KoMen: Domain Knowledge Guided Interaction Recommendation for Emerging Scenarios

Yiqing Xie[1*], Zhen Wang[2], Carl Yang[3], Yaliang Li[2], Bolin Ding[2], Hongbo Deng[2], Jiawei Han[1]
[1]University of Illinois, Urbana Champaign, IL, USA [2]Alibaba Group [3]Emory University, GA, USA
{xyiqing2, hanj}@illinois.edu {jones.wz, yaliang.li, bolin.ding, dhb167148}@alibaba-inc.com j.carlyang@emory.edu

ABSTRACT

User-User interaction recommendation, or interaction recommendation, is an indispensable service in social platforms, where the system automatically predicts with whom a user wants to interact. In real-world social platforms, we observe that user interactions may occur in diverse scenarios, and new scenarios constantly emerge, such as new games or sales promotions. There are two challenges in these emerging scenarios: (1) The behavior of users on the emerging scenarios could be different from existing ones due to the diversity among scenarios; (2) Emerging scenarios may only have scarce user behavioral data for model learning. Towards these two challenges, we present KoMen, a Domain **Kno**wledge Guided **Me**ta-learning framework for Interaction Recommendatio**n**. KoMen first learns a set of global model parameters shared among all scenarios and then quickly adapts the parameters for an emerging scenario based on its similarities with the existing ones. There are two highlights of KoMen: (1) KoMen customizes global model parameters by incorporating domain knowledge of the scenarios[1], which captures scenario inter-dependencies with very limited training. (2) KoMen learns the scenario-specific parameters through a mixture-of-expert architecture, which reduces model variance resulting from data scarcity while still achieving the expressiveness to handle diverse scenarios. Extensive experiments demonstrate that KoMen achieves state-of-the-art performance on a public benchmark dataset and a large-scale real industry dataset. Remarkably, KoMen improves over the best baseline w.r.t. weighted ROC-AUC by 2.14% and 2.03% on the two datasets, respectively. [2]

ACM Reference Format:
Yiqing Xie, Zhen Wang, Carl Yang, Yaliang Li, Bolin Ding, Hongbo Deng, Jiawei Han. 2022. KoMen: Domain Knowledge Guided Interaction Recommendation for Emerging Scenarios. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485447.3512177

[1]An example is a taxonomy that organizes scenarios by their purposes and functions.
[2]Our code is available at: https://github.com/Veronicium/koMen.

*Work done when Yiqing was an intern at Alibaba. This work was supported by Alibaba Group through Alibaba Research Intern Program.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512177

1 INTRODUCTION

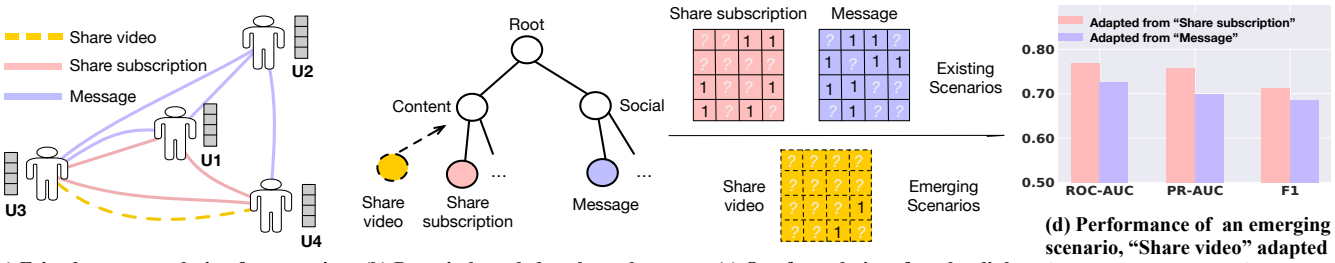
Interaction recommendation is an essential factor for improving user stickiness and activeness [14, 25, 29, 30, 34, 42], which aims to predict the interpersonal interactions between users. A typical scenario of interaction recommendation is "item sharing" in e-commerce platforms, where the system predicts with whom the current user wants to share an item.

Prior studies on interaction recommendation mainly focus on a single scenario with one type of interaction, such as "follow" or "like" [10, 22, 29, 38]. However, in real-world social platforms, we observe that there are often diverse scenarios where user interactions occur [31, 37]. As shown in Fig. 1a, a user may send messages, share videos, or subscribe to the same blogger (i.e., share subscriptions) with others. Furthermore, new scenarios constantly emerge (e.g., two users may play together in a newly-released game or share coupons in a new sales promotion). It is challenging to handle these emerging scenarios for two reasons: (1) A user may interact with different groups of people in different scenarios (e.g., they could message many people but only share videos with friends with mutual interest). (2) There are very limited records in these emerging scenarios, which makes it difficult to train a model from scratch.

Considering these two challenges, we formulate interaction recommendation as *few-shot link prediction on multiplex graphs*, as illustrated in Fig. 1(a) and 1(c). We regard users as nodes and interactions among users as edges. Each scenario corresponds to an individual type of edge, making the graph "multiplex". Each emerging scenario corresponds to a new edge type that only emerges in inference with few visible edges, making the problem "few-shot".

Recent studies on representation learning of multiplex graphs are closely related to our formulation. These methods learn different parameters for different edge types (scenarios) based on their own training data [3, 20, 44]. However, this may lead to over-fitting when it comes to emerging scenarios as the training data is scarce. To handle the emerging scenarios, one solution is to regard each scenario as one task and apply meta-learning techniques [8, 11, 39, 40], which aims to quickly adapt a model to individual tasks. Nevertheless, general meta-learning methods keep a set of global parameters for all the diverse scenarios and only rely on a few gradient steps for customization [8, 12, 45, 46]. To better customize for each scenario, existing studies show that model performance can be improved if edge types with similar topological structures share more parameters and vice versa [2, 39–41]. Yet, emerging scenarios may only have a small percentage of edges available, which have limited expressiveness to represent the generic topological structure.

One key observation missing in prior studies is that, in practice, there is often readily available domain knowledge that reflects the meaning of different scenarios, such as a taxonomy organizing the scenarios by their purposes and functions [36]. As shown in



(a) Friend recommendation for emerging scenarios (in this case, Share video).

(b) Domain knowledge about the purpose and function of each scenario.

(c) Our formulation: few-shot link prediction on multiplex networks.

(d) Performance of an emerging scenario, "Share video" adapted from the parameters of two existing scenarios separately.

Figure 1: The illustrations of our setting and formulation, and a case study indicating the usefulness of domain knowledge.

Fig. 1b, "share video" and "share subscription" are both about the content of a user's interest, while "message" reveals the user's social relations. Intuitively, user behaviors under scenarios with similar purposes and functions are also similar. For instance, a user may send messages to many friends, but only prefer to share videos and subscriptions with friends with mutual interest. Our experimental results also support this intuition – we observe that the emerging scenario "share video" obtains better performance when directly copying the parameters of "share subscription" than "message" (Fig. 1d). Although the complicated inter-dependencies between scenarios may not be perfectly captured by a taxonomy, the purposes and functions of scenarios represented by the taxonomy are still helpful for understanding and predicting user behaviors, especially when we do not have massive user data.

