IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 July 2023, accepted 4 August 2023, date of publication 14 August 2023, date of current version 18 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3304915

RESEARCH ARTICLE

Double Graph Attention Network Reasoning Method Based on Filtering and Program-Like Evidence for Table-Based Fact Verification

HONGFANG GONG, CAN WANG, AND XIAOFEI HUANG
School of Mathematics and Statistics, Changsha University of Science and Technology, Changsha 410114, China

Corresponding author: Can Wang (cwang0324@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61972055, and in part by the Natural Science Foundation of Hunan Province under Grant 2021JJ30734.

ABSTRACT Table-based fact verification requests parsing table and statement structure and performing numerical and logical reasoning. Previous methods may select erroneous programs and ignore the interpretability of table-based fact verification. Thus, we propose a double graph attention network reasoning method based on filtering and program-like evidence (DGMFP). In detail, we initially obtain the filtering evidence based on tables and the program-like evidence based on logical forms to incorporate the semantic and symbolic information of evidence. Then, we construct an evidence graph with statement-evidence pairs as nodes and use the kernel in graph neural network to conduct more fine-grained joint reasoning and improve the interpretability of table-based fact verification. We also construct a connected graph with all entities and functions in the program-like evidence as nodes and use the graph attention network (GAT) to capture more fine-grained relationships within the program-like evidence. Finally, we connect the outputs of two GAT models and BERT model to predict labels. Experimental results on TABFACT show that DGMFP outperforms all baselines with 76.1% accuracy. Ablation studies further indicate that constructed two graphs, filtering evidence, and program-like evidence play an important role in better understanding the semi-structured table.

INDEX TERMS Natural language inference, table-based fact verification, GAT, filtering evidence, program-like evidence.

I. INTRODUCTION

Fact verification is important to examining fake news [1], [2], [3], rumors [4], [5], as well as scientific fact-checking [6]. Most existing research has focused on unstructured text information [7], [8], [9], [10], [11], which are meaningful facts. Structured or semi-structured information, such as tables or databases, are also ubiquitous. Recently, table-based fact verification with authenticity measured by two labels, namely, ENTAILED or REFUTED, which demonstrates statements are correct/incorrect through the given semi-structured table, has received considerable attention. Figure 1 shows an example.

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia.

Table-based fact verification involves linguistic inference and symbolic operations (e.g., *counting*, *addition* or *sorting*), which brings challenges to the verification. Pre-trained models (e.g., BERT [12]) show excellent performance to verify simple statements, but tend to fail when statements with complex logical reasoning characteristics, such as *greater than* and *total*, are encountered. For example, the table in Figure 1 shows that given a statement, 4.0 *is the lane total when rank is* 3, we can infer that the label is refuted because the actual lane is 3.0. Therefore, learning complex logical reasoning features in statements is crucial in this task.

To address these challenges, we summarize the existing approaches into two categories: (1) program enhanced approaches [13], [14], [15], [16], which mainly utilize programs (i.e., logical forms) generated by the semantic parser to

Table with title swimming at the 2008 summer olympics - women 's 100 metre backstroke

rank	lane	name	nationality	time
1	5	natalie coughlin	united states	59.43
2	4	reiko nakamura	japan	59.64
3	3	gemma spofforth	great britain	59.79
4	6	hanae ito	japan	01:00.13
5	7	elizabeth simmonds	great britain	01:00.39
6	2	julia wilkinson	canada	01:00.60
7	1	sophie edington	australia	01:01.05
8	8	kseniya moskvina	russia	01:01.06

Statement : reiko nakamura was in lane 4 and ranked 2. **Label :** ENTAILED

Statement : 4.0 is the lane total when rank is 3. **Label :** REFUTED

Program-like Evidence :

- eq { 2 ; hop { filter_eq { filter_eq { all_rows ; name ; reiko nakamura } ; lane ; 4 } ; rank } } = True
- eq { 2 ; hop { filter_eq { all_rows ; name ; reiko nakamura } ; rank } } } = True
- eq { 2 ; hop { filter_eq { all_rows ; name ; reiko nakamura } ; rank } } } = True
- ...

