Few-Shot Link Prediction for Event-Based Social Networks via Meta-learning

Xi Zhu[1], Pengfei Luo[1], Ziwei Zhao[1], Tong Xu[1(✉)], Aakas Lizhiyu[2], Yu Yu[2], Xueying Li[2], and Enhong Chen[1]

[1] University of Science and Technology of China, Hefei, China
{xizhu,pfluo,zzw22222}@mail.ustc.edu.cn, {tongxu,cheneh}@ustc.edu.cn
[2] Alibaba Group, Hangzhou, China
{aakas.lzy,greta.yy,xiaoming.lxy}@alibaba-inc.com

Abstract. With the thriving of social network analysis, large efforts have been made on link prediction for event-based social networks (EBSNs). Unfortunately, since society is evolving with constantly emerging social events, it is extremely difficult to accurately capture their semantics and evolution rules at an early stage. Meanwhile, traditional solutions require extensive training from scratch to accommodate new events, leading to lagging predictions and high maintenance costs. To tackle these challenges, we investigate this cross-network few-shot problem and propose a novel meta-learning model for link prediction on new EBSNs. To accurately simulate the few-shot scenarios, we first utilize existing EBSNs to define a task distribution that augments the new event with other observed events. Specifically, we define a unified and generalized target event to be transferred as the few-shot event. Then, we empower a simple but effective event-aware graph attention network to encode existing fine-grained events and the few-shot target events. Furthermore, we follow gradient-based episode learning to obtain transferable knowledge and adapt to unseen EBSNs with sparse connections. Finally, extensive experiments on both public and industrial datasets have demonstrated the performance of fast adaption and even overall performance.

Keywords: few-shot · event-based social networks · meta-learning

1 Introduction

The widespread popularity of social services has brought us a tremendous volume of social interaction data and various spontaneously formed communities. Over the last decade, service providers are motivated to create and promote a series of interest-driven social events to improve information dissemination and further increase the vitality of platforms. Along this line, the concept of **Event-Based**

X. Wang et al. (Eds.): DASFAA 2023, LNCS 13945, pp. 31–41, 2023.
https://doi.org/10.1007/978-3-031-30675-4_3

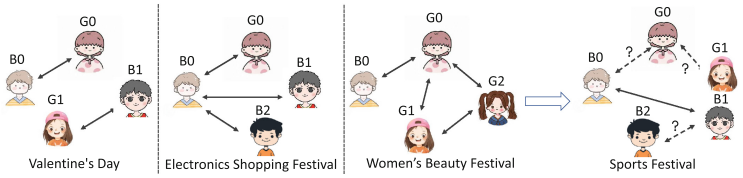

Fig. 1. An example of overlapping EBSNs generated by product share records on an e-commerce platform, where B and G denote boys and girls, respectively.

Social Networks (EBSNs) has emerged which provides explicit fine-grained information to show the diverse social preferences of users.

With the prosperity of social services, society is evolving with successive social events. Taking Fig. 1 as an example, various social-oriented promotions are launched to promote communication within and outside communities. Specifically, couples (e.g., B0&G0, B1&G1) kept in touch on Valentine's Day. The girls (e.g., G0, G1, and G2) would share beauty products, while boys (e.g., B0, B1, and B2) would connect during Electronics Shopping Festival. Notably, the interaction (B0, G0) always holds, exhibiting an event-agnostic relationship, while most users show various social patterns. However, with the advent of the Sports Festival, there is an urgent need to identify and attract users to establish connections in the corresponding EBSN, thus we have to quickly find potential links on the sparsely-estimated network. This motivates us to consider a more challenging setting of graph few-shot learning, which expects to enable fast adaption and high-quality prediction on newly-deployed EBSNs without abundant data. We emphasize that an effective early prediction of the evolution of the novel community will provide indispensable guidance to assist decision-making.

