\mu_{click}} z_1 \xrightarrow{-\mu_{click}} u_2 \xrightarrow{\mu_{click}} z_2$, in which it is shown a multi-hop relation of length 3 between u_1 and z_2. This is similar to collaborative filtering (CF) methods, which recommend similar items to users with similar behavior. However, the example above only considers the interaction bipartite graph and we can go further by considering the UKG. For instance, two ads that are different (e.g., smartphone ad and smartwatch ad) can be connected through the UKG. Both ads can be connected with entities such as "gadget", "technology", and brand. This reveals that these ads can be a good match for users with an affinity for technology, even when previous records of these ads are not available. Similarly, we can find affinity among users by, for example, using their contextual information. Let us assume that we extract that two users have affinities for "tennis" and "badminton", respectively, based on their contextual information. These attributes will be connected in the UKG since they are "sports", and more specifically "racket sports". Thus, a high-order connection will be established between them. In contrast, CF and other supervised learning methods do not exploit these high-order relations.

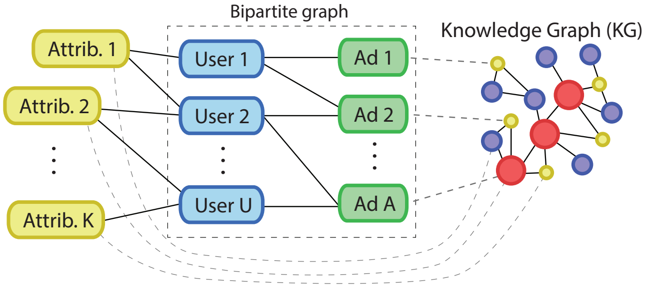

Figure 1: Unified Knowledge Graph. The available ads are represented as graph entities in green. The graph entities in blue represent the users that interacted with these ads, but they can also represent ad slots or user groups, depending on the setting. The entities in yellow represent attributes of the blue entities. For example, if the blue entities are ad slots the attributes can be contextual features. If considering IBA, the attributes can be the users' topics.

Embedding representation and affinity. Our final goal is to allocate the budget of the advertisers. To this end, we need to distill dense vector representations of users and ads. This is fundamental in order to build the state of the decision-making algorithm. The latent representation should encode not only the user-ad interactions but also the higher-order relation to better characterize the entities in the graph. Different techniques can be used to distill the structural and relational knowledge of the graph into a dense latent representation or embeddings, which we described in Section 4.1. Moreover, based on these embeddings, we also predict the interest of users in ads by learning the affinity $\delta_{u,z}$ between user u and ad z.

3.2 Budget Allocation as a Markov Game

We formulate the budget allocation in the sequential auctions as a stochastic game given by $\Gamma =< N, \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma >$. The number of players or agents is given by N, γ is the discount factor, and $\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}$ are the sets of states, joint actions, reward functions and transition probability functions, respectively. At each time t, the state observation of each agent i is denoted by $o_t^i \in \mathcal{S}^i$, and the system state is $\mathbf{s}_t = (o_t^1 \dots o_t^N) \in \mathcal{S}$. The joint action of all agents at time t is denoted by $\mathbf{a}_t = (a_t^1 \dots a_t^N) \in \mathcal{A}$. Each agent selects the actions based on a policy $\pi^i : \mathcal{S}^i \mapsto \mathcal{A}^i$ and the join policy is defined as $\pi = (\pi^1 \dots \pi^N)$. When the agent i selects the action a_t^i, it transits to the next state $o_{t+1}^i \sim \mathcal{P}_t^i(\mathbf{s}_t, \mathbf{a}_t)$, where $\mathcal{P}_t^i \in \mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$. Then, the agent receives the reward $r_{i,t} \sim \mathcal{R}_t^i(\mathbf{s}_t, \mathbf{a}_t)$, where $\mathcal{R}_t^i \in \mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$. Note that both the transition probabilities and the reward functions depend on the system state $\mathbf{s}_t$ and the joint action $\mathbf{a}_t$.

The value function of a certain joint policy π for the agent i is given by

$$v_\pi^i(\mathbf{s}) = \sum_{t=0}^{T} \gamma^t \mathbb{E}_{\pi,\mathcal{P}} \left[r_{i,t}(\mathbf{s}_t, \mathbf{a}_t) | \mathbf{s}_0 = \mathbf{s}, \pi \right], \tag{1}$$

where $\gamma \in [0, 1]$ is the reward discount factor and T indicates the number of auctions or bid requests in one day. Note that the value T may change depending on the day.

The objective of the agents is to find the optimal policy π_* that maximizes their value function shown in eq. (1). Note that each player has an independent reward function that depends on the behavior of other agents. Therefore, the definition of the optimal policy is not clear in this setting as the maximization of the value function for one user can harm the value function for others, making the game non-stationary. To solve this problem we propose an equilibrium-based approach [40]. That is, we aim at finding the Nash equilibrium (NE) optimal policy π_*. When the joint policy converges to π_* no agent can improve its value function while other agents keep their policies unchanged, that is,

$$v^i(s; \pi_*) \geq v^i(s; \pi^i, \pi_*^{-i}), \tag{2}$$

where π_*^{-i} denotes the joint policy of all the agents except agent i, i.e., $\pi_*^{-i} := [\pi_*^1, \dots, \pi_*^{i-1}, \pi_*^{i+1}, \dots, \pi_*^N]$.

4 KG-ENHANCED MARL APPROACH

In this section, we detail our proposal, KRAF, whose dataflow is depicted in Fig. 2. In a nutshell, the available information about ad slots[1] is sent to the UKG. Then, the embeddings of the ad slot and potential ads (both included in the UKG) are sent to the MARL algorithm. Each learning agent selects a bidding value on behalf of each advertiser and the ad with the highest bid is delivered. Finally, based on the result of the auction and user behavior, a feedback signal is provided to the MARL algorithm to enable learning.