FIGURE 1. Example of table-based fact verification.

represent the statement as prior information and employ the graph neural network (GNN) to acquire inexplicit relationships. And (2) table-based pre-trained approaches [17], [18], which mainly utilize elaborate model structure TAPAS [19] and pre-training tasks [20], [21], [22] to improve the reasoning ability of semi-structured information. Unlike program enhanced approaches, table-based pre-trained approaches based on BERT's encoder to encode tables without generating programs. Despite the significant progress of previous works, several challenges still remain in table-based fact verification.

The weakly supervised programs are generated from the semantic parser, which inevitably contain noise. Due to the weak supervised signals in the semantic parser, program enhanced models may select erroneous programs that return the true label. Ideally, a natural approach to use programs is to regard logical forms with mathematical operations as supplementary evidence for tables. Previous approaches also ignore the interpretability of table-based fact verification, focusing on improving the accuracy of validation, which consequently resulted in untrustworthy outcomes.Thus, the table-based fact verification becomes an extremely demanding task because of the need for fine-grained reasoning ability to judge that the statement is correct/incorrect.

Based on these considerations, the aim of our work is building a table-based fact verification model, performing more fine-grained reasoning and provide interpretability during the reasoning process. Thus, we newly define table-based fact verification task as a multi-stage task and propose a double graph attention network (GAT) reasoning method based on filtering and program-like evidence, namely, DGMFP. In detail, given a statement and the corresponding table, to incorporate the semantic and symbolic information of evidence, we retrieve the filtering evidence by using the table itself as part of the evidence and obtain the program-like evidence from logical forms according to a rule-based method. Then, we concatenate the statement, table caption, filtering and program-like evidence as the statement–evidence pair for the first time. We use BERT to generate the initial representations of statement–evidence pairs, as well as the initial representations of entities (such as *reiko nakamura*) and functions (such as *eq*[1]) in the program-like evidence. Subsequently, to fully explore the fine-grained relations between each piece of evidence and increase the interpretability of table-based fact verification, we construct an evidence graph with statement–evidence pairs as nodes and use the kernel [23] to carry out feature propagation between nodes. We also construct a connected graph by using the structure of program-like evidence, and use GAT [24] to catch implicit relations among different nodes. Finally, we connect the outputs of two GAT models and pre-trained BERT model to predict labels. We find in experiments that this output combination method can improve the model's performance as much as possible.

We conduct widespread experiments on a large-scale dataset TABFACT [13] to show that the proposed model. Generally, experimental results indicate that DGMFP can outperform all baseline systems in terms of label accuracy. Ablation studies and quality analysis also verify the effectiveness of DGMFP and the ability of DGMFP to select pivotal evidence.

This work's contributions are summarized in three folds:

(1) We newly define table-based fact verification task as a multi-stage task and design a double GAT reasoning method based on filtering and program-like evidence, helping to incorporate the semantic and symbolic information of the evidence and capture fine-grained relationships between and within the evidence.

(2) We propose to use the neural matching kernel for evidence representation learning of table-based fact verification for the first time, which helps to improve the interpretability of table-based fact verification by propagating clues among the pieces of evidence through multi-layer graph attention.

(3) We assess the proposed method through widespread experiments on TABFACT dataset and verify the effectiveness of DGMFP relative to baselines. DGMFP outperforms all baseline systems.