Based on our observation, in this paper, we propose a Domain **Kn**owledge Guided **Me**ta-learning Framework for Interaction Recommendatio**n** (denoted by KoMen) to tackle the challenge of data scarcity in emerging scenarios. Our framework consists of two modules: a domain knowledge guided scenario representation module and a scenario similarity aware link prediction module. The first module learns a representation for each scenario that reflects their similarities. The representations are initialized by encoding the domain knowledge about the purposes and functions of each scenario and updated based on training data, in case the domain knowledge is not perfect. The link prediction module extracts the information in each edge type (i.e., scenario) by a graph neural network (GNN) [13] and aggregates the information in all existing edge types with a mixture-of-expert (MoE) [24] architecture to facilitate the prediction. The aggregating coefficients depend on the scenario representations. The use of MoE reduces the variance of the aggregating coefficients for emerging scenarios with scarce data, while still preserving the expressiveness for handling diverse scenarios. The two modules are optimized using meta-learning, which aims to quickly adapt a model to individual scenarios.

We examine the effectiveness of KoMen on both a public dataset (Youtube) and a real industry dataset (Taobao). Empirical results show that KoMen achieves significant improvements over the state-of-the-art methods – 2.14% and 2.03% gains over the best baseline under weighted ROC-AUC on the respective datasets. We also conduct ablation and case studies to demonstrate the utility of domain knowledge and the effectiveness of our model structure for integrating domain knowledge into the meta-learning paradigm.

Contributions. (1) Our work is the first attempt for few-shot interaction recommendation, which predicts with whom a user wants to interact under an emerging scenario with scarce data. (2) We propose a novel method KoMen, which incorporates domain knowledge into data-driven learning to capture the similarities among scenarios, and leverages the MoE architecture to learn scenario-specific parameters with relatively small variance. (3) Experiments show that KoMen outperforms state-of-the-art methods on a public dataset as well as a real-world industry dataset.

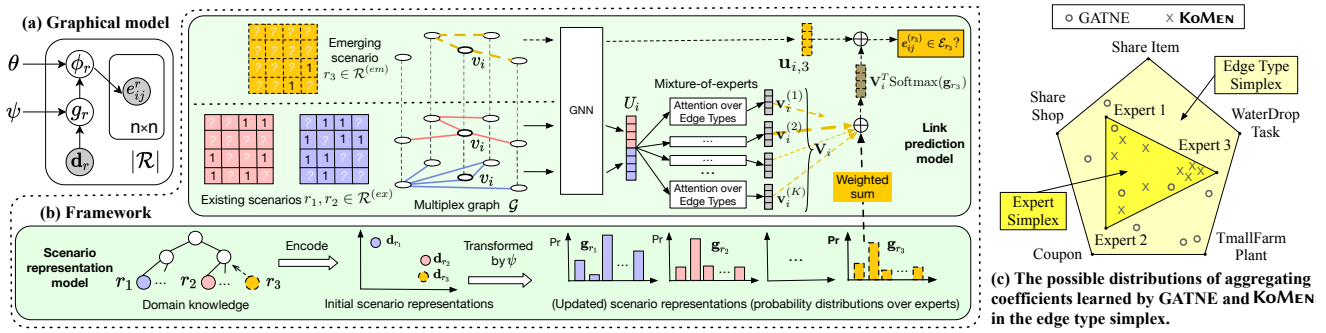
2 PROBLEM SETUP

Supervised Interaction Recommendation. We formulate interaction recommendation as link prediction. Considering diverse scenarios, we are given a graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where the node set $\mathcal{V}$ represents users, the node attributes $\mathcal{X}$ represents user profiles and the edge set $\mathcal{E} = \bigcup_{r\in\mathcal{R}} \mathcal{E}_r$ represents the interactions between users under $|\mathcal{R}|$ different scenarios. For each edge type $r \in \mathcal{R}$, only a subset of the edges $\mathcal{E}_r^{(\text{tr})} \subset \mathcal{E}_r$ are visible, and interaction recommendation aims to find all missing edges $\mathcal{E}_r^{(\text{ts})} = \mathcal{E}_r \setminus \mathcal{E}_r^{(\text{tr})}$.
Few-shot Interaction Recommendation. Considering the existence of emerging scenarios, we further distinguish the edge types into existing ones and emerging ones $\mathcal{R} = \mathcal{R}^{(\text{ex})} \bigcup \mathcal{R}^{(\text{em})}$. During training, only the edges of existing types $\mathcal{R}^{(\text{ex})}$ are visible. In inference, a small percentage of edges of emerging types $\mathcal{E}_r^{(\text{tr})}, r \in \mathcal{R}^{(em)}$ are available for fine-tuning, and few-shot interaction recommendation aims to find all missing edges of these emerging edge types $\mathcal{R}^{(\text{em})}$.

An example of this problem is shown in Fig. 1a and Fig. 1c, where "Share subscription" and "Message" are two existing edge types and "Share video" is an emerging one. The entries with "1" in the adjacency matrices correspond to visible edges $\mathcal{E}_r^{(\text{tr})}$. In addition, we assume the access to domain knowledge is available, which is exemplified by the taxonomy of scenarios (formulated as edge types) as in Fig. 1b but is not restricted to any particular form.

3 METHODOLOGY

The overall framework of KoMen is shown in Fig. 2(b), which consists of a scenario representation module (Sec. 3.1) and a link prediction module (Sec. 3.2). The scenario representation module aims to represent the similarities between scenarios in a set of low-dimensional vectors. These vectors are initialized based on



(c) The possible distributions of aggregating coefficients learned by GATNE and KoMen in the edge type simplex.

Figure 2: (a) The graphical model of KoMen. ψ is the parameters in scenario representation model and θ is the ones of link prediction model. (b) The overall framework. The scenario representation model encodes the domain knowledge as the initial representation $\mathbf{d}_r$ and maps it to the (updated) scenario representation $\mathbf{g}_r$ by a trainable neural network ψ. The link prediction model trains a set of experts, each of which learns the attention over edge types separately. Different scenarios choose different combinations of experts based on their representations $\mathbf{g}_r$. (c) Each emerging scenario aggregates the information from existing edge types with a set of coefficients (denoted by $\circ$ and $\times$). In KoMen, each expert learns the attention over existing edge types, and each scenario learns the attention over experts, so the coefficients will be confined in the expert simplex.

domain knowledge about the scenarios and are transformed by several trainable layers. By our design, the (updated) representation of a scenario depends on both its purposes and functions and the pattern of its training data. The link prediction module contains a Graph Neural Network (GNN) [13] that encodes the information in each edge type and an MoE [24] architecture that aggregates the information of different edge types, where the coefficients can be different among scenarios based on their (updated) representations. This allows scenarios with similar representations to use the multiplex graph in a similar way. Overall, as shown in Fig. 2(a), we exploit meta-learning to optimize the model with domain knowledge guided scenario-specific initialization (Sec. 3.3 and Sec. 3.4).

3.1 Domain Knowledge Guided Scenario Representation Learning

As shown in Fig. 2(b), KoMen aims to learn a set of low-dimensional vectors as scenario representations, so that two scenarios with similar user behaviors will also have representations with small distance in the Euclidean space.

Although pure data-driven methods may be effective for some existing scenarios with massive records, they may lead to over-fitting when the training data are limited, which is common for emerging scenarios. To solve the data scarcity issue, before training our model, KoMen deducts a prior from available domain knowledge and incorporates it into our learning-based models.

To exemplify this idea, one possible form of domain knowledge is a taxonomy of scenarios, where each scenario is a leaf node and is organized by its purposes and functions, as shown in Fig. 1b. We obtain the initial representation $\mathbf{d}_r$ of each scenario by training Poincaré embeddings [19] on the taxonomy. It is an embedding method designed for tree structures, which encourages scenarios with smaller distances in the taxonomy to have similar embeddings.