4.1 UKG-Enriched State Representation

In this section, we detail how the embeddings of the UKG are generated and how they are used to predict affinity between users and ads. We use the Knowledge Graph Attention Network (KGAT) model [33], a hybrid approach combining Knowledge Graph Embedding (KGE) [20] models and graph convolution networks (GCN) [17]. KGE models parameterize entities and relations as vector representations preserving the structural properties of the graph. Additionally, the GCN propagates recursively the information of the nodes. The embedding of each node is updated based on the embeddings of its neighbors, thus capturing high-order connectivity. This approach comprises three layers: embedding layer, GCN propagation layer, and prediction layer.

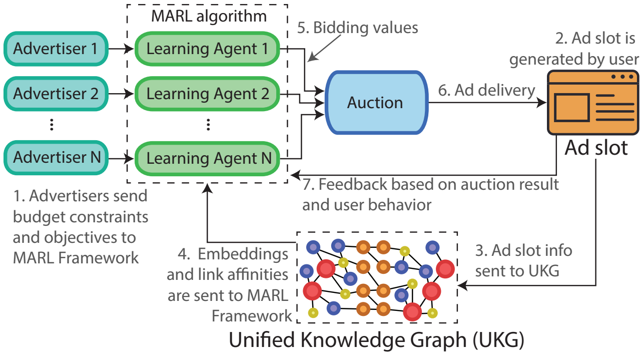
MARL algorithm
Advertiser 1
Advertiser 2
⋮
Advertiser N
Learning Agent 1
Learning Agent 2
⋮
Learning Agent N
5. Bidding values
2. Ad slot is generated by user
Auction
6. Ad delivery
Ad slot
7. Feedback based on auction result and user behavior
1. Advertisers send budget constraints and objectives to MARL Framework
4. Embeddings and link affinities are sent to MARL Framework
3. Ad slot info sent to UKG
Unified Knowledge Graph (UKG)

Figure 2: KRAF architecture and dataflow.

4.1.1 *Embedding Layer.*

In order to compute the initial vector representations, we use TransR [20]. In contrast to other translation KGE models, TransR supports 1-to-N relations that are present in the bipartite graph. We define $e_h, e_\omega \in \mathbb{R}^d$ and $e_\mu \in \mathbb{R}^k$ as the embeddings of the entities h, ω, and μ, respectively. The score (or plausibility) of a triple (h, μ, ω) is given by

$$g(h, \mu, \omega) = ||W_\mu e_h + e_\mu - W_\mu e_\omega||_2^2, \tag{3}$$

where $W_\mu \in \mathbb{R}^{k \times d}$ is the projection matrix of relation μ. The training of TransR encourages valid triples against false ones by using pairwise ranking loss:

$$\mathcal{L}_{\text{KGE}} = \sum_{(h,\mu,\omega,\omega') \in \mathcal{T}} - \ln \sigma \left(g(h, \mu, \omega') - g(h, \mu, \omega) \right), \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function and $\mathcal{T} = \{(h, \mu, \omega, \omega') | (h, \mu, \omega) \in \mathcal{G}, (h, \mu, \omega') \notin \mathcal{G}\}$, that is, (h, μ, ω') are false triples.

4.1.2 *GCN Propagation Layer.*

The embedding propagation enabling high-order connectivity is performed using a GCN architecture [17] enhanced with an attention mechanism [28]. We detail now the architecture of a single layer that can be stacked to build a multiple-layer network. We define $\mathcal{D}_h = \{(h, \mu, \omega) | (h, \mu, \omega) \in \mathcal{G}\}$ as the set of all the triples where h is the head entity. The embedding of the entity h based on first-order connectivity is given by

$$e_{\mathcal{D}_h} = \sum_{(h,\mu,\omega) \in \mathcal{D}_h} \tilde{\alpha}(h, \mu, \omega) e_\omega, \tag{5}$$

where $\tilde{\alpha}(h, \mu, \omega)$ is the normalized attention weight, indicating the amount of information transmitted to h from each of its neighbors. The attention weight is defined as [33]:

$$\alpha(h, \mu, \omega) = (W_\mu e_\omega)^\top \tanh \left(W_\mu e_h + e_\mu \right). \tag{6}$$

Note that this attention mechanism measures the affinity between the translated h and ω in the space of the relation μ, and the tanh is a non-linear activation function that increases the representation ability of the model. We normalize the attention weights across all the triples in $\mathcal{D}_h$ using a softmax function:

$$\tilde{\alpha}(h, \mu, \omega) = \frac{\exp(\alpha(h, \mu, \omega))}{\sum_{(h,\mu',\omega') \in \mathcal{D}_h} \exp(\alpha(h, \mu', \omega'))}. \tag{7}$$

[1] Note that we use the terms "user" and "ad slot" interchangeably as users generate the ad slots but user information (e.g., past user-ad interactions) may not be available depending on the privacy setting.

Finally, for a node h, we aggregate the embedding e_h computed by the KGE model in the first place and $e_{\mathcal{D}_h}$, which considers the information propagation of the first-order connectivity and indicates how to weight the information from each neighbor (attention mechanism). The new representation of the entity h is given by $e_h^{(1)} = f(e_h + e_{\mathcal{D}_h})$. Specifically, the information aggregation function is given by [33]:

$$f(e_h, e_{\mathcal{D}_h}) = \text{LeakyReLU}\left(W_1(e_h + e_{\mathcal{D}_h})\right) + \text{LeakyReLU}\left(W_2(e_h \odot e_{\mathcal{D}_h})\right), \tag{8}$$

where $W_1, W_2 \in \mathbb{R}^{d' \times d}$ are trainable weights used to obtain useful information to propagate, d' is the transformation size, and $\odot$ indicates element-wise product.

4.1.3 Multi-Layer Network. By stacking several layers, we can build a multi-layer network to propagate the information a number of steps forward. Specifically, the representation of the entity h at layer l is

$$e_h^{(l)} = f(e_h^{(l-1)} + e_{\mathcal{D}_h}^{(l-1)}), \tag{9}$$

where

$$e_{\mathcal{D}_h}^{(l-1)} = \sum_{(h,\mu,\omega) \in \mathcal{D}_h} \tilde{\alpha}(h, \mu, \omega) e_\omega^{(l-1)}. \tag{10}$$

Note that we consider $e_h^{(0)}$ as the initial vector representation given by the embedding layer, i.e., e_h.