II. RELATED WORK

A. NATURAL LANGUAGE INFERENCE

The aim of natural language inference (NLI) is to reason a natural language hypothesis as either entailment, contradiction or neutral based on a natural language premise. Many fact verification systems utilize NLI techniques [25], [26], [27], [28] to verify the claim. Chen et al. [25] demonstrated that LSTM-based inference methods outperform all existing methods. Peters et al. [26] improved six challenging natural language processing problems by using contextualized word representations that are easy to incorporate into models. Tay et al. [27] designed a new NLI model that uses factorization layers, enhancing the representations of words. Ghaeini et al. [28] coded the relationship between hypotheses and premises, significantly improving final predictions. Recently, more and more text frameworks have included

[1]The function *eq* indicates that the cell value is equal to the given number, as detailed in the appendix of Reference [13]. *not_eq*, *filter_eq*, *and* and *hop* in Section III are also described in detail in Reference [13].

structured or semi-structured information, for instance, knowledge graphs [29], tables [30], [31] or images [32], [33]. Table-based fact verification is also relevant to NLI task, where the premises are presented by the semi-structured tables composed of text.

B. TABLE-BASED FACT VERIFICATION

Table-based fact verification serves as a meaningful task because it provides reliable information and prevents the spread of structured and semi-structured disinformation. Chen et al. [13] proposed a TABFACT dataset and designed two different models: Latent Program Algorithm (LPA) and Table-BERT. Based on Table-BERT, which encodes tables and statements, Zhang et al. [34] considered only the corresponding row and column in the representation of each cell through masking, proposing the structure-aware transformer (SAT). However, they fall short in the symbolic reasoning aspects. A series of works use logical forms generated by the LPA for validation given that logical forms can bring substantial prior information to understand the statement. Ou and Liu [35] defined a operation-oriented tree based on LPA, mining structure features. Zhong et al. [14] used GNN to encode heterogeneous graphs containing logical forms and relevant table cells. Shi et al. [15] and Yang et al. [16] used program selection module to select the best logical forms. Program nodes, table nodes, and statement nodes were introduced into a heterogeneous graph to predict the label in [15], and different sources of evidence from the statement, table and program tree are integrated into GAT in [16]. Shi et al. [36] designed a graph-based verification network using processed logical forms as evidence. However, weakly supervised logical forms are generated from the semantic parser, which inevitably contains noise.

In addition, Yang et al. [37] designed a framework with strict robustness to row and column order perturbations, namely, TABLEFORMER. TAPAS-based models, such as [17], [18], [20], [21], and [22], encoded the features of rows and columns of a table and utilized data augmentation as intermediate evidence to enhance table-based fact verification models. However, while better models such as TAPAS have emerged for table-based fact verification task, TAPAS-based models require high requirements because pre-training requires significant computing resources, and they ignore the interpretability of table-based fact verification.

C. GRAPH NEURAL NETWORKS

Knowledge of the field of GNNs [38], [39], [40], [41] is required to provide connections between different evidence nodes and statement nodes. Key idea about GNN [42], [43] is learning node embedding from the aggregation of neighborhood features information. Velickovic et al. [24] first proposed GAT, which implicitly assigns different weights to neighbor nodes. Kipf and Welling [44] designed graph convolutional network to semi-supervise the classification of graph structure information. However, these methods fail to learn more fine-grained relationships among the pieces of evidence and the corresponding statement. Wang et al. [45] designed a heterogeneous GNN, which utilizes hierarchical attention to generate node representations by aggregating features from meta-path-based neighbors. Hu et al. [46] introduced a novel information aggregation method, named heterogeneous graph transformer, based on meta-relational learning and heterogeneous attention. Fu et al. [47] aggregated intra-metapath and inter-metapath features, generating node representations. However, the above approaches do not apply when no clear relationship or meta-path exists between nodes.

Liu et al. [48] proposed the kernel graph attention network (KGAT), which enables more fine-grained reasoning through kernel-based attentions. The neural matching kernel can learn the interaction of words or phrases in the embedding layer, so leveraging the neural matching kernel is an effective method to model text matches [23], [49]. Reference [50] had also shown that the correlation between query and documents can be better modeled through the integration of the kernel with contextualized representations (i.e., BERT [12]). In view of the advantages of these methods, we innovatively introduce the idea of the neural matching kernel into table-based fact verification, using KGAT and GAT to capture more fine-grained relationships between and within the evidence.