In this paper, we study few-shot link prediction for EBSNs. Intuitively, due to the cold-start problem, new EBSNs would suffer from data deficiency and even distorted topology [1]. A straightforward idea is to apply a multi-relation model to encode all events [6,8] and retrain it to accommodate new events, which results in lagging predictions and high maintenance costs. Recent years have witnessed the rapid development of few-shot learning (FSL) [1,2,4,10]. As a prevailing paradigm, MAML [2] obtains knowledge from similar tasks and transfers it to unseen tasks with a few instances. We notice that the key is to perfectly simulate the few-shot scenarios, which guarantees the transferability and robustness of the meta-knowledge. Specifically, we have the following observations:

– **The dependencies among EBSNs could be utilized to enhance the semantics and structures of the few-shot event.** Prior arts of FSL on disjoint attributed networks [1,4] are not compatible with densely connected EBSNs from the same domain. Since emerging EBSNs can be noisy, sparse, and unbalanced in scale, previous interactions could be naturally utilized. Correspondingly, we have the challenge to define a task distribution over dense EBSNs, further contributing to knowledge transfer.

– **The shared characteristics of few-shot events could be learned and transferred to guarantee generalizability.** Indeed, the exact semantics of constantly emerging events are difficult to capture [1]. However, the semantics of existing events can be explicitly captured and naturally transferred. Thus, we expect a unified well-initialized representation for the few-shot event as a good starting point for fine-tuning. Correspondingly, we have the challenge to design an effective model as the carrier of meta-knowledge.

We propose a meta-learning framework to address the aforementioned challenges. Briefly, new sparse target events leverage the knowledge from existing dense events, as there are shared semantics and structures worth exploiting. First, to simulate few-shot scenarios, we fully utilize existing EBSNs to define a task distribution, where a cross-network sampling strategy is designed to handle interconnected EBSNs. Afterward, a unified and generalizable target event is proposed to simulate the few-shot event. For each task, fine-grained source events and the special target event are jointly encoded by an event-aware graph attention network (EA-GAT), where both target and auxiliary loss are jointly optimized. Overall, we follow the standard episode learning to learn well-initialized parameters from tasks. In this case, when new events appear, we can individually fine-tune the tasks with a handful of associative instances, and adapt quickly with few resources. Our contribution can be summarized as follows: (1) To the best of our knowledge, we are the first to investigate FSL on interconnected EBSNs, which is universal and significant for early decision-making. (2) To achieve fast adaption, we propose a meta-learning framework for knowledge transfer. To simulate the few-shot scenario, we define a task distribution with network augmentation and learn a unified, generalizable few-shot event. (3) To validate the effectiveness, experiments are conducted on public and industrial datasets to show the superiority of fast adaption and convergence performance.

2 Related Works

Graph Neural Networks (GNNs). Recently, GNNs have been widely adopted to preserve properties and structures on graphs, such as GCN, GAT, and GraphSAGE. Moreover, R-GCN [6] and CompGCN [8] are proposed to model graph heterogeneity. HGT [3] decomposes the interactions with a Transformer-like architecture. However, these approaches learn global parameters with abundant data while we focus on emerging events (relations) with insufficient data.

Graph Few-Shot Learning. Remarkable success has been made on FSL of images and text while the exploration of graphs is still in its infancy, especially in multi-graph settings. Some studies formulate the transferable knowledge as meta-optimizer and metric space, e.g., Prototypical Network [7]. By contrast, Meta-GNN [10] integrates MAML with GNNs and facilitates gradient descent across tasks. However, little literature can be adapted to few-shot link prediction.

Algorithm 1. Meta-Training Task Sampling

Input: $\mathcal{R} = \{R_1, ..., R_N\}, \mathcal{G} = \{G_1, ..., G_N\}$ where $G_i = \{V_i, E_i\}$
Output: The distribution of meta-training tasks $p(\mathcal{T})$

while not done **do**
  Select $R_k \in \mathcal{R}$;
  Set R_k to R_{tgt}, Define $\mathcal{R}_{aux} = \mathcal{R} - \{R_k\}$;
  Split $E_{tgt} = E_{tgt}^{support} \cup E_{tgt}^{query}$;
  Construct $G = (V, E)$, where $E = E_{tgt}^{support} \cup E_{aux}$ and $E_{aux} = \bigcup_{i \neq k} E_i$;
  Compose $\mathcal{S} = (\mathcal{S}_{tgt}, \mathcal{S}_{aux})$ from $E_{tgt}^{support}$ and E_{aux}, respectively;
  Compose $\mathcal{Q} = (\mathcal{Q}_{tgt}, \mathcal{Q}_{aux})$ from E_{tgt}^{query} and E_{aux}, respectively;
  $T = (G, \mathcal{S}, \mathcal{Q})$
end while