4.1.4 User-ad affinity prediction layer. Considering a multi-layer network with L layers, we obtain a representation of each user and ad at each layer. We consider a layer aggregation mechanism [39] in order to concatenate the representation of an entity at each of the L layers. For instance, the representation of a user u is computed as

$$e_u^* = \text{concat}\left\{e_u^0, e_u^1, \dots, e_u^L\right\}. \tag{11}$$

Finally, the affinity between a user u and an ad z is given by the inner product of their representations, i.e.,

$$\delta_{u,z} = e_u^{*\top} e_z^*, \tag{12}$$

where e_z^* is the layer aggregation representation of the ad z computed as in eq. (11).

4.1.5 Training mechanism. In order to train the multi-layer network, we rely on the BPR loss, defined as [26]:

$$\mathcal{L}_{\text{GCN}} = \sum_{(u,z,z') \in O} -\ln \sigma\left(\delta_{u,z} - \delta_{u,z'}\right), \tag{13}$$

where $O = \left\{(u, z, z') | (u, z) \in \mathcal{I}^+, (u, z') \in \mathcal{I}^-\right\}$ is the training set, $\mathcal{I}^+$ is the set of positive user-ad interactions, and $\mathcal{I}^-$ is the set of negative or unobserved interactions. Finally, we define the global loss as follows [33]:

$$\mathcal{L} = \mathcal{L}_{\text{KGE}} + \mathcal{L}_{\text{GCN}} + \lambda ||\theta||_2^2, \tag{14}$$

where $\theta = \left\{E, W_\mu \ \forall \mu \in \mathcal{M}, W_1^{(l)}, W_2^{(l)} \ \forall l \in \{1, \dots, L\}\right\}$ is the set of model parameters, E is the embedding matrix of all the entities in the graph, and λ is the L2 regularization parameter.

The embedding representations of users and ads computed in eq. (11) and the affinity obtained in eq. (12) are used to define the state of the MARL algorithm as described in the next section. Note that both embeddings and affinity should be computed before training the MARL algorithm by minimizing eq. (14).

4.2 Multi-agent Reinforcement Learning for budget allocation

In this section, we detail the design of the MARL algorithm integrated into the proposed advertising framework. Each learning agent makes decisions on behalf of each advertiser. All the agents receive the same bid request and have to select a bidding value. Next, we define states, actions, and the reward signal.

State. The observation of the state of agent i at time t is defined as $o_t^i = [b_t^i, \bar{b}_t^i, e_i^*, e_{u_t}^*, \delta_t^i]$, where b_t^i is the current budget of the agent, $\bar{b}_t^i$ indicates its expected budget, e_i^* is the embedding representation of the advertiser i, $e_{u_t}^*$ is the embedding representation of the user u_t at time t, and δ_t^i indicates (with a little abuse of the notation) the affinity between the advertiser i and the user u_t at time t. We assume that each advertiser only has one ad to deliver and therefore the index i corresponds to only one ad z. We use $\bar{b}_t^i$ as a guideline of the budget that the agent should desirably have at time t. We found experimentally that this input stabilizes the convergence avoiding non-stationary situations. Also, we prevent the agents from finding some undesirable equilibrium points, such as bidding very high at the beginning of the day and running out of budget very early, or the opposite, leading to suboptimal solutions. We define the expected budget of agent i at time t as

$$\bar{b}_t^i = B_0^i - \frac{(B_0^i - B_T^i)t}{T}, \tag{15}$$

where B_0^i is the initial budget of agent i and B_T^i indicates the desired final budget of the agent at the end of the day. Generally, we configure $B_T^i = 0$ as we intend the agent to spend its whole budget during the day. However, with $B_T^i > 0$ other equilibrium points can be found. Note that our learning agents not only use the observations o_t^i but also the coordination signals defined in Sec. 4.3. These coordination signals are used to share information among agents to allow convergence without increasing the computational complexity.

State transition. The expected budget $\bar{b}_t^i$ is updated according to eq. (15). The embeddings of ads and users as well as the affinity are given by the UKG module, and the current budget is updated according to the second price auction dynamics[2]. That is, when the agent i wins the auction $a_t^i > a_t^j \ \forall j \neq i$, its budget is updated as $b_{t+1}^i = b_t^i - MP$, where MP is the market price (the value of the second-highest bid). The budget of the rest of the bidders $b^j \ \forall j \neq i$ remains the same for the next auction at $t + 1$.

Action. We define the action for agent i at time t as $a_t^i \in [\beta_{\min}, \beta_{\max}]$, where $\beta_{\min}$ and $\beta_{\max}$ are the minimum and maximum allowable bids, respectively.

Reward. The design of the reward function is crucial to achieving convergence and finding optimal equilibrium points among agents. It should encode two (opposed) metrics: the ROAS of the advertisers and the total platform revenue. Moreover, our problem presents some difficulties to achieve convergence in the learning

[2] We focus on second price auctions as they are dominant in the advertising ecosystem. However, first price auctions can also be considered by using the highest bid as the market price.

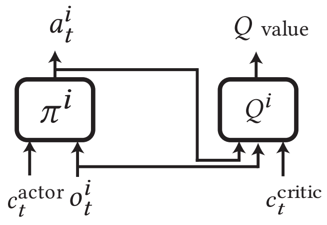

Figure 3: Actor-critic architecture of learning agent i including coordination signals.

phase. First, the metrics of interest for advertisers (e.g., click or purchase) are usually very sparse. Second, allocating the budget throughout the day (T decision periods) is a difficult task to learn due to the usually very large value of T.