In addition to the domain knowledge, the observed data of an emerging scenario also provides crucial information on user behavior. Hence, we further pass the initial representation $\mathbf{d}_r$ to several multilayer perceptron (MLP) layers h_ψ, which are trained on the observed data. We take one layer as an example:

$$\mathbf{g}_r = h_\psi(\mathbf{d}_r) = \sigma(\mathbf{W}\mathbf{d}_r + \mathbf{b}), \tag{1}$$

where σ is the activation function. Only the parameter $\psi = (\mathbf{W}, \mathbf{b})$ are updated during training while the initial representation $\mathbf{d}_r$ is fixed. In this way, the (updated) scenario representation $\mathbf{g}_r$ encodes the domain knowledge because it is conditioned on the initial representation $\mathbf{d}_r$. It also reflects the distribution of training data because it is also conditioned on ψ, which is trained on the observed data $\mathcal{R}^{(ex)}$. On one hand, the domain knowledge serves as a prior that confines the search space of $\mathbf{g}_r$. On the other hand, even if the domain knowledge is not perfect, $\mathbf{g}_r$ could still be adjusted based on the training data.

Notice that we only use taxonomy as an example and the model can take various forms of domain knowledge with the initial representations computed in different ways. For instance, given a short paragraph describing the function of each scenario, the model may use the document representation of each paragraph as the initial representation $\mathbf{d}_r$. We could also design several features for the scenario as $\mathbf{d}_r$, such as whether the interaction is one-way or two-way or whether the user gets rewards after the interaction.

Furthermore, in case that domain knowledge is not available, $\mathbf{g}_r$ can still be learned from the training data. For instance, we can also replace $\mathbf{d}_r$ with a set of low-dimensional embeddings, which is randomly initialized and updated during training. In this way, $\mathbf{g}_r$ is learned in a purely data-driven way.

3.2 Scenario Similarity Aware Link Prediction

As Fig. 2(b) shows, our link prediction module consists of a GNN [13] and an MoE [24] architecture. The GNN encodes the information of each edge type individually, and each expert in MoE computes the attention over all existing edge types. By passing the scenario representations reflecting the similarities between scenarios into the MoE, we encourage scenarios with similar user behaviors to use the multiplex graph in similar ways.

Base Encoder. For each node v_i, we calculate its base embedding under each edge type by applying the graph neural network (GNN) [13]. For simplicity, we take one layer as an example:

$$\mathbf{u}_{i,r} = \sigma(\mathbf{W}_g \text{mean}\{\mathbf{x}_j, \forall e_{ij} \in \mathcal{E}_r^{(\text{tr})}\} + \mathbf{b}_g) \tag{2}$$

where $\mathbf{W}_g$ and $\mathbf{b}_g$ are learnable parameters, $\mathbf{x}_i$ is the node attributes and σ denotes the Sigmoid function. For each node, we concatenate the base embeddings of all existing edge types as a s-by-m matrix $\mathbf{U}_i = [\mathbf{u}_{i,1}, \mathbf{u}_{i,2}, \ldots, \mathbf{u}_{i,m}]$, where s is the dimension of embeddings and m is the number of existing edge types. For an emerging scenario and its corresponding edge type r, the base embeddings $\mathbf{u}_{i,r}$ reflects its own training data, and the matrix $\mathbf{U}_i$ captures the information in all existing scenarios to facilitate the prediction.

Scenario Similarity Aware Aggregation. As the user behaviors could vary among diverse scenarios, we allow different scenarios to aggregate the information of all edge types in different ways. Specifically, we train an MoE [24] structure with K experts, each of which learns an aggregated embedding $\mathbf{v}_i^{(k)}$ as a weighted summation of the vectors in $\mathbf{U}_i$. The aggregating weights $\mathbf{a}_i^{(k)}$ over the existing edge types are computed using self-attention mechanism [16]:

$$\begin{aligned} \mathbf{a}_i^{(k)} &= \text{Softmax}(\mathbf{w}_k^{\mathrm{T}} \tanh(\mathbf{W}_k \mathbf{U}_i))^{\mathrm{T}}, \\ \mathbf{v}_i^{(k)} &= \mathbf{M}_k^{\mathrm{T}} \mathbf{U}_i \mathbf{a}_i^{(k)}, \end{aligned} \tag{3}$$

where $\mathbf{w}_k \in \mathbb{R}^p$, $\mathbf{W}_k \in \mathbb{R}^{p \times s}$ and $\mathbf{M}_k \in \mathbb{R}^{s \times s}$ are the parameters of the k-th expert, and p is the hidden dimension. Noted that the experts have the same architecture but respective model parameters.

After the transformation ψ, the scenario representation $\mathbf{g}_r$ has the same dimension as the number of experts. We then take the Softmax function over it so that each entry of g_r becomes the weight of its corresponding expert. The final node embedding used for predicting links of r is computed as follows:

$$\mathbf{x}_{i,r} = \beta \mathbf{u}_{i,r} + (1 - \beta) \mathbf{V}_i^T \text{Softmax}(\mathbf{g}_r), \tag{4}$$

where $\mathbf{V}_i = [\mathbf{v}_i^{(1)}, \ldots, \mathbf{v}_i^{(K)}]$ is a s-by-K matrix, β is a scalar to balance the influence of the information from the edge type r itself and that from all the existing edge types. In short, scenarios with similar user behaviors are trained to have similar representations $\mathbf{g}_r$, and the MoE allows them to aggregate the information in the graph in similar ways.

Advantages. Although both existing multiplex graph embedding methods [3, 44] and KoMen produce the final node embedding for each type by aggregating the information in all edge types, their aggregating coefficients are determined in different ways: (1) In [3, 44], each scenario directly aggregates $\mathbf{U}_i = [\mathbf{u}_{i,1}, \mathbf{u}_{i,2}, \ldots, \mathbf{u}_{i,m}]$ with the coefficients learned sorely from its own training data, $\mathcal{E}_r^{(\text{tr})}$. There could be multiple layers associated with the computation of the coefficients, easily leading to over-fitting for emerging scenarios with scarce training data. In contrast, KoMen aggregates the outputs of experts instead of edge types, each of which has been a mixture of $\mathbf{U}_i$. Thus, deducing from Eq. (3) and Eq. (4), the coefficients of each edge type should be $[\mathbf{a}_i^{(1)}, \ldots, \mathbf{a}_i^{(K)}]\mathbf{g}_r$. Since the parameters of each expert and ψ are shared by all the scenarios, none of the involved parameters are learned solely from the training data of edge type r. Hence, KoMen alleviates the over-fitting issue resulting from data scarcity, while still having the same expressiveness for diverse scenarios. (2) Fig. 2(c) illustrates the difference between KoMen and GATNE [3] by visualizing their possible distributions over the edge types (i.e., the possible aggregating coefficients). Due to the very limited number of training examples of emerging scenarios, in GATNE, the variance of their estimated mixing coefficients could be huge, separating around the edge type simplex. In comparison, KoMen regularizes the aggregating coefficients to lie in the expert simplex, which is often a more reserved subset of the edge type simplex so that the variance of the estimated aggregating coefficients would be reduced.

