Unaligned Federated Knowledge Graph Embedding

Deyu Chen[1], Hong Zhu[1], Jinguang Gu[2], Rusi Chen[3], and Meiyi Xie[1](✉)

[1] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
{deyuchen,zhuhong,xiemeiyi}@hust.edu.cn

[2] Department of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, China
simon@wust.edu.cn

[3] State Grid Hubei Electric Power Research Institute, Wuhan, China

Abstract. Recently, knowledge graph embedding (KGE) methods under the federated learning paradigm have received much attention. Its privacy-preserving decentralized training method effectively utilizes the knowledge graphs held by different clients. Existing federated KGE frameworks collaboratively train the global model by aggregating aligned entity embeddings among clients. However, in real-world scenarios, the lack of aligned entities and the high heterogeneity among knowledge graphs constrain their potential. To address these issues, we propose a federated KGE framework that does not depend on any aligned set but uses structure information. The framework introduces a set of basis edges to model the general structure information. Then, we use two separate modules on clients to encode structure and feature representations, respectively. Finally, clients only upload structure parameters for aggregation on the server. The framework uses a new unaligned federated KGE paradigm to tackle the heterogeneity of multi-source knowledge graphs. Experimental results on benchmark datasets show that UniFE achieves superior results even compared to federated KGE frameworks using the aligned set.

Keywords: Knowledge graph embedding · Federated learning · Heterogeneous data

1 Introduction

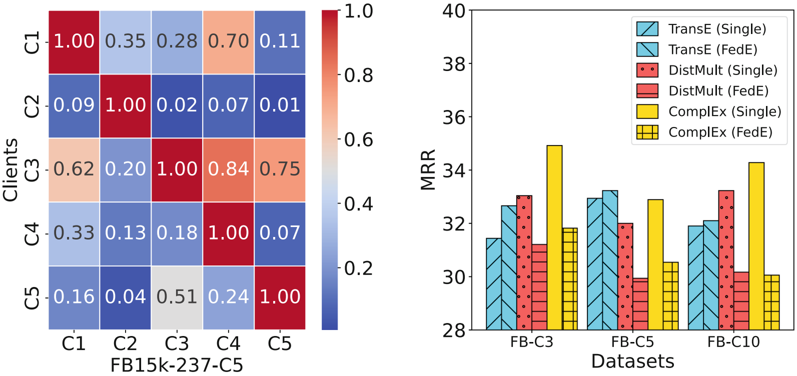
Knowledge graphs (KGs) are widely used data structures that model facts as (h, r, t) representing head entity h connected to tail entity t through relation r. Many large-scale knowledge graphs have been proposed and used on many downstream tasks [16,17,22]. Since knowledge graphs are usually incomplete, the knowledge graph completion (KGC) task is proposed, which predicts missing links based on the observed triplets to discover new facts. A popular solution [2,

G. Demartini et al. (Eds.): ISWC 2024, LNCS 15231, pp. 180–198, 2025.
https://doi.org/10.1007/978-3-031-77844-5_10

20] is to embed the observed triplets in a low-dimensional space to find potential connections of entities, known as knowledge graph embedding (KGE).

The application of KGs relies on the availability of large-scale data. However, organizations tend to maintain their private knowledge graphs, and laws also restrict data sharing across organizations, such as the GDPR[1]. So the KGs objectively become data silos due to commercial or privacy agreements [13].

Federated learning (FL) is a learning paradigm proposed recently, which allows clients to learn a global model jointly without data sharing directly [15]. In order to better utilize multi-source KGs with privacy-preserving, some works [4,5,30] proposed and explored the federated knowledge graph completion (FKGC) task which doing KGC task in the federated setting. The task is set to have overlapping entities among clients without overlapping relations. The server aggregates the parameter of aligned entities among clients to train a centralized global model. The paradigm improved the performance of the KGC task while maintaining privacy preservation. It also alleviated large graphs' computing resource consumption and data maintenance costs through server aggregation.



(a) (b)

Fig. 1. (a) An example of inter-client aligned entity proportion on FB15k-237-C5. The proportion is calculated by dividing the number of aligned entities between two clients and the entities' number of client owning more entities. (b) The MRR performance of *Single* setting and FedE using different KGE methods.

In real-world scenarios, the combination of KGE and FL has more challenges. First, entity alignment is a costly work and a significant research problem. Figure 1(a) shows the aligned entity investigation for the federated KGE benchmark dataset. The result represents the aligned entity set between two clients as a percentage of the client with the larger entity set.[2] The investigation

[1] https://gdpr-info.eu.

[2] The specific calculation for each cell: $\frac{|\mathcal{E}_x \cap \mathcal{E}_y|}{max(|\mathcal{E}_x|,|\mathcal{E}_y|)}$, where $\mathcal{E}_x$ and $\mathcal{E}_y$ are the entity sets owned by the corresponding clients.

demonstrates that aligned entities are often fewer or even unavailable in the heterogeneous federated setting. Second, simple aggregation of highly heterogeneous clients often leads to adverse results. The performance comparison between local training (*Single* setting) and average aggregation method FedE [4] in heterogeneous environments shown in Fig. 1(b), and the results show that an increase in heterogeneity often corresponds to a decrease in performance that is even worse than local training. Entities may also have different features in distinct contexts, so it is necessary to adopt better approaches than direct aggregation. Finally, the holders of KGs may be concerned about uploading their complete entity list to the server for reasons. The less information transmitted to the server brings a lower contribution cost and the more willing the clients are to participate [1].

These issues lead us to explore federated knowledge graph embedding learning without any aligned set. We introduce the structure perspective, which aims at the general structure representation for heterogeneous knowledge graphs. Structure information is a crucial feature in graph-related tasks. Existing works propose parameter aggregation among different graphs through decoupled learning [8,14,21] and generative approaches [29], which have been effective on tasks such as federated graph classification but not considered the multi-relational data. Inductive KGE models provide the feasibility of capturing the general patterns over knowledge graphs using meta-learning [3,6].