To alleviate these issues we introduce the concept of *reward shaping* [14]. The main idea is to replace the sparse reward signal with a dense reward signal incorporating some domain knowledge in a supervised way. Hence, when the agents succeed in maximizing the dense reward signal, they are also obtaining useful knowledge and getting closer to their end goal. The reward for agent i at time t is given by two terms:

$$r_{i,t} = r_{i,t}^{\text{auct}} + \beta \cdot r_{i,t}^{\text{budget}}, \tag{16}$$

where β is a parameter weighting the importance of each term. The first term $r_{i,t}^{\text{auct}}$ only takes a positive value for the winning agent:

$$r_{i,t}^{\text{auct}} = \begin{cases} r_{i,t}^{\text{bid}}, & a_t^i > a_t^j \ \ \forall j \neq i \\ 0, & \text{otherwise,} \end{cases} \tag{17}$$

where $r_{i,t}^{\text{bids}}$ return a high reward when there is an interaction of the interest for the advertiser and otherwise returns a value based on the affinity in order to mitigate the effect of sparsity, i.e.:

$$r_{i,t}^{\text{bid}} = \begin{cases} 1, & \text{if user-ad interaction from } \mathcal{M}^i \\ \mathcal{F}(\delta_t^i), & \text{otherwise} \end{cases} \tag{18}$$

where $\mathcal{M}^i \subseteq \mathcal{M}_j$ is the set interactions (e.g., click, visualization, purchase) that the agent i wants to maximize and $\mathcal{F}(\cdot) \mapsto [0,1]$ is a function that incentives the winning of users with high affinity. Specifically, we consider $\mathcal{F}(\delta^i) = \eta^{\delta^i_{\text{rank}}}$, where $\delta^i_{\text{rank}} \in [0, \ldots N-1]$ indicates the ranking of user i based on the affinity. For example, when the user i has the highest affinity $\delta^i_{\text{rank}} = 0$; if it has lowest affinity $\delta^i_{\text{rank}} = N-1$.

The goal of the second term, $r_{i,t}^{\text{budget}}$, is to make the agents spend the full budget throughout the day but not run out of it too early. Based on the shaped reward principle, we distribute this penalty along the day (instead of penalizing the agent at the end of each epoch), encouraging agents to keep their current budget b_t^i close to their respective expected budget $\bar{b}_t^i$, that is,

$$r_{i,t}^{\text{budget}} = -\left|b_t^i - \bar{b}_t^i\right|. \tag{19}$$

4.3 Large-Scale Multi-Agent System Design

Most MARL methods are limited typically to a small number of agents. When the number of learning agents is very large, the learning process becomes intractable as the computational complexity grows exponentially with the number of agents (curse of dimensionality).

As an example, the well-known MADDPG algorithm uses an actor-critic architecture based on the principle of *centralized learning and decentralized execution* [21]. Thus, the actor (or policy as defined in Sec. 3.2) selects an action based on local observations, i.e., $a_t^i = \pi^i(o_t^i)$. Then, the critic approximates the Q-value for a given action-state pair, i.e., $Q(\mathbf{s}_t, \mathbf{a}_t)$. Note that the input of the critic is the system state and the joint action (all agents). Therefore, this technique can only be applied with a fixed number of agents (which may vary in our setting) and with limited scalability.

There are a few works addressing the scalability problem by including ideas from Mean Field Theory [5, 40]. In these works, the interactions among the population of agents are approximated by those between a single agent and the average effect from the overall population.

However, some of the assumptions made in these previous works do not hold in our problem. First, they consider discrete action spaces, while in our case the bids are continuous and their discretization may limit the expressivity of our solution. Second, the average effect of the population (e.g., average bid value, average affinity) is not very informative in our case, hindering the coordination among learning agents and convergence. Based on these ideas, we design two coordination signals based on domain knowledge of this setting. We define the critic and actor coordination signals as:

$$c_t^{\text{critic}} = \left(\delta_t^{\max}, \delta_t^*, a_t^{\max}, MP\right), \quad c_t^{\text{actor}} = \left(\delta_t^{\max}\right), \tag{20}$$

where $\delta_t^{\max}$ is the maximum affinity in the auction time t, δ_t^* is the affinity of the winner of the auction at time t, and $a_t^{\max}$ is the maximum bid. Note that all the values in c_t^{critic} (except $\delta_t^{\max}$) can only be computed after the agents select an action, making this signal only suitable for training. Conversely, c_t^{actor} only depends on the state of the game $\mathbf{s}_t$ and therefore can be used to feed the actor. These coordination signals encode the state of the game more effectively than the average state or action of the population, avoiding non-stationarity and enabling convergence.

Let us formally define the actor and critic functions of our solution. We define the actor function as $\pi^i(o_t^i, c_t^{\text{actor}}) : \mathcal{S}^i \times C^{\text{actor}} \mapsto \mathcal{A}^i$, where C^{actor} is the set of possible values of the actor coordination signal. The critic function, which returns the Q value, is given by $Q^i(o_t^i, a_t^i, c_t^{\text{critic}}) : \mathcal{S}^i \times \mathcal{A}^i \times C^{\text{critic}} \mapsto \mathbb{R}$, where C^{critic} is the set of possible values of the critic coordination signal. Fig. 3 depicts the architecture of a learning agent. Note that the dimensionality of the critic's input for a standard MARL solution such as [21] is $dim\{o^i\} \cdot N$. The dependency with N has serious implications on the computational complexity of the function approximator of the critic. In contrast, the dimensionality of the critic's input of our proposal is $dim\{o^i\} + dim\{C^{\text{critic}}\}$, which is independent of N.

Finally, both actor and critic are shared across all learning agents. This is feasible because all specific information about the agent (e.g., current budget) is encoded in the state. This strategy is used in previous works speeding up learning and reducing the amount of data needed for convergence [18, 40]. The details of the implementation of the actor and critic as well as the training process are explained in Sec. 5.2.