To list a few, G-Meta [4] proposes to reduce parameters by encoding local subgraphs for large-scale graphs. Meta-Graph [1] recovers graphs with a variational auto-encoder and learns knowledge from disjoint attributed graphs. Despite the progress, these works fail to handle the cross-network dependency for EBSNs.

3 Problem Definition

Definition 1. ***Event-Based Social Networks (EBSNs).*** *Suppose the user set is denoted as $\mathcal{V}$ and there have been N events $\mathcal{R} = \{R_1, R_2, ..., R_N\}$. The corresponding EBSNs are denoted as $\mathcal{G} = \{G_1, G_2, ..., G_N\}$, where the i-th EBSN is represented as $G_i = (V_i, E_i)$, where $V_i \subseteq \mathcal{V}$ and $E_i = \{(u, R_i, v) | u, v \in V_i\}$.*

Definition 2. ***Few-shot Link Prediction for EBSNs.*** *Assume we have observed the social events $\mathcal{R} = \{R_1, R_2, ..., R_N\}$, and the corresponding EBSNs are $\mathcal{G} = \{G_1, G_2, ..., G_N\}$. For an emerging few-shot event R_{few}, and $G_{few} = (V_{few}, E_{few})$, $V_{few} \subseteq \mathcal{V}$. We follow the few-shot setting and split $E_{few} = E_{few}^{support} \cup E_{few}^{query}$, where $E_{few}^{support} \cap E_{few}^{query} = \emptyset$. As a small fraction of interactions are available to support the network inference, i.e. $|E_{few}^{support}| \ll |E_{few}|$, our goal is to predict E_{few}^{query} with limited true edges from $E_{few}^{support}$.*

4 Methodology

In this section, we present the technical details of our model in Fig. 2, including task sampling, event-aware link prediction task, and meta-learning framework.

4.1 Cross-Network Task Sampling

The meta-learning approaches assume there are exploitable, shareable structures across similar tasks. Thus, its success relies heavily on making full use of existing data to create tasks that delicately simulate real-world few-shot scenarios. Specifically, when new events emerge, we enhance the new, sparse, and noisy

EBSN with previous, dense, and complete EBSNs, denoted as auxiliary and target events, respectively. To this end, we leverage existing EBSNs to define a task distribution. Taking Fig. 1 as an example, we possibly choose Women's Beauty Festival as the **target event (few-shot event)**, so the Valentine's Day and Electronics Shopping Festival become **auxiliary events**. Note any of them can be selected as the target event to generate sufficient tasks.

Unfortunately, the support samples are insufficient to express the semantic meaning of the target event. Therefore, we propose a unified, generalizable target event and attempt to make it transferable to unseen few-shot events, i.e. transfer R_{tgt} to R_{few}. In other words, the learnable target event shared across tasks will be adopted to represent the real emerging few-shot event in meta-testing.