To address the above challenges, we proposed an **Un**al**i**gned **F**ederated Knowledge Graph **E**mbedding framework, UniFE for short, which does not depend on any aligned set for federated KGE. Our framework follows the common setting of FL with a server and a set of clients. Due to no aligned set, we intuitively use a set of desensitized basis edges based on adjacency structure of entities as the bridge among clients. Then, we construct the Basis Structure Graph (BSG) using the basis edges to represent the general patterns on KGs. Finally, we learn the structure representation of entities based on the BSG. To tackle the adverse effect on aggregation of highly heterogeneous KGs, we use separate modules on clients: the structure representation module and the feature representation module, inspired by the decoupled learning [8,14,21]. The server aggregates only the parameters related to structure learning. Moreover, in order to validate the effectiveness, we evaluate our proposed framework in unaligned setting for FKGC benchmark datasets. The results demonstrate that our framework outperforms existing FKGC baselines, even if they use the aligned set.

In summary, main contributions of our work are as follows:

- We consider a more realistic scenario for the FKGC task without any aligned set among clients. For the new task, we provide the structure perspective for learning KGE models in heterogeneous federated setting.
- We propose a heterogeneous federated KGE framework, UniFE. The framework encodes the general structure patterns among clients, uses decoupling learning the feature representations and structure representations of entities, and supports plugging in different existing KGE methods.

- We conduct extensive experiments on the FKGC benchmark datasets without aligned set. Experimental results show that our framework is effective in the unaligned setting.

2 Related Work

2.1 Knowledge Graph Embedding

In recent years, many KGE models have been proposed. They aim to map entities and relations into low-dimensional spaces to find unobserved facts. TransE [2] is a representative approach based on a translation mechanism that keeps the entities and relations in space with the constraint that $h + r \approx t$. DistMult [28] uses tensor decomposition, which only handles symmetric relations due to the use of diagonal matrices. ComplEx [24] introduces complex embeddings to solve the problem of modeling asymmetric relations. RotatE [20] models the relations as the rotation from the source entity to the target entity in the complex space, and it can infer the composition pattern. ConvE [7] introduces a multi-layer convolutional network model that effectively models common scenarios about the nodes with high indegree.

2.2 GNN for Multi-relational Data

Graph neural networks [12], which aggregate the neighbor embeddings and generate representations of nodes, do not consider the problem in multi-relational data scenarios. R-GCN [18] extends to relation-specific aggregation of node neighbors, enabling an encoder for relational data. SACN [19] combined the benefit of GCN and ConvE using learnable weights for more accurate embeddings of graph nodes. CompGCN [25] jointly embeds nodes and relations to improve effectiveness. It also solves the over-parameterization problem of previous methods and can be easily generalized to them.

2.3 Federated Learning over Knowledge Graphs

Federated learning uses data decentralized across different clients to train a global model under privacy-preserving. It alleviates the lack of data and computing resources on the clients, which is a promising approach to training large, effective models. FedAvg [15] first proposed a paradigm for FL, proving common initialization and average aggregation of parameters to be effective.

FedE [4] first defines the FKGC task and proposes a federated KGE framework based on FedAvg. Its main idea is to optimize the global model using average aggregation of aligned entity embeddings. FedEC [5] introduces embedding-contrastive learning to tackle the problem of data heterogeneity. It extends FedE to alleviate the adverse effects of heterogeneity by maximizing the agreement between the global embeddings and locally updated embeddings. FedLU [30] maintains local and global models on each client. It uses mutual distillation from two models and only aggregates global models to avoid drift between local optimization and global convergence due to high heterogeneity. However, this line of work relies on the paradigm proposed by FedE, aggregating only through aligned entities among clients.

DP-FLames [10] proposes three inference attacks under above paradigm to quantify the privacy risks of federated KGE. Then, it proposes a differential private federated KGE with private selection and an adaptive privacy budget allocation policy to mitigate the privacy threat. Since our setting does not have explicitly aligned entities, it avoids the three attacks.

Additionally, existing works [8,21] studied FL over graphs for tasks such as graph classification, but the features of multi-relational data have yet to be considered. Other works [3,6] focus on inductive KGE and knowledge extrapolation in the federated setting, which differs from our goal.

3 Preliminaries

A knowledge graph is defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ denotes a set of entities, $\mathcal{R}$ denotes a set of relations, and $\mathcal{T}$ denotes a set of triplets. Large-scale knowledge graphs are usually incomplete, and the KGC task is to predict missing links based on existing triplets. A popular solution is the KGE models, which embed entities and relations to low-dimensional vector spaces to predict the triplets formed as $(h, r, ?)$ or $(?, r, t)$ to be a true fact or not.

The FKGC task typically has two roles: a centralized server S and a set of clients with local knowledge graphs $\{\mathcal{G}_i\}_{i=1}^{C}$ of number C. The task aims to enhance the local clients' performance of the KGC task by aggregating the embeddings of overlapped entities among clients.

FedE [4] first proposed a federated knowledge graph embedding framework. The server performs a global initialization $\mathbf{E}_1^g$ for entities during initialization. In communication round r, it utilizes the global embedding $\mathbf{E}_r^g$ aggregated from the previous round $r-1$. The global embedding is conducted with a permutation $\mathbf{P}^c\mathbf{E}_r^g$ to each client $c \in C$. Once a local training of l epochs is completed, the local embedding $\mathbf{E}^c$ is obtained. After all clients have completed their training, the weights uploaded by the clients are aggregated as follows:

$$\mathbf{E}_{r+1}^g = (\mathbb{1} \oslash \sum_{c=1}^{C} \mathbf{v}^c) \otimes \sum_{c=1}^{C} \mathbf{P}^{c\top} \mathbf{E}_{r+1}^c, \tag{1}$$

where $\mathbb{1}$ represents all-one vector, $\mathbf{v}^c$ represents existence vector of client c, $\oslash$ and $\otimes$ represents element-wise division and multiply respectively. The framework repeats this process to minimize the average local training objective.