3.3 Optimization

In this section, we describe how to optimize the two modules to make them cooperate with each other well and be capable of quickly adapting to emerging scenarios. For simplicity, we pack the model parameters shared by all scenarios as $\theta_G = (\mathbf{W}_g, \mathbf{b}_g)$, the parameters in experts as $\theta_E = \{(\mathbf{w}_k, \mathbf{W}_k, \mathbf{M}_k) | 1 \le k \le K\}$, and the parameters regarding the scenario representations as $\psi = (\mathbf{W}, \mathbf{b})$.

To estimate the existence of e_{ij}^r, an edge of edge type r, we compute the conditional probability of observing v_j given v_i:

$$\Pr(e_{ij}^r \in \mathcal{E}_r) = \Pr(v_j | v_i, r) = \frac{\mathbf{x}_{i,r}^{\mathrm{T}} \mathbf{x}_{j,r}}{\sum_{v_{j'} \in \mathcal{V}} \mathbf{x}_{i,r}^{\mathrm{T}} \mathbf{x}_{j',r}}, \tag{5}$$

where the probability is parmeterized by the model parameters of the link prediction module (i.e., θ_G, θ_E) customized by $\mathbf{g}_r$ according to Eq. 4. With the defined conditional probability, we aim to optimize θ_G, θ_E and ψ by minimizing the negative log-likelihood of observed co-occurrences for each existing edge type $r \in \mathcal{R}^{(\text{ex})}$:

$$\mathcal{L}\Big(\bigcup_{r \in \mathcal{R}^{(\text{ex})}} \mathcal{E}_r^{(tr)}\Big) = -\sum_{r \in \mathcal{R}^{(\text{ex})}} \sum_{v_i \in \mathcal{V}} \sum_{v_j : e_{ij}^r \in \mathcal{E}_r^{(\text{tr})}} \log \Pr(v_j | v_i, r). \tag{6}$$

In practice, we construct the negative samples by randomly choosing node pairs with no edge in the training set. Due to the huge number of nodes, we adopt either noise contrastive estimation (NCE) [9] or negative sampling (NS) [18] to approximate Eq. 5 by $\log \Pr(v_j | v_i, r) \approx \log(\sigma(\mathbf{x}_{i,r}^{\mathrm{T}} \mathbf{x}_{j,r})) + \sum_{j' \sim \Pr_n(v)} [\log(\sigma(-\mathbf{x}_{i,r}^{\mathrm{T}} \mathbf{x}_{j',r}))]$, where $\Pr_n(v)$ stands for a distribution over $\mathcal{V}$ used for sampling "negative" examples.

To handle emerging scenarios, we adopt gradient-based meta-learning [8] to optimize θ_G, θ_E and ψ jointly, which aims to learn an initialization for the model parameters that can quickly adapt to any particular task after fine-tuning on very few examples. Unlike traditional meta-learning methods [8] which uses a global initialization (i.e., θ_G, θ_E) for all scenarios, since the combination of experts is based on scenario representations $\mathbf{g}_r$, we allow each scenario to have its specific initialization (i.e., ϕ_r). This idea is presented via the graphical model shown in Fig. 2 and discussed in Sec. 3.4. The pseudocode of our training procedure is presented in Algorithm 1.

To make the training procedure more efficient, instead of updating all parameters in both the inner and outer loop as most existing methods [8, 39] do, we fix the parameters of the base encoder, i.e., θ_G in the inner loop and only update it in the outer loop [23].

Analysis. Theoretically, we can recast optimization-based meta-learning in the hierarchical Bayesian framework, which enables us to interpret KoMen by the graphical model shown in the upper left

Algorithm 1 Meta-learning with task-specific customization.

Input Multiplex graph $G = (\mathcal{V}, \bigcup_{r\in\mathcal{R}^{(\text{ex})}} \mathcal{E}_r^{(\text{tr})}, \mathcal{X})$, initial scenario representations $\mathbf{d}_r, r \in \mathcal{R}^{(\text{ex})}$, step size α_1, α_2, update steps K
Output Model parameters θ_G, θ_E and ψ.

```
Randomly initialize θ_G, θ_E and ψ
while not converge do
    for r ∈ R^(ex) do
        Sample E_r^(spt), E_r^(qry) from E_r^(tr)
        Initialize ψ^(0) ← ψ
        for t in [0 : T − 1] do
            Update ψ^(t+1) ← ψ^(t) − α1 ∇_{ψ^(t)} L(E_r^(spt); ψ^(t))
        end for
        Compute g_r based on ψ^(T) by Eq. 3
        Initialize θ_{E,r}^(0) ← {Softmax(g_r)θ_{E,k} | 1 ≤ k ≤ K}
        for t in [0 : T − 1] do
            Update θ_{E,r}^(t+1) ← θ_{E,r}^(t) − α2 ∇_{θ_{E,r}^(t)} L(E_r^(spt); θ_{E,r}^(t))
        end for
    end for
    Update θ_G, θ_E and ψ by minimizing overall loss Σ_{r∈R^(ex)} L(E_r^(qry); {θ_G, θ_{E,r}^(T), ψ^(T)})
end while
```

of Fig. 2. The corresponding generative process can be defined by:

$$\Pr(\mathcal{E}, \mathcal{D}|\theta_G, \theta_E, \psi) = \prod_{r\in\mathcal{R}} \Big(\iiint \Pr(\mathcal{E}_r|\phi_r, \theta_G) \Pr(\phi_r|\theta_E, \mathbf{g}_r) \Pr(\mathbf{g}_r|\psi, \mathbf{d}_r) \Pr(\mathbf{d}_r)\, \mathrm{d}\mathbf{d}_r\, \mathrm{d}\phi_r\, \mathrm{d}\mathbf{g}_r \Big), \tag{7}$$

where ϕ_r represents the parameters of the link prediction module customized by $\mathbf{g}_r$ from θ_E. In KoMen, we derive $\mathbf{d}_r$ from the domain knowledge by a certain adopted embedding algorithm and regard it as a point estimate: $\Pr(\mathbf{d}_r) = \delta(\mathbf{d}_r)$, where δ denotes the Dirac delta function. Moreover, we consider the point estimates $\Pr(\mathbf{g}_r|\psi, \mathbf{d}_r) = \delta(h_\psi(\mathbf{d}_r))$ and approximate $\Pr(\mathcal{E}_r|\phi_r) \Pr(\phi_r|\theta_E, \mathbf{g}_r)$ by using Eq. 5 as well as Eq. 4. The approximations enhance the computational efficiency and thus enable KoMen to deal with large-scale graphs.

3.4 Discussion

Our framework is highlighted by (1) its mechanism of synthesizing information from the multiplex graph, and (2) the domain knowledge guided parameter adaptation. In this section, we elaborate the rationale behind them by comparing with related prior studies.

GATNE [3]. KoMen is reduced to GATNE when we ignore the domain knowledge and force each scenario to use its own dedicated expert. In this way, the attention over edge types $\mathbf{a}_i^{(k)}$ becomes scenario-specific (i.e., $\mathbf{a}_i^{(r)}$), and is only updated by $\mathcal{E}_r^{(tr)}$.

MAML [8]. KoMen is reduced to MAML when $K = 1$, i.e., using only one expert for all the scenarios. In this case, the prediction no longer depends on $\mathbf{g}_r$ and each scenario directly adapts from the same global parameter without any customization.

Meta-Graph [2] and GFL [41]. Both of the methods customize the global initialization θ_E for ϕ_r by a graph signature which is extracted based on the graph topological structure. Since the access to the whole graph structure is unavailable in the few-shot link prediction setting, we further leverage the domain knowledge to guide the customization.