D. PRE-TRAINED LANGUAGE MODELS

Models for pre-trained language representations (e.g., ELMo [26] or OpenAI GPT [51]) have been proven to be very efficient in NLI tasks. BERT [12] is a new method for pre-trained language representation and pre-trains deep bidirectional representations through jointly tuning the right and left context in embedding layer. We use BERT to encode text in this work.

III. METHODOLOGY

A. PROBLEM DEFINITION

Given an unidentified sentence called a statement s and the corresponding table T, we newly define table-based fact verification as a multi-stage task. It initially collects a set of programs P related to the statement by LPA [13], then generates evidence set $E = \{e_1, e_2, \ldots, e_n\}$ from the table and programs according to a rule-based method, and finally predicts the statement label $y \in$ {ENTAILED, REFUTED} based on the evidence, as in Equation (1).

$$\begin{cases} \mathcal{F}_{LPA}(s, T) \rightarrow P \\ \mathcal{F}_{retriveal}(s, T, P) \rightarrow E \\ \mathcal{F}_{predict}(s, E) \rightarrow y \end{cases} \tag{1}$$

Notably, a successful table-based fact verification should satisfy two criteria: 1) The predicted result of label y is true; 2) At least one sentence is included in set E.

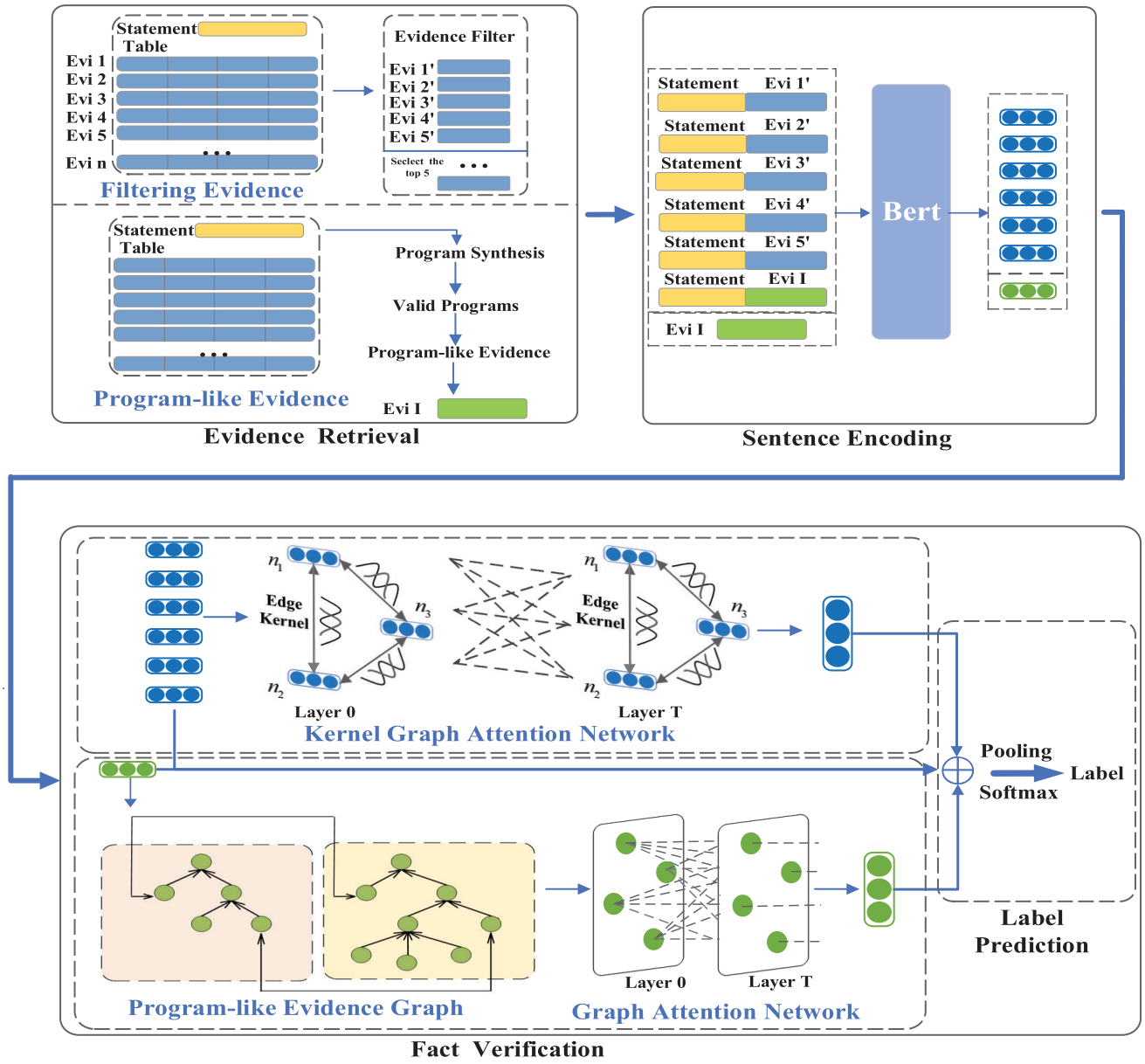

FIGURE 2. Overall architecture of DGMFP.

B. PIPELINE

Figure 2 shows the overall framework of DGMFP, consisting of three components, i.e., the evidence retrieval, sentence encoding, and fact verification component. In the upper section, we first use the evidence retrieval module to obtain evidence for table-based fact verification. Then we use BERT [12] to generate the initial representations of statement-evidence pairs, as well as the initial representations of entities and functions in the program-like evidence. In the lower section, to predict the final label, we construct two graphs based on the statement-evidence pairs and the program-like evidence for propagating and aggregating the representations of the statement and evidence.