Notably, we assume that the clients' knowledge graphs do not have overlapping entities or relations, which presents a significant challenge compared to prior works [4,5], as there is no aligned set as a bridge among clients. We denote the task as unaligned FKGC, which is formalized as follows:

Given a centralized server S and a set of clients with local knowledge graphs $\{\mathcal{G}_i\}_{i=1}^{C}$ of number C, $\bigcap_{i=1}^{C} \mathcal{G}_i = \varnothing$. The unaligned FKGC aims to establish a set of general basis representations $\mathcal{B} = \{b_1, b_2, ..., b_n\}$ among different KGs and aggregate the locally learned parameters of basis representations on the server, which enhances the effectiveness of KGC for the clients.

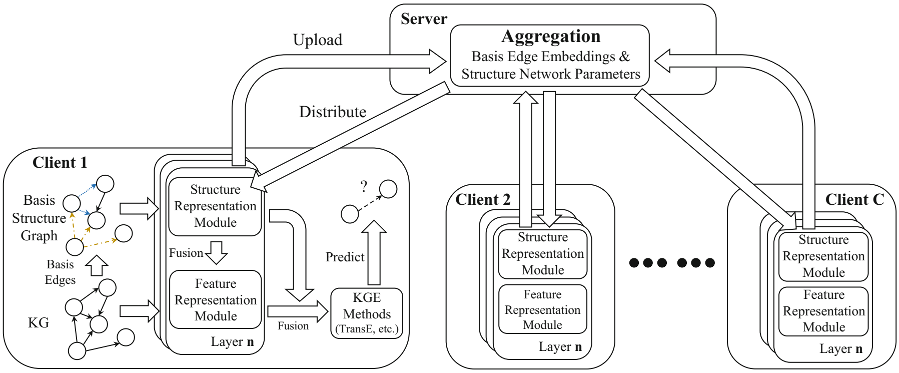


Fig. 2. The overview of UniFE. The framework uses the server-client FL architecture. Each client contains a structure representation module and a feature representation module. It protects client privacy better by only aggregating the structure parameters, which avoids the transfer of aligned entity embeddings.

4 Methodology

To solve the above issues, we proposed an **Un**al**i**gned **F**ederated Knowledge Graph **E**mbedding framework, UniFE. We focus on improving the KGC task in the heterogeneous federated environment without any aligned set. In order to address the absence of aligned entities among clients, an intuitive idea is to find a set of general structure representations over clients' KGs called basis edges. Our proposed framework constructs the Basis Structure Graphs to guide structure learning through the basis edges. For the data heterogeneity among clients, we use two separate modules to learn the feature and structure representations of entities on clients. The framework applies the fusion operation at each layer to enhance the encoding expression [21]. Finally, we only upload structure parameters to the server for aggregation, which avoids the adverse effect of global aggregation on local entity feature embeddings. The overview of our proposed framework is shown in Fig. 2.

In this section, we first introduce the modeling of general structure representations on KGs in Sect. 4.1. Next, we describe the local training procedure mainly using two separate modules in Sect. 4.2. Finally, we describe the global aggregation procedure in Sect. 4.3. The overall algorithm for the proposed framework is described in Algorithm 1.

```
Algorithm 1: UniFE Framework
Require: number of clients $C$, selected faction of clients $F$, number of local
         epoch $E$ and batch size $m$, learning rate $\eta$.
Server: Initialize basis embeddings $\mathbf{B}$ and structure encoder parameters $w^s$;
Clients: Each client initialize feature embeddings $\mathbf{H}^f$ and feature parameters
   $w^f_c$, then traverse to get relation-specific degree set $\mathcal{D}_c$ and generate BSGs;
for round $t = 0, 1, ...$ do
    $C_{sub} \leftarrow \lfloor F \times C \rfloor$;
    Server distributes $\mathbf{B}$ and $w^s$ to each client;
    for each client $c \in C_t$ in parallel do
        $\mathbf{B}^c_t, w^c_t \leftarrow LocalTraining(c, \mathbf{B}^c_t, w^{sc}_t)$;
    end
    $\mathbf{B}_{t+1} \leftarrow \{\mathbf{b}_{t+1} \leftarrow \sum_{c=1}^{C} \frac{N_c(b)}{N(b)} \mathbf{b}^c_t\}$;
    $w^s_{t+1} = \sum_{c=1}^{C} \frac{|\mathcal{D}_c|}{|\mathcal{D}|} w^s_{t_c}$;
end
// Ignore clients' locally held and updated parameters for
   clarity.
Function LocalTraining($c, \mathbf{B}, w^s$):
    for each local epoch $e$ from 1 to $E$ do
        $\mathcal{T}_m \leftarrow$ (split triplets $\mathcal{T}_c$ into batches of size $m$);
        for batch $n \in \mathcal{T}_N$ do
            Update two modules by Eq. 3 to 7 and compute loss $L$ by Eq. 8;
            $\mathbf{B} \leftarrow \mathbf{B} - \eta \nabla L$;
            $w^s \leftarrow w^s - \eta \nabla L$;
        end
    end
    return $\mathbf{B}, w^s$;
end
```

4.1 Basis Structure Modeling

Our framework considers KGE models in a highly heterogeneous federated setting without any aligned set. The existing FKGE paradigm requires maintaining a dictionary of entity mappings corresponding to each client on the server, which does not satisfy our task. Therefore, it is necessary to create a set of basis edges that can represent general structure patterns across different KGs.

Some works [3,9] explore modeling transferable information on KGs, which show that relative positions and interactions between relations are effective transferable features. Inspired by them, we design a way to model a node-level basis structure.