T-NAS [15]. When we regard the scenario representation $\mathbf{g}_r$ as the architecture parameters and force $\mathbf{g}_r$ to be hard (i.e., one-hotted), updating ψ is equivalent to adapting the neural architecture.

Table 1: Dataset Statistics.

Datasets	# Nodes	# Edges	# Existing e-types	# New e-types
YouTube	2,000	1,310,617	2	2
Taobao	267,869	258,306	20	5

4 EXPERIMENTS

In this section, we conduct extensive experiments to answer four research questions: (**RQ1**) How do KoMen and its ablations perform compared against adapted state-of-the-art approaches for interaction recommendation on emerging scenarios? (**RQ2**) Can KoMen also help with existing scenarios? (**RQ3**) Which kind of scenarios benefit the most from KoMen, and how does KoMen perform on specific scenarios? (**RQ4**) How is KoMen impacted by its hyper-parameters?

4.1 Experimental Settings

Datasets. We validate the performance of our method on few-shot interaction recommendation on a publicly available benchmark dataset YouTube [26] and a real-world industry dataset Taobao. We also conduct experiments under the standard supervised setting to demonstrate that our method can also help existing scenarios. The statistics of both datasets are listed in Table 1. Under the few-shot setting, we train our model on existing edge types and test the performance on emerging edge types. Under supervised setting, we only focus on link prediction on existing edge types. More details about dataset construction are put in the Appendix.

Baselines. We compare the performance of KoMen with several state-of-the-art methods adapted for interaction recommendation under both the few-shot and supervised settings.

Baselines for the few-shot setting. To test our framework on emerging scenarios, we first compare KoMen with *Meta-learning methods*, which are state-of-the-art methods in few-shot learning. These methods include general meta-learning methods: **MAML** [8] and **HSML** [39], and graph-based meta-learning methods: **G-Meta** [12] and **Meta-graph** [2]. To adapt general meta-learning methods to our setting, we use GATNE [3] as the model structure and train its parameters using meta-learning approaches. For G-Meta and meta-graph, we follow their original setting and treat each edge type as an individual graph. Following previous studies [12, 32], we also compare with several basic methods [27]: **KNN**, **Fine-tune** and **No-fine-tune**. All three methods use GCN [13] to encode the input graph. KNN compares each test pair with the K-nearest train pairs in the embedding space. Fine-tune and No-fine-tune further train several fully connected layers on the top of GCN. In Fine-tune, both the GCN model and the top layers will be updated on each emerging scenario's support set, while No-fine-tune only updates the parameters of the top layers. Finally, we compare our

Table 2: Experimental results under the few-shot setting on YouTube and Taobao datasets.

Methods	*YouTube-new*						*Taobao-new*					
	ROC	PR	F1	wROC	wPR	wF1	ROC	PR	F1	wROC	wPR	wF1
KNN	48.90	48.18	50.76	47.16	46.34	49.99	48.71	49.43	49.13	50.03	50.08	50.06
No-fine-tune	54.96	53.84	53.54	54.26	52.98	53.44	54.13	51.86	52.63	51.58	50.27	50.73
Fine-tune	58.61	57.29	54.69	62.61	58.71	57.38	52.48	48.71	50.29	51.25	50.16	50.13
MoE	70.17	68.17	65.68	69.19	66.69	64.94	54.50	53.53	53.39	55.90	54.64	54.42
G-Meta	68.21	67.62	66.98	66.27	65.38	64.46	57.78	55.43	57.27	60.46	57.19	57.82
Meta-graph	71.35	64.71	65.32	70.27	62.95	65.26	N/A	N/A	N/A	N/A	N/A	N/A
MAML	70.61	67.91	66.29	69.60	66.54	65.44	61.60	59.86	58.62	60.89	59.28	58.09
HSML	71.85	68.73	66.95	70.32	66.62	65.85	60.63	58.58	57.48	60.08	58.07	57.59
KoMen (no taxo.)	72.82	69.64	67.77	71.24	67.65	66.33	61.86	59.53	58.86	62.02	59.20	58.80
KoMen (no expert)	73.17	69.95	68.06	71.40	67.74	66.63	62.01	59.69	58.90	62.39	59.54	59.01
KoMen	**73.74**	**70.69**	**68.84**	**72.46**	**68.76**	**67.59**	**62.63**	**60.49**	**59.47**	**62.92**	**60.15**	**59.47**

method with a multi-task learning method **MoE** [24], which does not contain any scenario-specific parameters or distinguish existing and emerging scenarios. We also use GCN as its base encoder and fine-tune the model on the test support set.

Baselines for the standard supervised setting. Although our focus is interaction recommendation for emerging scenarios, we also want to see how our method affects the performance under existing scenarios. We first compare with *Graph-based methods*, which include **GCN** [13] and **GATNE** [3]. To adapt GCN to our setting, we consider each edge type as an individual graph and make predictions independently. We choose GATNE among all other link prediction methods for multiplex graphs because it is the current state-of-the-art method on the YouTube dataset. We also compare with *Multi-task learning methods*, including **MoE** [24], **MMoE** [17], and **Tree-MMoE** [14]. To be fair, we use GATNE as the graph embedding model for all these methods and add several fully connected layers as top layers.

Ablations. To validate the utility of the domain knowledge and the MoE architecture, we compare KoMen with two ablations that remove the two modules respectively. **KoMen (no taxo.)** substitutes the initial representations $\mathbf{d}_r$ with one-hot vectors. **KoMen (no expert)** forces each scenario to use its own expert during training and initializes the parameters of an emerging scenario by copying the parameters of the existing scenario closest to it in the taxonomy. We also compared to an additional ablation in the Appendix.

Metrics. Following previous work [3], we use ROC-AUC, PR-AUC, and F1 as our evaluation metrics. Since the number of edges may be imbalanced for different edge types, we also introduce weighted ROC-AUC, weighted PR-AUC, and weighted F1 (noted as wROC-AUC, wPR-AUC, and wF1), which sum up the performance on all edge types with weights proportional to their test sample size.

4.2 Main Results

Results under the Few-shot Setting. Table. 2 presents the results under the few-shot setting. The highest score under each metric is highlighted in bold, and the highest score among baselines is underlined. Overall, KoMen consistently yields the best performance among all methods on both the public dataset and real-world industry dataset. For instance, KoMen improves over the best baseline w.r.t. weighted ROC-AUC by relative gains 3.04% and 3.33% on the two datasets, with p-values $\ll 0.01$ in t-test.

Among all the baselines, we observe that meta-learning methods, especially MAML and HSML, significantly outperform basic methods. With the help of meta-optimization by gradient, MAML is able to quickly generalize for an emerging scenario. However, MAML initializes the parameters for all scenarios in the same way, while KoMen customizes the initialization for each scenario. The scenario representations allow similar scenarios to share more information, allowing the model to better use the meta-knowledge.

Although HSML and Meta-graph also use scenario-specific initialization, they customize the initialization in a solely data-driven way, where the limited data for emerging scenarios may not be adequate to represent the generic topological structures. Their weakness is especially serious when the graph is relatively large and sparse. We can observe that HSML beats MAML under all metrics in YouTube, but is significantly outperformed by MAML in Taobao. Both MAML and HSML are outperformed by KoMen, which incorporates domain knowledge into data-driven learning, which captures the scenario similarities better and hence obtains a better scenario-specific initialization. Although MoE is not designed for fast generation to emerging scenarios, MoE (with fine-tune) is also quite competitive in YouTube. This again shows the importance of obtaining a global initialization with high quality.