C. EVIDENCE RETRIEVAL

1) FILTERING EVIDENCE

In order for the evidence to contain both semantic and symbolic information, we include the table itself as part of the evidence. Based on the number of words that each row of the table shares with the corresponding statement, we only retain the top five rows of the table related to the statement as the filtering evidence to reduce the memory space and shorten training time. Considering that the counting operation account for a large proportion, we also improve our model's performance through converting counting operation into the semantic matching question. In detail, the frequency of repeated cell contents in each column is calculated as a summary cell, resulting in a summary row that is filled to

the end of the table. Take the fourth column in Figure 1 as an example, and its summary cell is *count*: *japan*, 2, *great britain*, 2.

2) PROGRAM-LIKE EVIDENCE

Programs have rich logical operations. We consider that logical forms can provide valuable information beyond tables for table-based fact verification. In this work, to obtain the program-like evidence, we follow LPA [13] to synthesize valid programs with pre-defined functions, denoted as $P = \{(P_i, A_i)\}_{i=1}^{N}$, where P_i represents the *i-th* program and A_i represents the label returned by the program executed on the table, i.e., *True* or *False*. Then we follow Shi et al. [36] to select logical forms with a returned label of *True* and decompose logical forms containing function *and* into two separate pieces, while removing logical forms that contain functions with negative meanings (such as *not_eq*). Finally, we integrate the program-like evidence into the dataset TABFACT [13] to enhance our model's ability to understand semi-structured tables.