We first generate the BSGs from the structure of the KGs. The BSG is used to model the adjacent structure information of entities in corresponding KG, where entities are used as nodes and a specific adjacent structure pattern corresponds to an edge, called a basis edge. Figure 3(a) shows a typical example of transferable structure information. We can observe that nodes at corresponding

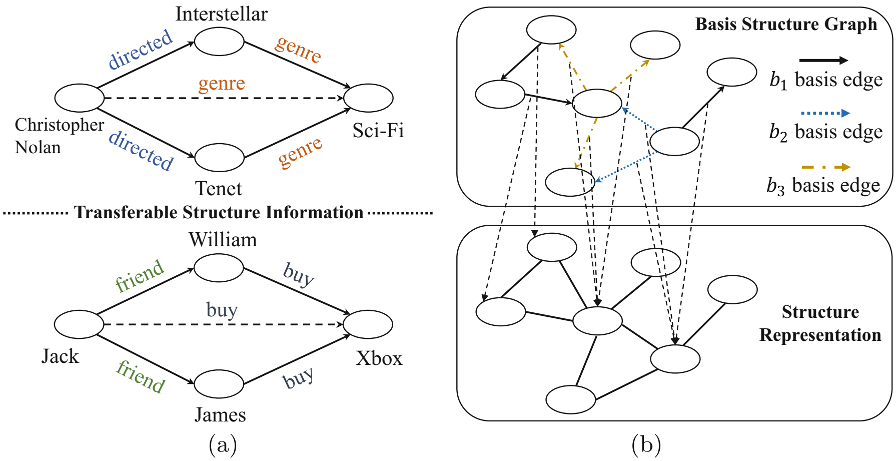


Fig. 3. (a) An example of transferable structure information about relative positions and interactions between different relations. (b) The process of generating structure representations over BSG. Relation-specific adjacent patterns correspond to different basis edges, and the aggregation of the basis edges obtains the structure representation.

positions in two KGs have the same degree composition. The degree is often applied as typical domain-invariant information in many graph-related tasks [21,26]. For multi-relational data, we consider entity connections under relations as slices. One slice represents structure within this relational domain. In UniFE, we intuitively consider the different relation-specific degrees as different patterns.

Specifically, we create a set of basis edges based on relation-specific degree sets. For a client's KG, $\mathcal{G}_c = (\mathcal{E}_c, \mathcal{R}_c, \mathcal{T}_c)$, where $\mathcal{E}_c$, $\mathcal{R}_c$, and $\mathcal{T}_c$ represent the set of entities, relations, and triplets on the client c, respectively. The set $\mathcal{D}_c$ of relation-specific degrees is generated by traversing the entities $e \in \mathcal{E}_c$, which compute the degree of each e with r as the outgoing edge, i.e., to compute the tail entities' number of distinct $(e_i, r_j, ?)$ in $\mathcal{T}_c$. The relation-specific degree set obtained:

$$\mathcal{D} = \{(e_1, r_1) : d_{11}), (e_1, r_2) : d_{12}, \ldots, (e_m, r_n) : d_{mn}\}, \tag{2}$$

where (e_i, r_j) represents the tuple composed of different entities and relations, and d_{ij} represents the number of tail entities under the tuple.

We use basis edges $\mathcal{B} = \{b_1, b_2, \ldots, b_k\}$ based on the relation-specific degree set to replace the edges in the original KG, i.e., the tuple $(e_i, r_i) = d$ in the relation-specific degree set, the corresponding edge is set to b_d basis edge. Traversing entities and corresponding relations generates BSG for the client's KGs, which denotes the structure patterns' composition of the entities, as shown in Fig. 3(b).

4.2 Local Training Procedure

Entity Structure Representation Module. The entity structure representation module learns structure representations of entities on the BSG. For the basis edges set $\mathcal{B}$, the module receives the embeddings of the basis edges $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k\}$ from the server. Then, it generates the input structure embedding $\mathbf{h}^s \in \mathbf{H}^s$ by the aggregation of the outgoing basis edges of entities as follows:

$$\mathbf{h}_e^s = \frac{1}{|O(e)|} \sum_{b_d \in O(e)} d^{-1} \mathbf{b}_d, \tag{3}$$

where $O(e)$ represents outgoing basis edges collection of entity e on the BSG, and $\mathbf{h}_e^s$ represents the structure embedding of entity e. The structure representation aggregates the adjacent relation-specific structure patterns of the entities. For each basis pattern to have the same importance, we multiply the basis edge embeddings by the reciprocal of its degree.

The structure representation module only focuses on the general structure information among clients. Since entities and relations have no alignment part with each other among clients, there is no need to model the attributes of edges. During the structure representations training process, specific relations are hidden. Based on the structure embeddings, a graph convolutional network (GCN) [12] is used. This network is initialized with parameters from the server distribution. The updating process of the structure representation at each layer is as follows:

$$\mathbf{H}^{s_{l+1}} = f(\hat{A}\mathbf{H}^{s_l}\mathbf{W}^l), \tag{4}$$

where $\hat{A} = \tilde{D}^{-\frac{1}{2}}(A+I)\tilde{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix with added self-loop, and $\tilde{D}$ defined as $\tilde{D}_{ii} = \sum_j (A+I)_{ij}$, $\mathbf{H}^{s_l}$ represents the entity hidden state of the l-th layer, which is the structure embeddings at the 0-th layer, i.e., $\mathbf{H}^{s_0} = \mathbf{H}^s$, and $\mathbf{W}^l$ is the transformation matrix of the l-th layer.