5 EVALUATION

5.1 Datasets and metrics

For evaluation, there are no publicly available advertising datasets that fit our purpose because they are anonymized and information about ads is not available (e.g., iPinYou [43], Taobao dataset [22]), preventing them to be linked with a KG. For that reason, we rely on recommender system datasets to evaluate our framework. We consider the items to recommend as ads and a like/dislike from a user to an item is considered as a click/no click from a user to an ad. We follow the setup in [33] on the following datasets:

- Amazon-book[3] is based on a large corpus of product reviews filtered specifically for books.
- Last-FM[4] is a music listening dataset collected between January and June 2015, where the items are artists, and edges represent that a user has listened to an artist.
- Yelp2018[5] is based on the Yelp challenge from 2018, where the items are restaurants and edges are based on user reviews.

All of these datasets have been filtered down to their 10-core, ensuring that all users and items have at least 10 connections. After this, they were each joined with a KG. In the case of Amazon-book and Last-FM, this is based on the KB4Rec dataset [45] connecting these datasets with the Freebase KG. For Yelp2018, the KG is instead based on metadata about local businesses such as locations, categories, and other attributes. Note that, due to the nature of these datasets, user attributes are not considered in our evaluation. Entities with fewer than 10 connections were also dropped from the KGs, as well as relations with fewer than 50 occurrences overall.

To train the embedding model, we use 70% of the known user-item interactions. After this, we filter for items with a high degree: at least 100 for Amazon-book, 150 for Yelp2018, and 200 for Last-FM. We split these items into groups of 25 for training and evaluating the MARL system. For each group of 25, we fetch all the users that interacted with any of these items and run a series of ad auctions. 90% of these groups are used for training the RL agents, with 10% reserved for evaluation. The evaluation is based on four metrics:

- **ROAS (Return On Ad Spend)** measures the revenue over the cost. We are interested in measuring the global ROAS (across all the advertisers) to compare different strategies. We consider that the revenue is equal to 1 when the user clicks and 0 otherwise. The cost is price paid by the advertisers to deliver the ads (the second highest bid) taken from $[\beta_{\min}, \beta_{\max}]$. Thus, the ROAS is computed as ROAS = (Total # of clicks)/(Total budget spent)
- **# clicks** is the accumulated number of click interactions for all the advertisers during one day.
- **Fairness** measures the similarity on the ROAS across advertises. We use the Jain's fairness index [15] across the individual ROAS of the advertisers, that is,

$$\text{Fairness} = \frac{\left(\sum_{i=1}^{N} \text{ROAS}_i\right)^2}{N \sum_{i=1}^{N} \text{ROAS}_i^2}, \tag{21}$$

where ROAS_i is the ROAS of advertiser i. Our objective is to measure inequalities among the advertisers as their ROAS can change depending on several factors such as the affinity and the initial budget. This fairness metric takes values between $1/N$ (the highest inequality) and 1 (same ROAS for all the advertisers)
- **Platform revenue** measures the total amount of budget spent by the advertisers on the platform. As this metric can vary a lot depending on the initial budget of the advertisers, we normalize dividing by the aggregated initial budget of all the advertisers.

5.2 Implementation Details

Auctions. Our framework computes the bidding value of all the agents and then a second price auction is performed. The advertiser with the highest bid pays the market price (second highest bid) and delivers the ad. As our approach computes the bidding value of all the agents, there is no need for external bidding information for the evaluation (e.g., from a dataset). We consider $N = 25$ advertisers and $T = 1000$ auctions per day. The bidding values are normalized so that $\beta_{\min} = 0$ and $\beta_{\max} = 1$ monetary units. The initial budget of the advertisers is drawn from a normal distribution with mean 14 and a standard deviation 5 monetary units. Thus, the advertisers have to allocate the budget in a smart way to avoid running out of budget before the end of each day. We consider clicks to be the interaction of interest for all the advertisers, i.e., $\mathcal{M}^i = \{\text{click}\}$ for $0 \leq i \leq N-1$.

UKG embeddings and affinity. To generate user and item embeddings from the UKG, we train the model defined in Sec. 4.1 with the hyperparameters suggested in [33]. This model is trained before the MARL algorithm to obtain the embeddings and affinity needed to define the MARL state. For real deployments, the model should be trained regularly in background to benefit from the new information added to the UKG. The embedding size is 64 and we consider three message-passing steps with 64, 32, and 16 dimensions, that concatenated according to eq. (11) form the final embeddings of 176 dimensions. Dropout is applied at each message-passing step at a rate of 0.1, and L2 regularization of 10^{-5} is used on the embeddings.

MARL algorithm. We design the learning agents based on the Twin Delayed Deep Deterministic Policy Gradient (TD3) [8] with an actor-critic architecture and off-policy learning. For each agent, both actor and critic have the same neural network (NN) architecture with 4 hidden layers, $(64, 128, 128, 64)$ number of units, respectively, and ReLU activation. The algorithm includes a twin critic and takes the smallest clipped value of the two critics. The actor training is delayed every two updates and some noise is added to the target actor for regularization. We set the learning rate to 10^{-4} and $\beta = 0.1$.

5.3 Performance evaluation

In this section, we evaluate the performance of our proposal and compare it against several baselines. We perform an ablation study to evaluate the contribution to the final performance of the two main components in KRAF: the UKG and the coordination strategy. In addition, we implement a reward strategy from the literature and a common strategy in advertising based on pCTR. In detail, we evaluate the following solutions:

- **Random policy**. Advertisers select a random bid in $[\beta_{\min}, \beta_{\max}]$ at each time t. This policy is used as baseline for comparison.
- **Affinity-based Policy (ABP)**. A common approach in online advertising is to select the bid for each advertiser proportional

[3] https://jmcauley.ucsd.edu/data/amazon/
[4] https://grouplens.org/datasets/hetrec-2011/
[5] https://www.yelp.com/dataset/