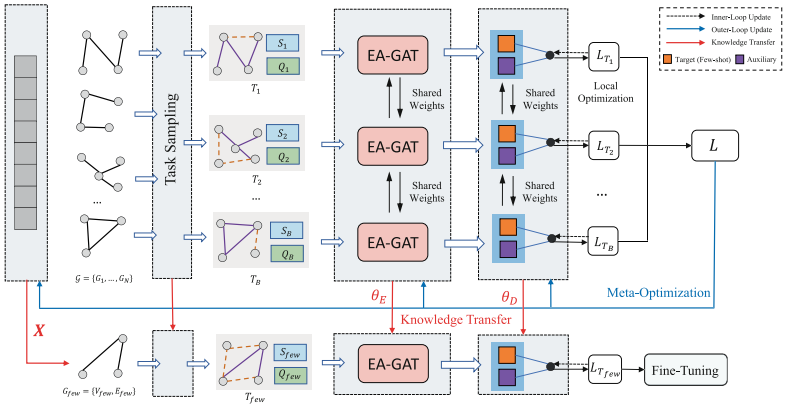

Fig. 2. The proposed meta-learning framework including (1) Few-shot task sampling with network augmentation, (2) EA-GATs, and (3) Joint learning for link prediction. Without loss of generality, we sample B meta-training tasks as a batch.

Meta-Training Tasks. As illustrated in Algorithm 1, we first randomly select R_k from $\mathcal{R} = \{R_1, ..., R_N\}$ as the target event, and set the others are auxiliary events, i.e. $R_{tgt} = R_k, R_{aux} = \mathcal{R} - \{R_k\}$. Then, following the few-shot setting, we allow $G_{tgt} = \{V_{tgt}, E_{tgt}\}$ to be divided into $E_{tgt} = E_{tgt}^{support} \cup E_{tgt}^{query}$, where $E_{tgt} = E_{tgt}^{support} \cap E_{tgt}^{query} = \emptyset$. Next, we utilize E_{aux} to augment the sparse $E_{tgt}^{support}$ and build $G = (V, E)$, where $E = E_{tgt}^{support} \cup E_{aux}$, $E_{aux} = \bigcup_{i \neq k} E_i$. Afterwards, we dynamically sample edges from $E_{tgt}^{support}$ and E_{aux} and compose the support set as $\mathcal{S} = (\mathcal{S}_{tgt}, \mathcal{S}_{aux})$. Similarly, we compose the query set $\mathcal{Q} = (\mathcal{Q}_{tgt}, \mathcal{Q}_{aux})$ from E_{tgt}^{query} and E_{aux}. So far, we have defined a task as $T = (G, \mathcal{S}, \mathcal{Q})$. e can repeat the above steps to sample batches of training tasks. Finally, we target at making prediction on $\mathcal{Q}_{tgt}$ ($\mathcal{Q}_{aux}$ only assists the training) with limited $\mathcal{S}_{tgt}$ to support the inference.

Meta-Testing Tasks. Once event R_{few} comes, we follow the above steps to create a meta-testing task T_{few} in a similar way. Differently, all existing events $\mathcal{R} = \{R_1, ..., R_N\}$ could be auxiliary events as their semantics have been accurately captured. Since R_{tgt} exactly simulates the few-shot event, the prior knowledge of R_{tgt} will be naturally transferred to R_{few} for fine-tuning.

4.2 Event-Aware Link Prediction Task

Event-Aware Graph Attention Networks (EA-GATs). We first denote a generic encoder $\boldsymbol{H}^G = f_E(G; \boldsymbol{X}, \theta_E)$ parameterized by θ_E. $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is learnable user embeddings. To incorporate the unified target event, $\tilde{\mathcal{R}} = \mathcal{R} \cup \{R_{tgt}\}$ is involved in the augmented network $G = (V, E)$. Generally, we set $\boldsymbol{X} = \boldsymbol{h}^{(1)}$ and iteratively update the embedding of user u as follows:

$$\boldsymbol{h}_u^{(l)} = \sigma\left(\sum_{v \in \mathcal{N}_u} \alpha_{uv}^{(l)} \boldsymbol{W}_a^{(l)} \boldsymbol{h}_v^{(l-1)}\right) \tag{1}$$

where $\alpha_{uv}^{(l)}$ is the attention weight between u and v, $\boldsymbol{W}_a^{(l)}$ is the weight matrix. Following [5], to handle event semantics, we allocate a d_r-dimensional embedding $\boldsymbol{z}_r$ for each event $r \in \tilde{R}$. As multiple events (relations) may exist between any pair of users, suppose the co-occurred events are $\varphi(u, v)$, we accumulate all contributions and compute the raw attention with event-specific representation:

$$\hat{\alpha_{uv}} = \frac{\sum_{r \in \varphi(u,v)} exp(\boldsymbol{a}^T g(\boldsymbol{W}\boldsymbol{h_u} || \boldsymbol{W}\boldsymbol{h_v} || \boldsymbol{W_r}\boldsymbol{z_r}))}{\sum_{k \in \mathcal{N}_u} \sum_{r \in \varphi(u,k)} exp(\boldsymbol{a}^T g(\boldsymbol{W}\boldsymbol{h_u} || \boldsymbol{W}\boldsymbol{h_k} || \boldsymbol{W_r}\boldsymbol{z_r}))} \tag{2}$$

where $\boldsymbol{W} \in \mathbb{R}^{d \times d}$ and $\boldsymbol{W}_r \in \mathbb{R}^{d \times d_r}$ are node- and relation-type transformation matrices, and we omit superscript l for simplicity. $\boldsymbol{a}^T \in \mathbb{R}^{3d}$ is a weight vector. $\mathcal{N}_u$ denotes the neighbors of u and $g(\cdot)$ denotes LeakyRELU. Following HGB [5], with residual connections on attention scores, the final attention score is $\alpha_{uv}^{(l)} = \beta \alpha_{uv}^{(l-1)} + (1 - \gamma)\hat{\alpha_{uv}}^{(l)}$, where γ is a scaling factor, i.e., 0.05. Besides, we adopt multi-head attention mechanism to learn from different subspaces. To capture long-range dependency, we stack L layers and use layer normalization to stabilize the training process. The final embeddings are $\boldsymbol{h}_u^g = Pooling\left(\boldsymbol{h}_u^{(1)}, \boldsymbol{h}_u^{(2)}, ..., \boldsymbol{h}_u^{(L)}\right) \in \mathbb{R}^{d_g}$ with concatenation or mean pooling.

Decoder. We define a generic decoder as $\mathbf{Y} = f_D(\mathcal{A}; \theta_D)$ to decode the triples in support or query set, i.e. $\mathcal{A} \in \{S, Q\}$. Inspired by [9], we instantiate the decoder with bilinear score function with $\boldsymbol{W}' \in \mathbb{R}^{(N+1) \times d_g \times d_g}$. Thus for a triple $(u, r, v) \in \mathcal{A}$, the probability y_{uv}^r that u and v holds in the event $r \in \tilde{R}$ is:

$$y_{uv}^r = \sigma\left(\boldsymbol{h}_u^g \boldsymbol{W}'_r \boldsymbol{h}_v^g\right) \tag{3}$$

where $\boldsymbol{W}'_r \in \mathbb{R}^{d_g \times d_g}$ is the event-aware matrix of r indexed from $\boldsymbol{W}'$.

Joint Learning with Auxiliary Triples. Given candidate triple set $\mathcal{A} = \{(u, r, v)\}$, the binary cross-entropy loss can be written as:

$$\mathcal{L}(\mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum_{(u,r,v)\in\mathcal{A}} y_{uv}^r \log(\hat{y}_{uv}^r) + (1 - y_{uv}^r) \log(1 - \hat{y}_{uv}^r), \mathcal{A} \in \{\mathcal{S}, \mathcal{Q}\} \quad (4)$$

where y_{uv}^r is the ground-truth. Due to the overlapping between observed and few-shot EBSNs, the reconstruction of auxiliary links will be helpful for prediction on new EBSNs. Hence, for $\mathcal{S} = \{\mathcal{S}_{tgt}, \mathcal{S}_{aux}\}$, the task-oriented loss is:

$$\mathcal{L}_{tgt} = \mathcal{L}(\mathcal{S}_{tgt}), \mathcal{L}_{aux} = \mathcal{L}(\mathcal{S}_{aux}), \mathcal{L} = \mathcal{L}_{tgt} + \lambda_0 \mathcal{L}_{aux} + \lambda_1 \|\theta\| \quad (5)$$

where λ_0 is the trade-off parameter and θ denotes all task-level parameters.