D. SENTENCE ENCODING

For sentence encoding, we make use of BERT [12] to generate the token representations of statement and evidence. Specifically, for table-based fact verification task, we concatenate the statement s, table caption t, filtering and program-like evidence e as the statement-evidence pair (s, evi) (where evi is t # e) for the first time to form the input sequence x:

$$x = [[CLS]; s; [SEP]; t \# e; [SEP]]. \tag{2}$$

where $[CLS]$ and $[SEP]$ are the identifiers for BERT. Then, we feed sequence x into BERT to generate the token representations of x, represented as $C \in R^{L_1 \times d_1}$:

$$C = BERT(x). \tag{3}$$

where d_1 is the size of BERT hidden states, as well as L_1 represents the length of x. The initial representation of the statement-evidence pair can be represented by the representation of the first token ($[CLS]$) as $C_0 \in R^{L_1 \times d_1}$, and the remaining sequences $C_{1:m+n} \in R^{L_1 \times d_1}$ indicate the statement and evidence representations. The statement representations are $C_{1:m} \in R^{L_1 \times d_1}$, and the evidence tokens are $C_{m+1:m+n} \in R^{L_1 \times d_1}$.

At the same time, similar to Equations (2) and (3), we separately feed the program-like evidence into BERT to generate the token representation of the program-like evidence. The initial representation of the program-like evidence can be represented by the representation of the first token ($[CLS]$) as $E_0 \in R^{L_2 \times d_1}$, where L_2 represents the length of the program-like evidence.

E. FACT VERIFICATION

This section describes our double GAT and its application in table-based fact verification. Figure 2 shows that the double GAT model includes two components, i.e., KGAT and GAT.

1) KGAT

To fully explore the fine-grained relationships between each piece of evidence, we adopt the neural matching kernel in GNN to carry out feature propagation between pieces of evidence. Based on previous research [48], we initially construct an evidence graph with statement-evidence pairs as nodes, as well as connect all statement-evidence pairs with edges to obtain a fully-connected evidence graph $N = \{n_1, n_2, \ldots, n_r\}$ with r nodes.

The evidence feature propagation in KGAT [48] is performed through edge kernel, integrating information from neighbors through a hierarchical attention mechanism. It uses token level attentions to generate the representation of nodes, then uses sentence level attentions to integrate information from neighbors. In particular, we take the former as an example, at the t-th step, the representations of $t-1$ layer nodes are known, i.e., $N^{(t-1)} = \{n_1^{(t-1)}, n_2^{(t-1)}, \ldots, n_r^{(t-1)}\}$.

a: TOKEN LEVEL ATTENTION

This work uses token level attention to obtain a more fine-grained token representations $\hat{n}_b$ of neighbor node $n_b^{(t-1)}$.

To obtain the i-th token attention weight $\alpha_i^{b \to a}$ in $n_b^{(t-1)}$, we initially construct the translation matrix M, and each of its elements is the cosine similarity of the token representations between a-th node and b-th node, denoted as $M(i, j)$:

$$M(i, j) = cos(C_j^b, C_i^a). \tag{4}$$

where $C_j^b \in R^{L_1 \times d_1}$ is the j-th token representations of node b, as well as $C_i^a \in R^{L_1 \times d_1}$ is the i-th token representations of node a. Subsequently, for Equation (4), we extract the matching feature $\vec{K}(M(i, \cdot))$ with a K-dimensional vector by K kernels [49], [50], [52].

$$\vec{K}(M(i, \cdot)) = \{K_1(M(i, \cdot)), K_2(M(i, \cdot)), \ldots, K_{K-1}(M(i, \cdot)), K_K(M(i, \cdot))\}. \tag{5}$$

The effect of each kernel $K_k(M(i, \cdot))$ in Equation (5) is decided by kernel used. Our proposal uses Gaussian kernel to extract features, as in Equation (6):

$$K_k(M(i, \cdot)) = \log \sum_j exp(-\frac{(M(i, j) - \mu_k)^2}{2\delta_k^2}). \tag{6}$$

where δ_k is the width of k-th kernel, as well as μ_k is the mean of k-th kernel [23].