Entity Feature Representation Module. The entity feature representation module uses a multi-relational graph neural network encoder to learn features on the original KG. We use the fusion operation before feature encoding and output, which concatenates the feature and structure embeddings to enhance the encoding expression. Specifically, in order to benefit from the structure information, the entity feature representations and the structure representations are fused through the linear layer before each convolution layer and output layer, as in the following equation:

$$\mathbf{h}_e = \mathbf{W}_{fusion}[\mathbf{h}_e^f; \mathbf{h}_e^s] + b_{fusion}, \tag{5}$$

where $\mathbf{h}_e$ is the entity representation after fusing the feature representation $\mathbf{h}_e^f \in \mathbf{H}^f$ and structure representation $\mathbf{h}_e^s$ of entity e, $\mathbf{W}_{fusion}$ and b_{fusion} are the transformation matrix and bias, and $[\cdot;\cdot]$ is the vector concatenate operation.

Then, the fused representation is used as the entity representation input into the CompGCN [25] layer to get the feature representation of entities and relations, and the entity representation is updated as follows:

$$\mathbf{h}_e^{l+1} = f\left(\sum_{(e',r) \in N(e)} \mathbf{W}_{\lambda(r)}^l \phi(\mathbf{h}_{e'}^l, \mathbf{h}_r^l) \right), \tag{6}$$

where $N(e)$ is the set of outgoing neighbors of entity e, $f(\cdot)$ is the activation function, $\mathbf{W}_{\lambda(r)}^l$ is the relation-type specific parameter in layer l, i.e., using different transformation matrix for original relations, inverse relations, and the self-loop, respectively, $\phi(\cdot, \cdot)$ is composition operator, such as vector subtraction, multiplication, circular-correlation.

The relation representations are updated by the transformation matrix at each layer as follows:

$$\mathbf{h}_r^{l+1} = \mathbf{W}_{rel}^l \mathbf{h}_r^l, \tag{7}$$

where $\mathbf{W}_{rel}^l$ is the relation transformation matrix of layer l.

Training Objective To train the proposed model, we optimize binary cross-entropy loss with label smoothing [7] for the observed triplets on each client as follows:

$$L = \frac{1}{n} \sum_{(h,r,t) \in \mathcal{T}} (y \cdot log\sigma(S(h,r,t)) + (1-y) \cdot log\sigma(1 - S(h,r,t))) \tag{8}$$

where y is the label vector, σ is the activation function, $S(\cdot)$ is the score function.

4.3 Global Update Procedure

In the global aggregation stage, each client uploads only the parameters of the structure network and the basis embeddings to the server at communication round t, as follows:

$$\mathbf{b}_{t+1} = \sum_{c=1}^{C} \frac{N_c(b)}{N(b)} \mathbf{b}_t^c, \tag{9}$$

$$w_{t+1}^s = \sum_{c=1}^{C} \frac{|\mathcal{D}_c|}{|\mathcal{D}|} w_{t_c}^s, \tag{10}$$

where $\mathbf{b}$ and w^s represent basis embeddings and structure network parameters after aggregation, respectively, and $N_c(b)$ and $N(b)$ represent the number of the basis embeddings participating in the aggregation on the current client and all selected clients, $|\mathcal{D}|$ represents the entity-relation composition number of all selected clients.

This aggregation is privacy-preserving because structure networks and basis embeddings are insensitive. Moreover, the basis embeddings indirectly improve the whole framework. They generate entity structure representation, which enhances the feature representation module through the fusion operation. The basis embeddings also improve relation embeddings through entity-relation composition in layers of the feature representation module.

5 Experiments

In this section, we evaluate the following questions through experiments: 1) Does the performance of the proposed framework have enough potential for applications? 2) Is each module of the proposed framework effective? 3) Does the proposed framework have an advantage in convergence speed when it does not need to consider different contexts of entities? 4) Are there any stricter private settings?

5.1 Experimental Settings

Datasets. In order to evaluate the effectiveness of our proposed framework, we used FB15k-237-C3, -C5, and -C10 for heterogeneous FKGC task [30] and NELL-995-Fed3 [4]. The first three datasets are generated from FB15k-237 [23] by a clustering-based partition algorithm, which enables it to conform more to the realistic data distribution. The NELL-995-Fed3 is to randomly assign the relations of NELL-995 [27] to three clients and distribute triplets to clients based on the relations. This dataset is not divided for evaluating heterogeneity but can be used to evaluate the general applicability of our framework. It must be emphasized that in our proposed framework, the entity embeddings of each client are initialized independently, i.e., entities with identical names of different clients can be regarded as distinct instances, and they are invisible to each other, conforming with our setting. The data statistics are shown in Table 1.

Table 1. The statistics of datasets. It shows the average number of entities, relations, and triplets for different numbers of clients.

Dataset	Clients	Entities	Relations	Triplets
FB15k-237-C3	3	6262.7	79.0	103372.0
FB15k-237-C5	5	5119.0	47.4	62023.2
FB15k-237-C10	10	2176.9	23.7	31011.6
NELL-995-Fed3	3	33453.7	66.7	51404.3

Baselines. We compare our proposed framework in the unaligned setting with three existing federated knowledge graph embedding frameworks using the aligned entity set. FedE [4] first proposed the federate KGE framework, FedEC [5] first introduces contrastive learning to solve the heterogeneity of KGs, and FedLU [30] is the state-of-art federated KGE framework currently, which addresses data heterogeneity by maintaining different local and global models and transferring knowledge through mutual knowledge distillation. We use predictions of local models as results of FedLU.

The scoring of triplets uses several typical score functions TransE [2], DistMult [28], and ComplEx [24] for the baselines. Note that our framework used the composition of entities and relations, which is unsuitable to be applied to score functions with the unequal dimensions of the entity and relation embeddings such as RotatE [20]. Additionally, we implemented a convolution-based method ConvE [7] for UniFE. Since negative sampling is not used, we only set the comparison with the *Single* setting.

Evaluation Metrics. Following common evaluation metrics, we use the clients' average Hits at N (Hits@N) and Mean Reciprocal Rank (MRR) to evaluate the link prediction performance. The results use the rank in the *filtered* setting [2], which filters out the triplets that appear in the training, validation, or test set during the evaluation.