4.3 Meta-learning Framework for EBSNs

Inspired by model-agnostic MAML [2], for emerging events, we learn general-purpose parameters from meta-training tasks as the prior knowledge, so that the model could produce good fine-tuning results through a few gradient steps.

Table 1. Statistics of the datasets for the proposed model

Dataset	# events	# nodes	# connections	# pairs	Event Type
DBLP	11	37947	210260	183040	conference
Tmall	8	16961	48782	41186	promotion

Formally, we consider the link prediction model as a function f_θ with $\theta = \{\theta_E, \theta_D, \boldsymbol{X}\}$. When adapting to $\mathcal{T}_i = \{G_i, \mathcal{S}_i, \mathcal{Q}_i\}$ from $p(\mathcal{T})$, we first update the task-level parameters with feeding $\mathcal{S}_i$, which can be expressed as:

$$\theta_i' \leftarrow \theta_i - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta) \quad (6)$$

We only perform a one-step update here while extending to multiple steps (e.g., $K = 5, 10, 20$) is straightforward. For each meta-training task, we perform gradient descents individually. For the best performance of f_θ with respect to θ across tasks from $p(\mathcal{T})$, we validate each task $\mathcal{T}_i$ with $\mathcal{Q}_i$, so that the meta-objective is to minimize the accumulated loss on queries across sampled tasks:

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)}\right) \quad (7)$$

Formally, we optimize meta-parameters θ as follows:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'}) \quad (8)$$

where α, β is the meta-level and task-level learning rate, respectively.

5 Experiments

5.1 Experiment Setup

Data Description. We validate the proposed model on both public and industrial datasets. As summarized in Table 1, **(1) DBLP**[1] is a synthetic dataset, where we select 11 top AI conferences as events, and build EBSNs by connecting the first and other authors of each paper in a pairwise manner as social links. **(2) Tmall**[2] is a real-world dataset collected from 8 category-aware e-commerce promotions. Each social link corresponds to a product-sharing record during the promotion. Users with only one neighbor are eliminated.

Reproducibility Settings. We use PyTorch to implement our model[3]. To avoid accidental deviation caused by different event splitting, we use 5-fold cross-validation and report the mean value of each metric. Following [1,4], we hold a small percentage of true edges as the support set, i.e. $\{10\%, 20\%, 30\%\}$, 10% for validation and the rest for testing. 10 gradient update steps in meta-training and 20 steps in meta-testing are adopted. AdamW and SGD are applied for meta-optimization and task-level optimization, respectively, while their learning rates are turned in $\{1e^{-3}, 1e^{-2}, 1e^{-1}\}$. As for the EA-GAT module, the dimension is fixed to 32. The number of layers is selected in $\{1, 2, 3\}$, and the number of attention heads is tuned in $\{1, 2, 4\}$. As for the task-level loss, λ_0 is tuned in $\{0.2, 0.4, 0.6, 0.8, 1.0\}$, and λ_1 is set to $1e^{-4}$. The batch sizes of auxiliary and target triples are set to 512 and 2048. We tune hyperparameters by grid search and use AUC, Average Precision (AP), and Accuracy as the evaluation metrics.

Table 2. The convergence performance on both DBLP and Tmall datasets. '-' means the metrics are not reported in the original implementation. Best results in **bold**.