The comparison between our full model and KoMen (no taxo.) further demonstrates that by combining domain knowledge with data-driven learning, KoMen can capture the similarities among scenarios in a more accurate way and thus give better generalization results. The comparison between KoMen and KoMen (no expert) validates the utility of the MoE architecture, which balances between generalization and customization.

Results under Supervised Setting. Although our method aims to handle interaction recommendation for emerging scenarios, experiments also show that KoMen outperforms the adapted baselines on existing scenarios under the supervised setting. For instance, as shown in table 3, KoMen outperforms the best baseline w.r.t. PR-AUC by 1.68% and 1.46% on the two datasets. We also achieve $p < 0.01$ in t-test under weighted PR-AUC score in both datasets.

Table 3: Experimental results under the supervised setting. Although the major focus of our work is on the few-shot setting, KoMen also achieves the best performance when there is abundant training data.

Methods	*YouTube-existing*			*Taobao-existing*		
	ROC	PR	F1	ROC	PR	F1
GCN	81.52	81.68	74.08	79.78	77.38	72.60
GATNE	84.61	81.93	76.83	81.52	80.44	73.69
MoE	83.55	78.18	76.37	80.04	75.20	70.38
MMoE	84.73	80.77	77.20	79.96	78.91	68.74
Tree-MMoE	83.57	80.75	76.19	80.28	77.52	69.91
G-Meta	75.30	77.50	74.33	75.26	72.39	68.76
Meta-graph	82.83	81.55	74.79	N/A	N/A	N/A
MAML	83.30	80.69	76.21	80.41	80.19	72.32
HSML	81.89	79.37	74.68	79.99	78.86	71.72
KoMen (no taxo.)	85.11	82.85	77.46	81.99	81.35	73.98
KoMen	**85.51**	**83.61**	**77.80**	**82.26**	**81.90**	**74.39**

Among the baselines, meta-learning methods are less competitive compared to the standard supervised methods. They aim to capture the general knowledge shared by all scenarios while not having enough customization. In comparison, KoMen customizes the shared knowledge for each scenario based on both its function and the training data, balancing customization and generalization.

Among the baselines, GATNE, MoE, and MMoE capture the complicated inter-dependencies among scenarios in a purely data-driven way. However, we observe in the datasets that even the existing scenarios follow the long-tail distribution. Namely, a large number of scenarios also have scarce training data, which again easily leads to over-fitting and sub-optimal results. For the same reason, our full model outperforms KoMen (no taxo.) in both datasets. Although Tree-MMoE also employs taxonomy as a form of domain knowledge as we do, it is still outperformed by KoMen. The reason might be Tree-MMoE hard-codes the taxonomy into the model structure, while KoMen encodes the domain knowledge into scenario representations and also updates the representations during training, using the domain knowledge more flexibly.

4.3 Performance Analysis

Performance Breakdown on Different Scenarios. In addition to the overall performance, we also focus on which kind of scenarios (edge types in our formulation) benefit the most from KoMen. We divide the existing 20 edge types in Taobao dataset evenly into three groups by their size, noted as large, medium, and small scenarios. For each group, we compare the relative lift of KoMen and KoMen (no taxo.) over GATNE. The result is shown in Fig. 3.

In all three groups, KoMen outperforms both our ablation and GATNE and the improvement is especially significant for large and small scenarios. One possible explanation is that scenarios with massive records are always related to a large number of scenarios. For instance, "gold coin mission" has 7 siblings under the same parent node "user-interaction in a game". The intuition is that, when a game is proved to be popular, the platform tends to develop

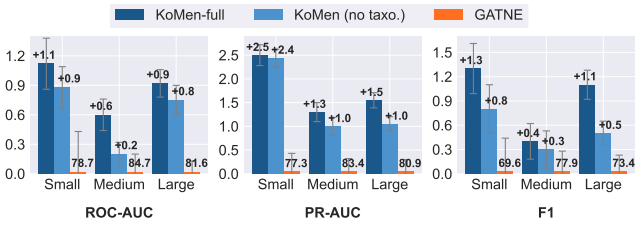


Figure 3: Performance breakdown with three evaluation metrics on the Taobao dataset.

other games to attract more users. By incorporating domain knowledge, KoMen "groups" these scenarios together, and allows them to mutually enhance each other through parameter sharing.

When it comes to small scenarios, the number of training samples is relatively small, hence GATNE may suffer from inadequate training. In KoMen, these small scenarios are able to share the same experts with large ones, which are updated more completely by sufficient training samples. Furthermore, both KoMen and KoMen (no taxo.) are capable to capture the complex inter-dependencies among scenarios to some degree. This enables small scenarios to share more similar parameters with more related ones and vice versa, which further improves the performance.

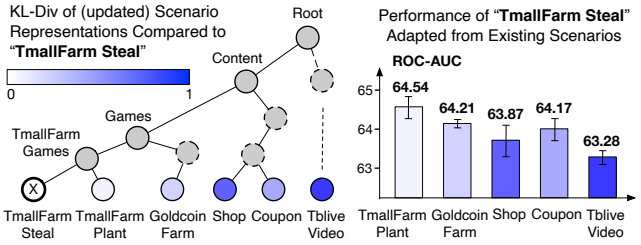


Figure 4: *(Left)* A subtree of the taxonomy and the visualization of similarities between (updated) scenario representations. *(Right)* A case study that empirically reflects the "real" scenario similarities.

Visualization and Case Study. We attempt to understand how the (updated) scenario representations reflect their similarities. Towards this end, we choose "TmallFarm Steal" as an anchor, and visualize the KL-divergence between its representation and other scenarios in Fig. 4. Overall, scenarios with smaller distances in the taxonomy will share relatively similar representations, but there are also exceptions. For instance, although "shop" and "coupon" are equally close to the anchor, the KL-divergence of "coupon" is much smaller. One commonality of "coupon" and "TmallFarm Steal" is that users can get discounts by both sharing coupons and interacting in the games. This shows that KoMen can encode the domain knowledge in (updated) scenario representations, and still allow it to update during training.

To empirically obtain scenario similarities, we train KoMen (no expert) and initialize the emerging scenario "TmallFarm Steal" from the existing scenarios one by one. As shown in Fig. 4, the ROC-AUC adapted from "TmallFarm Plant" is 1.26 higher than that of "Tblive

Video". This validates our motivation that scenarios with similar (updated) representation tend to have similar user behavior, and hence are better to share similar model parameters.

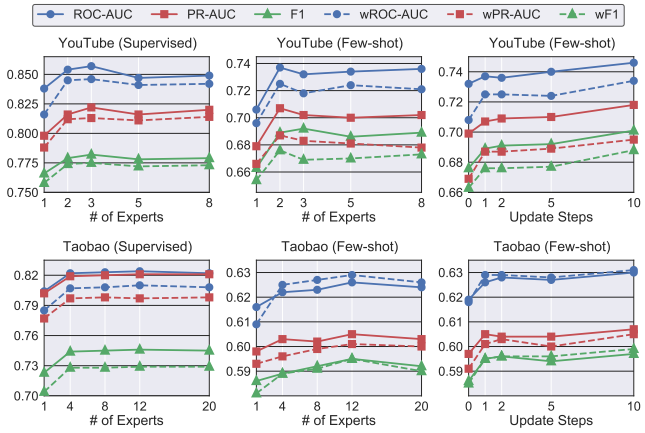


Figure 5: Parameter analysis of number of experts on both existing and emerging scenarios, and the effect of update steps for emerging scenarios. We vary the number of experts from {1, 2, 3, 5, 8} in YouTube and {1, 4, 8, 12, 20} in Taobao, and vary update step t from {0, 1, 2, 5, 10} in both datasets.