Dataset	Method	ROAS	# clicks	Fairness	Platf. rev.
Books	Random	0.086	29.5	0.672	**1**
	ABP	0.355	119.1	**0.898**	0.984
	TRCA$_{\tau=1}$	0.335	114.5	0.690	**1**
	TRCA$_{\tau=10^3}$	0.168	56.8	0.637	0.988
	KRAF$_{\text{-UKG}}$	0.186	62.5	0.569	0.980
	KRAF$_{\text{-coord}}$	0.415	141.7	0.763	0.999
	KRAF	**0.498**	**168.4**	0.857	0.999
Last-FM	Random	0.101	34.4	0.823	**1**
	ABP	0.444	148.5	0.941	0.980
	TRCA$_{\tau=1}$	0.573	195.3	0.896	0.998
	TRCA$_{\tau=10^3}$	0.206	70.2	0.769	0.999
	KRAF$_{\text{-UKG}}$	0.212	72.4	**0.977**	0.999
	KRAF$_{\text{-coord}}$	0.675	230.2	0.905	**1**
	KRAF	**0.808**	**275.8**	0.892	0.999
Yelp2018	Random	0.091	31.1	0.914	**1**
	ABP	0.214	72.6	0.750	0.993
	TRCA$_{\tau=1}$	0.340	114.1	0.912	0.982
	TRCA$_{\tau=10^3}$	0.242	81.7	0.935	0.987
	KRAF$_{\text{-UKG}}$	0.311	106.0	0.721	0.998
	KRAF$_{\text{-coord}}$	0.647	220.3	0.948	0.998
	KRAF	**0.690**	**235.4**	**0.985**	0.999

Table 1: Performance comparison.

to its pCTR. Using affinity as a proxy, the advertisers select a bid proportional to their affinity (eq. (12)).

- **TRCA** [37]. We adopt a temperature regularized credit assignment (TRCA) to distribute the reward among the agents. This strategy distributes the reward among the agents according to their contribution to the auction using a softmax function with temperature parameter τ. To integrate this strategy in KRAF, we distribute $r_{i,t}^{\text{bid}}$ (eq. (18)) among all the agents based on TRCA (see eq. (4) in [37]). Then, the second term in eq. (16) is added. For a fair comparison, the UKG and coordination signals are also used.
- **KRAF$_{\text{-UKG}}$**. We disable the use of the UKG in our proposal, using only the bipartite graph to compute the embeddings and the link prediction. For that purpose, we compute the embeddings using Neural Graph Collaborative Filtering (NGCF) [34]. Our goal is to evaluate the impact of KG on the final performance.
- **KRAF$_{\text{-coord}}$**. We disable the coordination signals for both the actor and the critic (i.e., c_t^{actor} and c_t^{critic}). The objective is to evaluate their the impact on the final performance.

Table 1 shows the performance comparison results where KRAF consistently obtains the best performance in terms of ROAS and number of clicks. We observe that the use of the UKG brings a remarkable performance boost. On average, our proposal improves the ROAS by 190.2% and the number of clicks by 190.8% across the datasets with respect to the benchmark with the KG disabled. This highlights the impact of KG techniques on the final performance compared to other techniques that do not rely on high-order connectivity information among entities. Note that Amazon-book and Last-fm rely on Freebase KG, while Yelp2018 uses the metadata in the dataset as KG, pointing out the flexibility of this technique to different topologies. On the other hand, the impact of the coordination strategy is smaller, showing an improvement of 15.4% in the ROAS and 15.2% in the number of clicks.

For the TRCA evaluation, we consider two cases: *i)* The distribution of the reward among agents is proportional to their respective bid value following a softmax function with temperature $\tau = 1$; *ii)* The reward is distributed equally among all the agents independently of their action ($\tau = 10^3$), i.e., a pure cooperative setting. We observe that TRCA$_{\tau=10^3}$ obtains consistently a lower performance as the reward distribution does not incentive the learning of better strategies. Although TRCA$_{\tau=1}$ works better, it encourages agents that are unlikely to get a click to bid higher to get a better reward. This favors the competition but shows a lower performance in our setting. Conversely, KRAF assigns a positive reward only to the winner agents, encouraging competition in a more efficient manner. Specifically, KRAF exhibit a 64.2% and 224.59% improvement of the ROAS with respect to TRCA$_{\tau=1}$ and TRCA$_{\tau=10^3}$, respectively.

The fairness metric measures the similarity of the ROAS across the agents, helping us to identify inequalities. For example, advertisers with lower affinity can be excluded from the auctions to increase the overall performance in the short term. However, this can be counterproductive for the advertising platforms in long term, as the number of advertisers can decrease over time (due to their poor performance), decreasing the competition and the overall revenue. We observe in Table 1 that KRAF exhibits fairness higher than 0.85 in all the cases and the highest value for Yelp2018 dataset.

Finally, note that all the MARL-based solutions expend the whole budget at the end of the day thanks to the reward signal r_t^{budget} (eq. (19)). This avoids collusion behavior on the agent policies by which they decrease the average bid value to increase the ROAS, harming the platform revenue (also reported in [37]).

Fig. 4 shows the average of the accumulated clicks at time t and the 10th and 85th percentiles with a colored shadow. The average and percentiles are computed over 10 consecutive days using data from the test set as in Table 1. We can observe in Fig. 4 how the budget is allocated over time by the different strategies. The curve of the strategies using on MARL (i.e. KRAF and TRCA) is almost linear, indicating that the clicks are evenly allocated throughout the day. This is a desirable property since it allows advertisers to bid for ad slots independently of the moment of the day, and stabilize the cost of the ad slots (e.g., avoiding higher bids when the advertisers have more budget). Conversely, we observe that ABP performs very close to the KRAF during the first third of the day with Amazon-book and Last-fm, but then the curve flattens because some of the agents run out of budget. This shows that a policy based on the pCTR (affinity) alone is working in a one-step prediction fashion without foreseeing the long-term results, and farsighted policies like MARL are much needed here for the constrained budget allocation task. Note that the affinity for ABP is computed using the UKG, and for this reason in some cases it outperforms KRAF$_{\text{-UKG}}$. Similarly, although KRAF$_{\text{-coord}}$ is very close to KRAF, it tends to flatten at the end of the day. This highlights the impact of the coordination strategy in KRAF that allows the agents to learn better farsighted budget allocation policies.