Models	DBLP									Tmall								
	30%			20%			10%			30%			20%			10%		
	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc	AUC	AP	Acc
GCN	71.9	71.9	67.7	71.2	72.9	67.2	69.5	70.9	66.0	72.9	77.5	70.8	71.5	76.6	69.4	70.4	74.9	68.7
GAT	75.0	75.3	68.8	74.6	76.4	68.5	72.5	75.1	67.0	76.6	80.8	70.8	75.6	79.7	70.1	74.8	78.9	69.3
GraphSAGE	76.2	78.8	70.4	74.1	77.0	68.8	72.1	75.6	67.6	76.3	81.1	71.3	75.5	80.4	70.6	74.6	79.5	69.4
R-GCN	70.2	72.4	66.7	69.4	70.4	65.8	67.5	69.2	64.2	60.0	64.8	60.4	57.6	62.5	58.6	56.8	61.3	57.7
CompGCN	77.5	79.8	67.4	76.9	79.4	66.1	74.6	76.9	64.5	76.4	80.9	68.9	74.7	79.2	68.0	74.5	79.0	67.0
HGB	76.7	77.7	69.5	74.9	76.3	67.9	73.6	74.4	66.9	74.8	80.1	69.6	72.4	78.4	66.2	72.0	76.4	65.9
MAML	71.8	73.8	64.2	68.7	71.8	61.6	65.7	69.1	59.6	71.9	75.8	67.0	67.5	70.1	62.5	63.5	66.8	58.6
Meta-Graph	77.2	79.1	-	76.2	78.6	-	73.9	75.7	-	75.8	80.1	-	73.4	78.9	-	73.0	77.0	-
Ours	**81.3**	**84.7**	**72.6**	**78.6**	**82.1**	**70.9**	**75.3**	**79.3**	**69.1**	**80.6**	**84.8**	**74.7**	**79.2**	**83.7**	**73.9**	**78.2**	**82.6**	**72.1**

[1] https://dblp.org/.
[2] https://www.tmall.com/.
[3] https://github.com/xizhu1022/FSLP-EBSNs.

Baselines. We compare our method against three categories of baselines: **(1) Single-relation models** operated on homogeneous networks, such as GCN, GAT, and GraphSAGE. **(2) Multi-relation models** that incorporate relation learning, including R-GCN [6], CompGCN [8] and HGB [5]. **(3) Meta-learning models.** Since little literature is applicable to link prediction, we adopt MAML [2] and the closest multi-network FSL work Meta-Graph [1] as baselines. For Meta-Graph, we use GraphSAGE pre-training embeddings as node attributes.

5.2 Experiments Results

Overall Convergence Performance. As shown in Table 2, we evaluate the proposed model against various baselines for final convergence. Here are three findings. First, our model outperforms other models on both datasets with steady improvement in all few-shot settings. For example, as for DBLP in the 30% setting, the absolute gains reach 3.81%/4.97%/2.16% for AUC/AP/Acc, which illustrates the effectiveness of our work. Second, Meta-Graph is slightly inferior to our model probably due to its individual nature and constant attributes. Third, homogeneous models show competitive results, in some cases, even outperform multi-relation models. We argue relations with many samples mistakenly dominate the training process and impair the relation learning with insufficient data.

Table 3. The AUCs of different variants with 20-step finetuning. RD-B indicates the relative decrease w.r.t. the convergence result of its backbone model (denoted in the bracket). RD-O is the relative decrease w.r.t. the best convergence result of our model.

Variants	10%			20%			30%		
	Full	RD-B	RD-O	Full	RD-B	RD-O	Full	RD-B	RD-O
Direct Training (CompGCN)	50.34	-32.52%	-33.12%	50.96	-33.68%	-35.13%	51.11	-34.03%	-37.12%
Train&Finetune (CompGCN)	73.07	-2.05%	-2.92%	73.03	-4.98%	-7.06%	74.73	-3.54%	-8.06%
MAML (GAT)	65.46	-0.39%	-13.3%	67.93	-1.09%	-13.54%	69.17	-3.62%	-14.90%
Ours (EA-GAT)	74.65	-0.83%	-0.83%	76.89	-2.14%	-2.14%	78.87	-2.97%	-2.97%

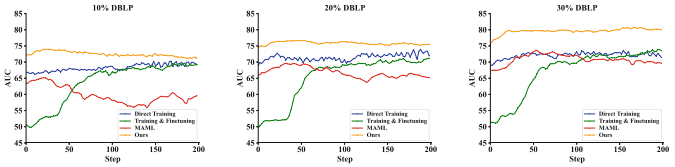

Fig. 3. The AUC curves for convergence of DBLP dataset (200-step finetuning).