Implementation Details. We use Adam optimizer [11] with a learning rate 0.001. For the baselines, we refer to the implementation provided by Zhu et al. [30] and follow the main parameters provided. We use the higher results of the previously reported experiments and our reproduction for some existing experiments. The embedding dimension of entity and relation is 256. For fair comparisons, UniFE set the dimension of entity and relation embeddings to 128 because we have two modules. The number of basis edges is set to 50. The batch size is set to 1024. The experiments for *Single* setting evaluate the average MRR on the validation set every 10 epochs and use the early stop with patience of 3. For the federated setting, we use 3 local training epochs, evaluate the average MRR on the validation set every 5 communication rounds, and use the early stop with patience of 3.

5.2 Main Results

The main results of link prediction on FB15k-237-C3, -C5, -C10, and NELL-995-Fed3 are shown in Table 2 about MRR and Hits@1, which are the metrics that have significant influences on the downstream applications.

Based on these results, our proposed framework performs better than the *Single* setting. Specifically, on FB15k-237-C3, for TransE, DistMult, ComplEx, and ConvE, our approach achieves a relative improvement of 8.02%, 12.95%, 9.22%, and 4.57% in MRR, respectively. On the FB15k-237-C5, -C10, and NELL-995-Fed3, the relative improvement in MRR ranges from 1.91% to 14.50%. This improvement proves that our framework can benefit from KGs for different clients while preserving privacy.

Compared to FedE, our approach achieves an average relative improvement in MRR of 14.47%, 16.11%, 16.35%, and 18.27% on FB15k-237-C3, -C5, -C10, and NELL-995-Fed3, respectively. FedE uses average aggregation for aligned entity embeddings between clients. We can observe that FedE performs worse than *Single* setting in some cases, e.g., the ComplEx on FB15k-237-C3. This indicated that the same entity has different representations in different contexts. The average aggregation of highly heterogeneous KGs has an adverse effect in this situation. Our results relative to FedE and *Single* setting show that our approach can handle data heterogeneity better.

Table 2. Main results of link prediction on FB15k-237-C3, -C5, -C10, and NELL-995-Fed3. **Bold** numbers denote the best results with the same KGE methods in different settings.

KGE	Setting	FB15k-237-C3		FB15k-237-C5		FB15k-237-C10		NELL-995-Fed3	
		MRR	Hits@1	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1
TransE	Single	31.44	19.48	32.94	20.47	31.99	19.58	23.53	10.67
	FedE	32.66	20.25	33.23	20.66	32.10	19.86	19.92	7.98
	FedEC	33.15	20.78	33.58	21.13	32.75	20.40	20.35	8.28
	FedLU	33.83	21.53	34.36	21.94	33.47	21.13	23.64	11.32
	UniFE	**33.96**	**23.14**	**35.02**	**23.81**	**33.67**	**22.56**	**24.82**	**14.56**
DistMult	Single	33.04	22.17	32.00	21.85	33.23	22.90	26.37	18.72
	FedE	31.21	21.04	29.94	19.95	30.17	20.43	23.79	17.14
	FedEC	32.85	20.92	29.25	18.13	31.28	20.37	25.85	18.46
	FedLU	34.93	23.18	33.65	22.46	34.83	24.40	26.67	19.23
	UniFE	**37.32**	**26.77**	**36.64**	**26.09**	**36.67**	**26.24**	**28.32**	**19.50**
ComplEx	Single	34.92	24.05	32.89	23.13	34.28	23.58	28.73	21.07
	FedE	31.82	21.41	30.54	19,88	30.06	19.80	26.34	18.96
	FedEC	34.58	22.44	31.30	19.63	32.53	20.59	28.89	20.68
	FedLU	37.10	25.38	35.66	24.05	36.33	25.14	29.04	**21.16**
	UniFE	**38.14**	**27.86**	**36.82**	**26.63**	**36.86**	**26.53**	**29.28**	20.93
ConvE	Single	36.94	26.55	35.39	25.18	35.03	24.91	29.62	19.98
	UniFE	**38.63**	**27.72**	**38.31**	**27.51**	**37.58**	**26.67**	**32.43**	**23.29**

Compared to FedEC and FedLU, our approach improves the average MRR across all settings by 11.65% and 3.62%, respectively. FedEC uses embedding-contrastive learning to tackle data heterogeneity, while FedLU maintains two models, local and global, and uses mutual distillation to tackle data heterogeneity. The restrictions of the above two approaches lie in the coverage of aligned entities among clients. The above results demonstrate that our approach outperforms results in the highly heterogeneous federated environment by mining and aggregating general structure information.

Furthermore, the absolute accuracy of our proposed framework is only for showing its potential, as the required conditions are different from existing methods. For question 1), we propose a learning paradigm for the FKGC task that has better privacy preservation and more general applicability without weakening performance in highly heterogeneous environments.

5.3 Ablation Study

In order to demonstrate the necessity and effectiveness of our proposed modules, we designed experiments by initializing them differently or removing modules. Specifically, we propose some variants:

- **w/o struct. 128/256:** The variant does not use the structure representation module. It initializes entity embeddings independently. For fair comparison and to observe the effect of embedding dimensions on the entity feature module, we used 128 and 256 dimensions. The former is used to evaluate the actual effect of single modules, and the latter doubles the parameters to compare the UniFE having two modules.
- **w/o struct. agg.:** The variant uses the structure representation module without any aggregation. It co-initializes basis embeddings and structure network parameters but initializes others independently.
- **w/ ent. init.:** The variant uses the structure representation module with basis embeddings and structure network parameters aggregation. It co-initializes basis embeddings and structure network parameters. Additionally, it co-initializes the entity feature embeddings.
- **w/ ent. agg.:** The variant uses the structure representation module with basis embeddings and structure network parameters aggregation. It co-initializes basis embeddings and structure network parameters. Additionally, it co-initializes the entity feature embeddings and uses entity feature aggregation with reference to FedE.