6 INDUSTRIAL USE CASES & ADS PRIVACY

In this section, we detail how KRAF can be adapted to different settings related to user privacy protection in advertising services, including contextual advertising, cohort-based advertising, and

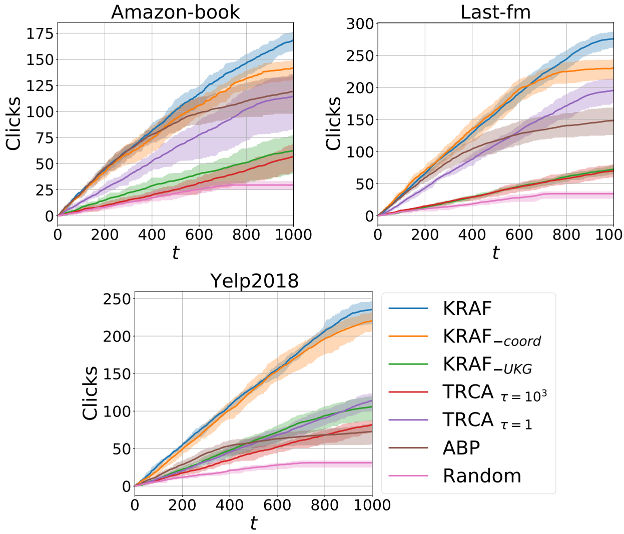

Figure 4: Number of clicks through one epoch for the different strategies and datasets.

interest-based advertising. We also illustrate how our advertising framework can be easily adapted to new advertising markets.

6.1 Contextual Advertising

A recent movement has appeared to regulate and restrict user tracking in online advertising, especially limiting the usage of 3rd party cookies, which are expected to be disabled permanently soon [2]. Contextual advertising is an approach that only uses the content the user is consuming when the ad slot is generated to recommend ads. Without using users' past behavioral history, users' privacy is improved but sacrificing service performance [13].

KRAF can be integrated into contextual advertising to enhance the richness of the information. We represent all the contextual tags as graph entities connected to the user who generated the ad slot. As all these tags have a corresponding entity in the KG, users and ads are indirectly connected through the UKG even when there is no past behavioral information about the users. Thus, via leveraging high-order connectivity, the advertising platform can provide high-quality recommendations. For instance, consider a user browsing the website of the Oscar Awards from which we extract contextual features. Some contextual features may be obvious like "movies", but we can also extract the latent concept of "luxury". Based on the latter, an expensive perfume ad could be recommended, relying on non-trivial relations between "luxury" and the perfume.

6.2 Cohort-based Advertising

Another approach to increase user privacy attempted by some leading companies is cohort-based advertising [36]. It relies on creating cohorts (or groups) of users with similar interests (e.g., behavioral data). Thus, when generating ad slots, users only reveal their cohort instead of behavioral or sensitive information. Cohort-based advertising presents some technical challenges as user clustering needs to be performed in a distributed manner, and it has a trade-off between user anonymity and ad serving performance.

To boost the performance, we could employ KRAF by considering cohorts of users instead of individual users in the UKG and record cohort-ads interactions. This will decrease the sparsity of our graph as a cohort will have more interaction (e.g., clicks) than an individual user, and increase the accuracy of predicted affinities. Similarly, we could define richer types of relations, for instance, different degrees of affinity relations based on the number of clicks. Thus, we increase the expressiveness of the model.

6.3 Interest-Based Advertising (IBA)

Interest-Based Advertising (IBA) has been recently proposed as an alternative to cohort-based advertising [9]. In IBA, at every epoch (e.g., one week) users generate a list of topics based on their behaviors (e.g., website visits or installed apps). A topic is a human-readable tag taken from the topics taxonomy set [9]. When a user generates an ad slot, a subset of the topics is sent for ad selection.

In our framework, each topic can be modeled as a graph entity connected to the UKG. When a user generates an ad slot, her past interactions are not available (the user is anonymous), but she will be connected to her topics and therefore integrated into the UKG. Although storing interaction with ads of individual users is not an option, we can store interactions between anonymous users' topics and ads, and thus, learn the high-order connectivity relations among them. Moreover, similar to cohort-based advertising, it can be beneficial to include different types of relations based on affinity

6.4 New Advertising Markets

New advertising markets are emerging in recent years and adaptability is a key factor for advertising platforms. Autonomous Advertising is an example, where companies offer free rides in autonomous cars in exchange for showing ads during the ride [7, 11]. In that case, assuming that we do not have previous information about the user (e.g., from the app to request the ride), other information can be added to the graph such as the origin and destination of the ride, the time, the day of the week, and so on. These graph entities can improve the targeting of the ads by providing geolocated advertisement adapted to the time and context of the users.

It is important to note that the different changes we describe in this section are related to the input graph. The following stages (i.e., embedding computation, affinity prediction, and bidding components) are not affected by introducing diverse and non-structure information, thus illustrating the flexibility of the advertising platform in very diverse scenarios.

7 CONCLUSION

In this work, we presented KRAF, a novel advertising frameworks that combines Knowledge Graph (KG) techniques with Multi-Agent Reinforcement Learning (MARL). It tackles the real-time bidding problem in advertising with heterogeneous and limited input information. We model the input information as a graph, which is integrated into a KG to leverage high-order connectivity. The embeddings of the graph nodes are used to define the states of the MARL algorithm that learns efficient bidding strategies for the advertisers using a novel coordination strategy among agents. We evaluated our advertising framework on real-world datasets assessing the impact of each of its components. Remarkably, KRAF is able to deal with different levels of user privacy and the advent of new advertising markets with more heterogeneous data.

REFERENCES

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. 1247–1250. https://doi.org/10.1145/1376616.1376746

[2] Pamela Bump. 2021. The Death of the Third-Party Cookie: What Marketers Need to Know About Google's 2022 Phase-Out. HubSpot. https://blog.hubspot.com/marketing/third-party-cookie-phase-out

[3] Han Cai, Kan Ren, Weinan Zhang, Kleanthis Malialis, Jun Wang, Yong Yu, and Defeng Guo. 2017. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 661–670.