Fast Adaption and Convergence Curves. We design the following variants to show the efficiency of rapid learning: **(1) Direct Training** directly trains CompGCN [8], which is among the best traditional multi-relational models (see Table 2). **(2) Training&Finetuning** first extensively trains CompGCN [8] based on existing events, then finetunes with additional few-shot instances. **(3) MAML** is a pure meta-learning method with pure GAT as the meta-model.

Fast Adaption Performance. As shown in Table 3, MAML and our model perform very closely to their convergence results with limited training (see RD-B), which illustrates the remarkable rapid adaptability of meta-learning methods. Notably, we outperform Training&Finetuning which follows the traditional solutions to handle new events. It indicates some common characteristics of few-shot events have been captured to enhance newly-deployed events.

Convergence Curves. As shown in Fig. 3, our model not only outperforms other variants but also achieves fast and stable convergence. Besides, in comparison, our model has a better starting point with more expressive embeddings based on meta-training tasks. However, meta-learning models tend to overfit, especially when the support set is small, which makes early stopping important.

Ablation Study. We conduct experiments on three ablations: **(1) w/o network augmentation** that directly inputs the sparse EBSN to EA-GAT while auxiliary triples are only for optimizing. **(2) w/o auxiliary learning** that removes auxiliary learning in the task-level loss. **(3) MAML** removes both components. According to Fig. 4, all ablations show relatively poor results compared to the full model. Surprisingly, a significant drop is observed without auxiliary learning, which identifies that the reconstruction of source EBSNs boosts user preference learning. Besides, another decrease is found without network augmentation. MAML shows the worst results without them, showing their synergistic effects. Actually, both components are inspired by the interconnection nature of EBSNs. Finally, as support edges decrease, a larger gap is found between the full model and ablations, showing its superiority in few-shot scera.

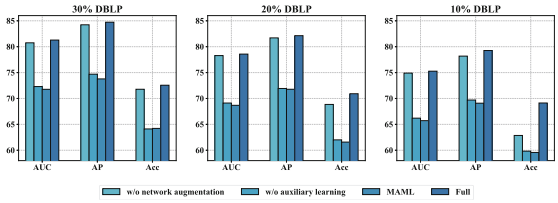

Fig. 4. The ablation results for DBLP dataset.

6 Conclusion

In this work, we studied FSL on new EBSNs. First, we defined a task distribution and considered a unified target event as the few-shot event. Then, for each task, an event-aware link prediction model was proposed with a joint objective. Overall, we followed MAML to achieve knowledge transfer. Finally, the experiments have illustrated the superiority of fast adaption and overall results.

Acknowledgements. This work was supported by the grants from National Natural Science Foundation of China (No. U20A20229, 62072423), the USTC Research Funds of the Double First-Class Initiative (No. YD2150002009), and the Alibaba Group through Alibaba Innovative Research (AIR) Program.

References

1. Bose, A.J., Jain, A., Molino, P., Hamilton, W.L.: Meta-graph: Few shot link prediction via meta learning. CoRR abs/1912.09867 (2019)
2. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML, vol. 70, pp. 1126–1135 (2017)
3. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: WWW, pp. 2704–2710 (2020)
4. Huang, K., Zitnik, M.: Graph meta learning via local subgraphs. In: NeurIPS (2020)
5. Lv, Q., et al.: Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In: KDD. pp. 1150–1160 (2021)
6. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., Navigli, R., Vidal, M.-E., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., Alam, M. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
7. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: NeurIPS, pp. 4077–4087 (2017)
8. Vashishth, S., Sanyal, S., Nitin, V., Talukdar, P.P.: Composition-based multi-relational graph convolutional networks. In: ICLR (2020)
9. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: ICLR (2015)
10. Zhou, F., Cao, C., Zhang, K., Trajcevski, G., Zhong, T., Geng, J.: Meta-gnn: on few-shot node classification in graph meta-learning. In: CIKM, pp. 2357–2360 (2019)