Table 3. Ablation study with different initialization and modules for UniFE. **Bold** numbers denote the best results and underline numbers denote the second best results in different variants.

Setting	FB15k-237-C3		FB15k-237-C5		FB15k-237-C10		NELL-995-Fed3	
	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1	MRR	Hits@1
w/o struct. 128	34.78	24.15	32.79	22.08	32.23	21.90	29.70	20.95
w/o struct. 256	38.35	**27.96**	36.26	25.79	34.52	24.18	30.95	21.57
w/o struct. agg.	37.78	26.88	37.89	26.96	36.17	25.54	30.69	21.11
w/ ent. init.	38.48	27.62	**38.42**	**27.72**	37.19	26.18	32.36	22.86
w/ ent. agg.	37.81	26.90	37.79	27.11	36.91	25.98	**32.76**	22.99
UniFE	**38.63**	27.72	38.31	27.51	**37.58**	**26.67**	32.43	**23.29**

Table 3 shows the ablation study for the different variant models. From the results, we obtain the following observations:

1. Compared to *w/o struct. 128*, UniFE has an average relative 13.42% improvement of MRR. When compared to *w/o struct. 256*, UniFE has an average relative 5.01% improvement of MRR. Note that from the absolute amount of parameters, *w/o struct. 256* doubles the parameters of the feature representation module, which is more than the structure representation module with basis edges' number of 50. Moreover, this variant cannot aggregate parameters without aligned entity embeddings. The results demonstrate the parameter efficiency and potential of UniFE.

2. Compared to *w/o struct. agg.*, UniFE has an average relative 3.23% improvement of MRR. The results demonstrate the effectiveness of structure parameter aggregation.
3. The variant of *w/ ent. init.* obtained comparable results to UniFE. It concluded that different initializations of entity embeddings do not significantly influence. Our framework does not use co-initialization of entity embeddings, which is privacy-preserving without affecting performance.
4. Compared to *w/ ent. agg.* on FB15k-237-C3, -C5, -C10, *w/ ent. init.* and UniFE have average relative 1.40% and 1.79% improvements of MRR, respectively. The results show that the average aggregation of entity embeddings in highly heterogeneous environments has an adverse impact while bringing the risk to clients' privacy.
5. On NELL-995-Fed3, *w/ ent. agg.* have average relative 1.24% and 1.02% improvements of MRR compared with *w/ ent. init.* and UniFE. The results demonstrate that the average aggregation of entities has certain effects in the general setting that is not highly heterogeneous. Our framework's structure aggregation and entity aggregation do not conflict if realistic conditions are available.

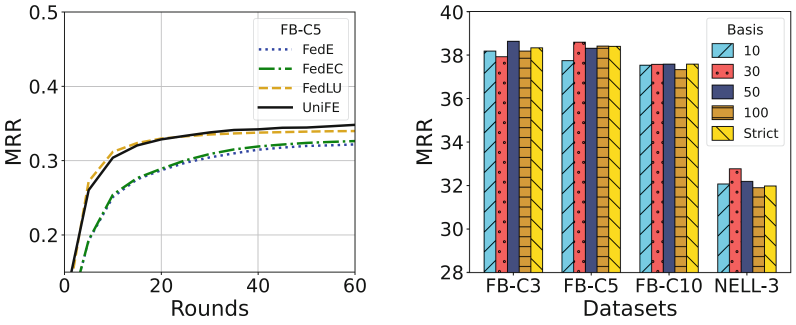
For question 2), the results of the ablation study show that each module of our proposed framework is effective.

5.4 Further Analysis

Convergence Speed. Figure 4(a) illustrates the MRR results on the validation set of FB15k-237-C5 when TransE is used for UniFE and baselines. From the results, our proposed framework has a comparable convergence speed with FedLU, and our convergence speed exceeds FedE and FedEC. The trend of convergence on other datasets is roughly the same. For question 3), the results demonstrate the favorable learning efficiency of our framework with less use of client information.

Stricter Privacy Setting. The client's willingness to engage in FL is related to the information transferred to the server. Specifically, less information transmitted to the server brings a lower contribution cost, and clients are more willing to participate [1]. So, we conducted experiments to analyze the impact of the number of basis edges on our framework. Figure 4(b) shows the model's MRR results when using different numbers of basis edges. The results show that the MRR tends to increase and then decrease in the range of the basis edges' number from 10 to 100. The number of basis edges needs a suitable value to model the general structure information among different KGs.

Based on these results, we consider whether minimizing the number of parameters exchanged further is possible. We design a variant with a stricter privacy setting for exploring question 4). The variant is based on a plain intuition that lower relation-specific degrees represent more specific structure patterns, while higher relation-specific degrees represent more broad structure patterns. So we



(a)

(b)

Fig. 4. (a) The MRR results with communication rounds on validation set of FB15k-237-C5. (b) The average MRR results of link prediction with different number of basis edges.

use the original number of relation-specific degrees when it is less than 20, whereas greater than 20, every ten items use the same basis edge, i.e., 20 to 29, 30 to 39, 40 to 49, and 50+ use different four basis edges.

From the results, compared to the default basis edges' number of 50, *Strict* setting further reduces the number of parameters and brings almost no performance loss. It is also a promising work on how to design basis edges for scenarios with higher privacy requirements.

6 Conclusion and Future Work

In this paper, we consider a more realistic scenario of the FKGC task, the heterogeneous federated environment without any aligned set. For the new setting, we propose UniFE, an unaligned federated knowledge graph embedding framework. The framework uses the pre-defined basis edges modeling the general structure information and two separate modules to learn the structure representations and feature representations in clients' local training. The clients only upload the structure parameters to the server for aggregation. This new learning paradigm does not depend on any aligned set and tackles the heterogeneity of multi-source KGs. We conducted extensive experiments on benchmark datasets, and the results demonstrate the effectiveness of our framework. In the future, we plan to explore more modeling approaches for general structure information over KGs and try to apply them to more tasks in the federated setting, such as the inductive KGC task.