4.4 Hyper-Parameter and Efficiency Analysis

The number of experts. Since KoMen is collapsed to MAML when there is only one expert, we are interested in how the number of experts affects the overall performance for both existing scenarios and emerging ones. As shown in Fig. 5, we observe that the optimal number of experts for YouTube is 2 and that for Taobao is 12. The overall performance is increasing as we add more experts to the model at the beginning, but does not differ much around the optimal setting on both datasets.

The number of update steps in meta-learning. We also investigate how the number of adaptation steps t affects the performance of emerging scenarios. We observe that the performance of KoMen increases when we update the parameters more sufficiently. It is generally stable for different values of t, except when $t = 0$, where we do not have adaptation at all.

Efficiency Analysis. We test the convergence time of KoMen with GATNE on a single RTX A6000 GPU. On the Taobao dataset, under both the few-shot and supervised settings, both methods take less than 2 hours to converge, with the best score presents after around 30000 gradient steps. This shows optimizing with meta-learning does not significantly increase the training time of our method.

5 RELATED WORK

In this section, we first review existing methods for interaction recommendation, highlighting the most relevant ones to our work and then briefly summarize the recent progress in few-shot learning.

Interaction Recommendation. Interaction recommendation, or User-User interaction recommendation, aims to predict with whom a user wants to interact in the social network. It can be grouped into three categories: classification, fitting, and ranking [6, 29]. Our method falls into the category of classification, which treats the interaction between users as a binary variable. One major line of studies is graph-based interaction recommendation, which formulates the prediction of user interactions as link prediction on graphs [6, 14, 29, 38, 42]. These studies make use of recent graph representation learning methods such as random walk based methods [22] and graph neural networks [10, 13]. Under our setting that considers diverse scenarios in interaction recommendation, we may refer to multiplex graph representation learning methods, where most of the methods aim to capture complex interactions of different edge types on the same set of nodes by utilizing higher-order graph structures such as meta-paths [7, 25, 33, 35] or assigning different attention to different edge types [3, 43]. Our work falls in the latter category. Although many studies have been conducted on interaction recommendation in fixed scenarios, they hardly consider newly emerging scenarios, which we argue is common and crucial in real e-platforms. Hence, we make the first attempt for few-shot interaction recommendation, which tackles emerging scenarios.

Few-shot Learning. Few-shot learning intends to rapidly generalize to new tasks containing only a few samples [28]. Recent studies on meta-learning are shown to be effective in few-shot learning for various applications [8, 11], which learns to generalize by capturing the general knowledge across similar learning tasks. There are meta-learning methods that explicitly model the relationship between tasks[21, 39, 40]. [39, 40] train a task encoder to capture task similarity and organize the tasks into relational graphs or taxonomies. [1] also learns a low-dimensional embedding for each task. This high-level idea can be well adapted to graph-based meta-learning. [2] designs a graph signature function to encode each graph into low dimensional space. [4] use a relational learner to capture the relatedness between different relations in a knowledge graph. Other graph-based meta-learning methods include [12], which takes advantage of the local topological structure, [5], which especially focuses on node classification, link prediction, and graph classification, and etc [45]. However, most of the few-shot learning methods use pure data-driven approach to capture the relations between tasks, suffering from the scarce data of new tasks. KoMen alleviate this problem by considering and utilizing human's prior knowledge on the purposes and functions of the scenarios before training on the observed data.

6 CONCLUSIONS

In this paper, we formalize the interaction recommendation problem as few-shot link prediction for multiplex graphs and propose a novel framework called KoMen to tackle it. KoMen incorporates domain knowledge into data-driven learning to capture the similarities among scenarios and leverages a mixture-of-expert structure to allow semantically related scenarios to share similar parameters and vice versa. Extensive experiments on a public dataset and a real-world industry dataset demonstrate that KoMen consistently outperforms state-of-the-art baselines adapted to our setting. Further investigation into other forms of domain knowledge or other model structures capturing causal relations between scenarios could inspire a series of meaningful studies in the future.

REFERENCES

[1] Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2Vec: Task Embedding for Meta-Learning. In *ICCV*.

[2] Avishek Joey Bose, Ankit Jain, Piero Molino, and William L. Hamilton. 2019. Meta-Graph: Few shot Link Prediction via Meta Learning. *ArXiv* (2019).

[3] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation Learning for Attributed Multiplex Heterogeneous Network. In *KDD*.

[4] Mingyang Chen, Wen Zhang, Wei Zhang, Qiang Chen, and Huajun Chen. 2019. Meta Relational Learning for Few-Shot Link Prediction in Knowledge Graphs. In *EMNLP*.

[5] Buffelli Davide and Vandin Fabio. 2020. A Meta-Learning Approach for Graph Representation Learning in Multi-Task Settings.

[6] Daizong Ding, Mi Zhang, Shao-Yuan Li, Jie Tang, Xiaotie Chen, and Zhi-Hua Zhou. 2017. BayDNN: Friend Recommendation with Bayesian Personalized Ranking Deep Neural Network *(CIKM)*.

[7] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *KDD*.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*.

[9] Michael U. Gutmann and Aapo Hyvärinen. 2012. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *JMLR* (2012).

[10] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NeurIPS*.

[11] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2020. Meta-Learning in Neural Networks: A Survey.

[12] Kexin Huang and Marinka Zitnik. 2020. Graph Meta Learning via Local Subgraphs. In *NeurIPS*.

[13] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[14] Bofang Li, Chenliang Li, Xichuan Niu, Jun Tan, Rong Xiao, Haochuan Sun, and Hongbo Deng. 2020. Heterogeneous Graph Augmented Multi-Scenario Sharing Recommendation with Tree-Guided Expert Networks. In *WSDM*.

[15] Dongze Lian, Yin Zheng, Yintao Xu, Yanxiong Lu, Leyu Lin, Peilin Zhao, Junzhou Huang, and Shenghua Gao. 2020. Towards Fast Adaptation of Neural Architectures with Meta Learning. In *ICLR*.

[16] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. arXiv:1703.03130

[17] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. 2018. Modeling Task Relationships in Multi-Task Learning with Multi-Gate Mixture-of-Experts. In *KDD*.

[18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *NeurIPS*.

[19] Maximilian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *NeurIPS*.

[20] Chanyoung Park, Carl Yang, Qi Zhu, Donghyun Kim, Hwanjo Yu, and Jiawei Han. 2020. Unsupervised Differentiable Multi-aspect Network Embedding. In *KDD*.

[21] Massimiliano Patacchiola, Jack Turner, Elliot J. Crowley, and Amos Storkey. 2020. Bayesian Meta-Learning for the Few-Shot Setting via Deep Kernels. In *NeurIPS*.

[22] Bryan Perozzi, Rami Al-Rfou, and S. Skiena. 2014. DeepWalk: online learning of social representations. In *KDD*.

[23] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. 2020. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. In *ICLR*.

[24] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *ICLR*.

[25] Jinghan Shi, Houye Ji, Chuan Shi, Xiao Wang, Zhiqiang Zhang, and Jun Zhou. 2020. Heterogeneous Graph Neural Network for Recommendation.

[26] Lei Tang and Huan Liu. 2009. Uncovering Cross-Dimension Group Structures in Multi-Dimensional Networks.

[27] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. 2020. Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples.