[4] Yang Deng, Yaliang Li, Fei Sun, Bolin Ding, and Wai Lam. 2021. Unified conversational recommendation policy learning via graph-based reinforcement learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1431–1441.

[5] Manxing Du, Alexander I Cowen-Rivers, Ying Wen, Phu Sakulwongtana, Jun Wang, Mats Brorsson, and Radu State. 2019. Know Your Enemies and Know Yourself in the Real-Time Bidding Function Optimisation. In *2019 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 973–981.

[6] Manxing Du, Redouane Sassioui, Georgios Varisteas, Mats Brorsson, Omar Cherkaoui, et al. 2017. Improving real-time bidding using a constrained markov decision process. In *International conference on advanced data mining and applications*. Springer, 711–726.

[7] Forbes. 2021. Autonomous Advertising: Mapping The Future Of Machine Learning In Ad Tech. Article. https://www.forbes.com/sites/forbestechcouncil/2021/03/30/autonomous-advertising-mapping-the-future-of-machine-learning-in-ad-tech/?sh=636968852960

[8] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*. PMLR, 1587–1596.

[9] Google. 2022. Interest-based advertising (IBA). Technical Specification. https://github.com/jkarlin/topics

[10] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A Survey on Knowledge Graph-Based Recommender Systems. *CoRR* (2020).

[11] Florian Herrmann, Sebastian Stegmüller, Lukas Block, and Maximilian Werner. 2021. Keynote Speech Disruption in mobility–new trends, new concepts and new business models?! In *Vehicles of Tomorrow 2019*. Springer, 1–9.

[12] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 505–514.

[13] IAB. 2021. A Guide to Modernized Contextual Advertising. Industry Paper. http://iabcanada.com/content/uploads/2021/04/IABCanadaContextualAdvertising_043021.pdf

[14] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. 2016. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397* (2016).

[15] Raj Jain, Arjan Durresi, and Gojko Babic. 1999. Throughput fairness index: An explanation. In *ATM Forum contribution*, Vol. 99.

[16] Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. 2018. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 2193–2201.

[17] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[18] Minne Li, Zhiwei Qin, Yan Jiao, Yaodong Yang, Jun Wang, Chenxi Wang, Guobin Wu, and Jieping Ye. 2019. Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning. In *The world wide web conference*. 983–994.

[19] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[20] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

[21] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems* 30 (2017).

[22] Zhaoqing Peng, Junqi Jin, Lan Luo, Yaodong Yang, Rui Luo, Jun Wang, Weinan Zhang, Haiyang Xu, Miao Xu, Chuan Yu, et al. 2020. Learning to Infer User Hidden States for Online Sequential Advertising. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2677–2684.

[23] Claudia Perlich, Brian Dalessandro, Rod Hook, Ori Stitelman, Troy Raeder, and Foster Provost. 2012. Bid optimizing and inventory scoring in targeted online advertising. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 804–812.

[24] PWC. 2021. Internet Advertising Revenue Report. https://www.iab.com/wp-content/uploads/2021/04/IAB_2020-Internet-Advertising-Revenue-Report-Webinar_4.7.21-PwC-1.pdf

[25] R. Qian. 2013. Understand your world with Bing. Bing search blog. https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[27] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S Yu, Yading Yue, and Bin Wu. 2015. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 453–462.

[28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[29] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 417–426.

[30] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *The world wide web conference*. 2000–2010.

[31] Jun Wang, Weinan Zhang, and Shuai Yuan. 2016. Display advertising with real-time bidding (RTB) and behavioural targeting. *arXiv preprint arXiv:1610.03013* (2016).

[32] Pengfei Wang, Yu Fan, Long Xia, Wayne Xin Zhao, ShaoZhang Niu, and Jimmy Huang. 2020. KERL: A knowledge-guided reinforcement learning model for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 209–218.

[33] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 950–958.

[34] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[35] Yu Wang, Jiayi Liu, Yuxiang Liu, Jun Hao, Yang He, Jinghe Hu, Weipeng P Yan, and Mantian Li. 2017. Ladder: A human-level bidding agent for large-scale real-time online auctions. *arXiv preprint arXiv:1708.05565* (2017).

[36] Web Incubator CG. 2021. Federated Learning of Cohorts (FLoC). Technical Specification. https://github.com/WICG/floc

[37] Chao Wen, Miao Xu, Zhilin Zhang, Zhenzhe Zheng, Yuhui Wang, Xiangyu Liu, Yu Rong, Dong Xie, Xiaoyang Tan, Chuan Yu, et al. 2022. A Cooperative-Competitive Multi-Agent Framework for Auto-bidding in Online Advertising. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1129–1139.

[38] Di Wu, Cheng Chen, Xun Yang, Xiujun Chen, Qing Tan, Jian Xu, and Kun Gai. 2018. A multi-agent reinforcement learning method for impression allocation in online display advertising. *arXiv preprint arXiv:1809.03152* (2018).

[39] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*. PMLR, 5453–5462.

[40] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. 2018. Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 5571–5580.

[41] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.

[42] Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1077–1086.

[43] Weinan Zhang, Shuai Yuan, Jun Wang, and Xuehua Shen. 2014. Real-time bidding benchmarking with ipinyou dataset. *arXiv preprint arXiv:1407.7073* (2014).

[44] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 635–644.

[45] Wayne Xin Zhao, Gaole He, Kunlin Yang, Hongjian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. 2019. Kb4rec: A data set for linking knowledge bases with recommender systems. *Data Intelligence* 1, 2 (2019), 121–136.

[46] Sijin Zhou, Xinyi Dai, Haokun Chen, Weinan Zhang, Kan Ren, Ruiming Tang, Xiuqiang He, and Yong Yu. 2020. Interactive recommender system via knowledge graph-enhanced reinforcement learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 179–188.