Then, this work utilizes a linear layer to calculate the i-th token attention weight $\alpha_i^{b \to a}$ in $n_b^{(t-1)}$:

$$\alpha_i^{b \to a} = softmax_i(Linear(\vec{K}_k(M(i, \cdot)))). \tag{7}$$

The more fine-grained token representations $\hat{n}_b$ of the neighbor node $n_b^{(t-1)}$ can be obtained from the combination of the attention weights in Equation (7), as follows:

$$\hat{n}_b = \sum_{i=1}^{m+n} \alpha_i^{b \to a} \cdot C_i^a. \tag{8}$$

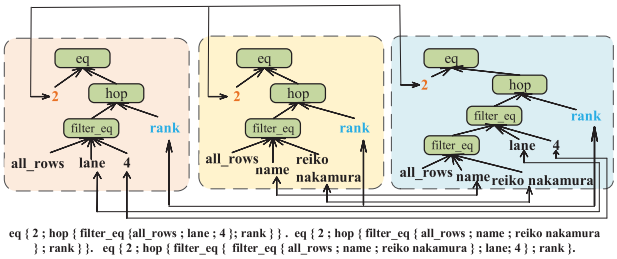

FIGURE 3. Example of the program-like evidence graph.

b: SENTENCE LEVEL ATTENTION

This work uses sentence level attention to integrate information from neighbors to node $n_a^{(t)}$. Integration is conducted through the attention mechanism, the same as in the previous work [6]. According to the *a-th* node $n_a^{(t-1)}$, we initially compute attention weight $\beta^{b\to a}$ of node $n_b^{(t-1)}$, as in Equation (9):

$$\beta^{b\to a} = softmax_i(MLP(n_a^{(t-1)} \| \hat{n}_b)). \tag{9}$$

where $\|$ denotes the concatenate operator.

Subsequently, this work updates the *a-th* node representations through combining neighbor's node representations $\hat{n}_b$, denoted as $n_a^{(t)}$:

$$n_a^{(t)} = (\sum_{b=1}^{r} \beta^{b\to a} \cdot \hat{n}_b) \| n_a^{(t-1)}. \tag{10}$$

2) GAT

Given the retrieved program-like evidence, according to the previous study of Shi et al. [36], we construct a connected graph with all entities and functions by using the structure of program-like evidence. An example is shown in Figure 3. In detail, to learn more fine-grained relationships within the program-like evidence, we consider each function and entity as a graph node, and add edges from the entity pointing to the function between entity nodes and the corresponding function node. To convert the graph to a connected graph, we add edges between each entity node that has the same content.

In sentence encoding, while outputting the representation of the program-like evidence node from BERT [12], we output the start index and length of each logical form, as well as the information required to encode the program-like evidence graph (i.e., start and end index of edge, node type, the entity node corresponding to the start index of edge, the entity node corresponding to the end index of edge). For the node with multiple word pieces in the program-like evidence graph, we conduct average pooling for the corresponding position.

After the graph construction and node initialization, our inference network is designed based on GAT [23] to catch implicit relations among different nodes. The nodes in the graph are either pre-defined functions in LPA [13], or entities linked to the table or statement, Thus, we model different types of nodes during messaging. In order to get different types of node representations, we combine the representations of each different type of node as follows:

$$h_i^u = W_u(o_i) + b_u. \tag{11}$$

$$h_i = [h_i^q \| h^s \| h_i^u]. \tag{12}$$

where $o_i \in R^U$ is a one-hot vector that presents node types (i.e., function node or entity node), as well as $W_u \in R^{U\times d_1}$ and $b_u \in R^{d_1}$ are trainable parameters. U is set to 2, indicating the kinds of node types. h_i^q and h^s in Equation (12) are obtained through the node initialization process of BERT. h_i^q represents node representation, and h^s represents statement representation.