[28] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a Few Examples: A Survey on Few-Shot Learning. *ACM Comput. Surv.* (2020).

[29] Zhibo Wang, Jilong Liao, Qing Cao, H. Qi, and Zhi Wang. 2015. Friendbook: A Semantic-Based Friend Recommendation System for Social Networks. *IEEE Transactions on Mobile Computing* (2015).

[30] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. 2017. Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation. In *KDD*.

[31] Carl Yang and Kevin Chang. 2019. Relationship Profiling over Social Networks: Reverse Smoothness from Similarity to Closeness. In *SDM*.

[32] Carl Yang, Yichen Feng, Pan Li, Yu Shi, and Jiawei Han. 2018. Meta-Graph Based HIN Spectral Embedding: Methods, Analyses, and Insights. In *ICDM*.

[33] Carl Yang, Mengxiong Liu, Frank He, Xikun Zhang, Jian Peng, and Jiawei Han. 2018. Similarity Modeling on Heterogeneous Networks via Automatic Path Discovery. In *ECML-PKDD*.

[34] Carl Yang, Aditya Pal, Andrew Zhai, Nikil Pancha, Jiawei Han, Chuck Rosenberg, and Jure Leskovec. 2020. MultiSage: Empowering GraphSage with Contextualized Multi-Embedding on Web-Scale Multipartite Networks. In *KDD*.

[35] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. 2020. Heterogeneous Network Representation Learning: A Unified Framework with Survey and Benchmark. In *TKDE*.

[36] Carl Yang, Jieyu Zhang, and Jiawei Han. 2020. Co-Embedding Network Nodes and Hierarchical Labels with Taxonomy Based Generative Adversarial Nets. In *ICDM*.

[37] Carl Yang, Jieyu Zhang, Haonan Wang, Sha Li, Myungwan Kim, Matt Walker, Yiou Xiao, and Jiawei Han. 2020. Relation Learning on Social Networks with Multi-Modal Graph Edge Variational Autoencoders. In *WSDM*.

[38] Carl Yang, Lin Zhong, Li-Jia Li, and Luo Jie. 2017. Bi-directional joint inference for user links and attributes on large social graphs. In *WWW*.

[39] Huaxiu Yao, Y. Wei, Junzhou Huang, and Z. Li. 2019. Hierarchically Structured Meta-learning. In *ICML*.

[40] Huaxiu Yao, Ying Wei, Junzhou Huang, and Li Zhenhui. 2020. Automated Relational Meta-learning. In *ICLR*.

[41] Huaxiu Yao, Chuxu Zhang, Ying Wei, Meng Jiang, Suhang Wang, Junzhou Huang, Nitesh Chawla, and Zhenhui Li. 2020. Graph Few-Shot Learning via Knowledge Transfer. (2020).

[42] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *KDD*.

[43] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous Graph Neural Network. In *KDD*.

[44] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. 2018. Scalable Multiplex Network Embedding. In *IJCAI*.

[45] Fan Zhou, Chengtai Cao, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Ji Geng. 2019. Meta-GNN: On Few-shot Node Classification in Graph Meta-learning.

[46] Qi Zhu, Carl Yang, Yidan Xu, Haonan Wang, Chao Zhang, and Jiawei Han. 2021. Transfer Learning of Graph Neural Networks with Ego-graph Information Maximization. In *NeurIPS*.

A APPENDIX

In the appendix, we will first give a more detailed introduction to the datasets we use in our paper. Then we will introduce the implementation details of our model and baselines.

A.1 Dataset Details

As no other work focus on few-shot friend recommendation, or in our formulation, few-shot link prediction on multiplex graphs, we construct a dataset called YouTube from a public supervised multiplex graph link prediction dataset and construct a real-world dataset called Taobao by collecting data from a leading e-commerce platform.

The YouTube dataset is adapted from [26] by splitting its 5 edge types into 2 meta-training graphs, 1 meta-validation graph, and 2 meta-testing graphs. Under the few-shot setting, following the conventional setting on few-shot link prediction [2, 12], we randomly split 30% of the edges in meta-validation and meta-testing graphs as support set and others as query set.

The Taobao dataset is collected from the user daily sharing logs of a leading e-commerce platform accumulated during July 2020. Once a user wants to share something with others, the system will recommend a list of potential target users. The clicked ones are recorded as positive examples while the exposed but not clicked are negative samples. We take the 5 scenarios with the fewest records as meta-testing graphs and the others as meta-training graphs.

To evaluate KoMen's ability to use domain knowledge, for each dataset, we organize the edge types into a taxonomy by their purposes and functions.

A.2 Implementation Details

We compare the experiment result after training for 30 epochs of all methods on YouTube, and 1 epoch (or around 30000 gradient steps) on the Taobao dataset.

For datasets with node features (Taobao), we set the size of each feature embedding as 16 for each feature and keep it updated during training. We construct the node embedding of each user by concatenating the feature embeddings and pass it through a linear layer. For datasets without node features (YouTube), we train a randomly-initialized node embedding for each node.

We keep the embedding dimension as 200 for all the methods. For all the methods using GATNE as base model (GATNE, MoE, MMoE, Tree-MMoE, MAML, HSMl, Ours), we keep the related hyper-parameters as the same: (edge-dim=10, att-dim=20, negative-samples=5, walk-length=6, num-walks=5, window-size=3, neighbor-samples=5) for Taobao, and (edge-dim=10, att-dim=20, negative-samples=5, walk-length=10, num-walks=20, window-size=5, neighbor-samples=10) for YouTube.

We tune the learning-rate and batch-size for all the methods. For none meta-learning methods, the search space is lr = 1e-3, 1e-4, batch-size = 16, 32, 64, 128 For meta-learning methods (G-Meta, Meta-graph, MAML, HSML, ours), we tune the learning rate of both inner loop and outer loop, noted as meta-lr = 1e-3, 1e-4 and update-lr = 1e-2, 1e-3, the batch size of both support set and query set, noted as spt-batch-size = 16, 32, 64, 128 and qry-batch-size = 32, 64, 128, 256, and the number of tasks in each batch (task-batch-size = 1,2,4,8).

For the methods that use GATNE as base model, we also tune alpha = 0.1, 0.5, 1.0, 2.0, the coefficient between base embedding and edge-type embedding.

A.3 Additional Ablation Study

In this section, we compare with an additional ablation of our method: KoMen (no meta.), where we remove the meta-learning component and train the model on the training data of both the existing and emerging scenarios.

We observe that the performance of our full model is much better when the model only make one update on each training sample of emerging scenarios. This is also a more realistic setting because when a new scenario emerges, we already have a model converged on existing scenarios.

When we fine-tune the model until it converges, the performance gap between the two methods decreases but our full model still outperforms the (no meta.) ablation. The reason might be in (no meta.), the model parameters are mainly trained to fit the existing scenarios with abundant data, which may not be the optimal for emerging scenarios.

Table 4: Comparison between KoMen and its ablation: KoMen (no meta.)

YouTube-new (one adaption step)						
Method	ROC	PR	F1	wROC	wPR	wF1
KoMen (no meta.)	53.21	52.55	52.37	54.24	53.38	53.05
KoMen	**73.74**	**70.69**	**68.84**	**72.46**	**68.76**	**67.59**
YouTube-new (finetune until converge)						
Method	ROC	PR	F1	wROC	wPR	wF1
KoMen (no meta.)	74.54	71.30	68.41	72.40	68.72	66.55
KoMen	**75.02**	**71.95**	**69.27**	**73.57**	**69.97**	**68.16**