Supplemental Material Statement: Source code for baselines and datasets FB15k-237-C3, -C5, -C10 are available at https://github.com/nju-websoft/FedLU. The dataset NELL-995-Fed3 is available at https://github.com/zjukg/FedE. Source code for UniFE is available at https://github.com/deyu-chen/UniFE.

Acknowledgements. The work was supported by the State Grid Corporation Science and Technology Project: Application research of a new generation of artificial intelligence in the security analysis and decision-making of future power grid (5100-202099522A-0-0-00).

References

1. Bi, X., Gupta, A., Yang, M.: Understanding partnership formation and repeated contributions in federated learning: an analytical investigation. Manag. Sci. (2023)
2. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, pp. 2787–2795 (2013)
3. Chen, M., et al.: Meta-learning based knowledge extrapolation for knowledge graphs in the federated setting. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence. IJCAI 2022, pp. 1966–1972 (2022)
4. Chen, M., Zhang, W., Yuan, Z., Jia, Y., Chen, H.: Fede: embedding knowledge graphs in federated setting. In: IJCKG'21: The 10th International Joint Conference on Knowledge Graphs, pp. 80–88 (2021)
5. Chen, M., Zhang, W., Yuan, Z., Jia, Y., Chen, H.: Federated knowledge graph completion via embedding-contrastive learning. Knowl. Based Syst. **252**, 109459 (2022)
6. Chen, M., et al.: Meta-knowledge transfer for inductive knowledge graph embedding. In: SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 927–937 (2022)
7. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), pp. 1811–1818 (2018)
8. Dwivedi, V.P., Luu, A.T., Laurent, T., Bengio, Y., Bresson, X.: Graph neural networks with learnable structural and positional representations. In: The Tenth International Conference on Learning Representations. ICLR 2022 (2022)
9. Galkin, M., Yuan, X., Mostafa, H., Tang, J., Zhu, Z.: Towards foundation models for knowledge graph reasoning. In: The Twelfth International Conference on Learning Representations. ICLR 2024 (2024)
10. Hu, Y., et al.: Quantifying and defending against privacy threats on federated knowledge graph embedding. In: Proceedings of the ACM Web Conference 2023. WWW 2023, Austin, TX, USA, 30 April 2023–4 May 2023, pp. 2306–2317 (2023)
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations. ICLR 2015 (2015)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations (2017)
13. Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-IID data silos: an experimental study. In: 38th IEEE International Conference on Data Engineering. ICDE 2022. pp. 965–978 (2022)

14. Liu, Y., Zheng, Y., Zhang, D., Lee, V.C.S., Pan, S.: Beyond smoothing: unsupervised graph representation learning with edge heterophily discriminating. In: Thirty-Seventh AAAI Conference on Artificial Intelligence. AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence. IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence. EAAI 2023, pp. 4516–4524 (2023)
15. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. AISTATS 2017, vol. 54, pp. 1273–1282 (2017)
16. Ren, H., et al.: LEGO: latent execution-guided reasoning for multi-hop question answering on knowledge graphs. In: Proceedings of the 38th International Conference on Machine Learning. ICML 2021, vol. 139, pp. 8959–8970 (2021)
17. Ruiz, C., Ren, H., Huang, K., Leskovec, J.: High dimensional, tabular deep learning with an auxiliary knowledge graph. In: Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023. NeurIPS 2023 (2023)
18. 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., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
19. Shang, C., Tang, Y., Huang, J., Bi, J., He, X., Zhou, B.: End-to-end structure-aware convolutional networks for knowledge base completion. In: The Thirty-Third AAAI Conference on Artificial Intelligence. AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference. IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence. EAAI 2019, pp. 3060–3067 (2019)
20. Sun, Z., Deng, Z., Nie, J., Tang, J.: Rotate: knowledge graph embedding by relational rotation in complex space. In: 7th International Conference on Learning Representations. ICLR 2019 (2019)
21. Tan, Y., Liu, Y., Long, G., Jiang, J., Lu, Q., Zhang, C.: Federated learning on non-IID graphs via structural knowledge sharing. In: Thirty-Seventh AAAI Conference on Artificial Intelligence. AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence. IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence. EAAI 2023, pp. 9953–9961 (2023)
22. Tang, Z., Pei, S., Peng, X., Zhuang, F., Zhang, X., Hoehndorf, R.: Neural multi-hop logical query answering with concept-level answers. In: 22nd International Semantic Web Conference, ISWC 2023, vol. 14265, pp. 522–540 (2023)
23. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality. CVSC 2015, pp. 57–66 (2015)
24. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of the 33nd International Conference on Machine Learning. ICML 2016, vol. 48, pp. 2071–2080 (2016)
25. Vashishth, S., Sanyal, S., Nitin, V., Talukdar, P.P.: Composition-based multi-relational graph convolutional networks. In: 8th International Conference on Learning Representations. ICLR 2020 (2020)
26. Wu, J., He, J., Xu, J.: Demo-net: degree-specific graph neural networks for node and graph classification. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 406–415 (2019)

27. Xiong, W., Hoang, T., Wang, W.Y.: Deeppath: a reinforcement learning method for knowledge graph reasoning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. EMNLP 2017, pp. 564–573 (2017)
28. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: 3rd International Conference on Learning Representations. ICLR 2015 (2015)
29. Zhang, K., Yang, C., Li, X., Sun, L., Yiu, S.: Subgraph federated learning with missing neighbor generation. In: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021. NeurIPS 2021, pp. 6671–6682 (2021)
30. Zhu, X., Li, G., Hu, W.: Heterogeneous federated knowledge graph embedding learning and unlearning. In: Proceedings of the ACM Web Conference 2023. WWW 2023, pp. 2444–2454 (2023)