We note that many nodes in the program-like evidence graph are semantically independent of statements, for instance, *filter_eq*, *hop* and *all_rows*. Thus, this work uses a node pruning method, automatically pruning and filtering these nodes. An example is shown in Figure 4. We initially obtain correlation scores between nodes and the corresponding statement as in Equation (13):

$$s_i = \sigma(W_\varphi[h_i^q \| h^s] + b_\varphi). \tag{13}$$

where $W_\varphi \in R^{2d_1\times d_1}$ and $b_\varphi \in R^{d_1\times d_1}$ are trainable parameters. Subsequently, based on the scores of nodes, we remove nodes with a score less than probability θ. At last, considering a removed node, we add edges between its parent node and child nodes, forming a new graph.

In particular, we take the former as an example, at the *t-th* step, the representations of $t-1$ layer nodes are known, i.e., $H^{(t-1)} = \{h_1^{(t-1)}, h_2^{(t-1)}, \ldots, h_v^{(t-1)}\}$. We update the node representations as follows:

$$p_{ij}^{(t)} = LeakyReLU(W_\Phi^T[h_i^{(t-1)} \| h_j^{(t-1)}]). \tag{14}$$

$$\eta_{ij}^{(t)} = softmax(p_{ij}^{(t)}) = \frac{exp(p_{ij}^{(t)})}{\sum_{k\in\mathcal{N}_i} exp(p_{ik}^{(t)})}. \tag{15}$$

$$h_i^{(t)} = \mathop{\Vert}_{z=1}^{Z} \sigma(\sum_{j\in\mathcal{N}_i} \eta_{ij}^{(t)z} W_\zeta^z h_j^{(t-1)}) + h_i^{(t-1)}. \tag{16}$$

where $W_\Phi^T \in R^{d_1\times d_1}$ and $W_\zeta^z \in R^{d_1\times d_1}$ are trainable parameters, as well as $\mathcal{N}_i$ indicates the neighbors of node i. Z indicates the number of attention heads. $\eta_{ij}^{(t)z}$ in Equation (16) indicates the normalized attention coefficient calculated by *z-th* attention head. $h_i^{(t-1)}$ and $h_j^{(t-1)}$ present the representations of node i and node j at $t-1$ layer. $h_i^{(t)}$ presents the representation of node i at t layer.

3) LABEL PREDICTION

We first formulate the final representation of all nodes in KGAT and GAT as $N = \{n_1, n_2, \ldots, n_r\}$ and $H = \{h_1, h_2, \ldots, h_v\}$, and concatenate the outputs of KGAT and GAT and two $[CLS]$ tokens from BERT model's output to improve DGMFP's performance as much as possible. Then we utilize an attention pool layer to obtain final representation g as follows:

$$g = \mathcal{F}_{Pooling}(\mathcal{F}_{concat}(C_0, N, H, E_0)). \tag{17}$$

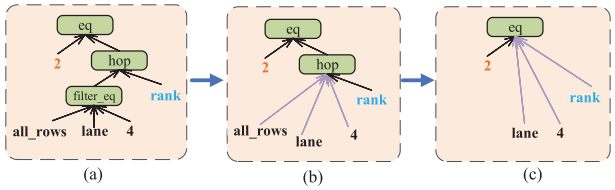

FIGURE 4. Example of the node pruning method. Figure (a) to Figure (b) indicates removing *filter_eq* and adding edges between *all_rows*, *lane*, 4 and *hop*. Subsequently, Figure (b) to Figure (c) indicates removing *all_rows* and *hop* and adding edges between *lane*, 4, *rank* and *eq*.

At last, we predict the accuracy of labels by feeding the vector g in Equation (17) into a classifier.

IV. EXPERIMENTS

In this section, we first describe the datasets, experimental settings, and baseline systems. Then, we compare DGMFP with all baseline systems. Next, in order to obtain the effects of different modules in DGMFP, we perform ablation studies. Finally, by obtaining the attention weight entropy and distribution from KGAT, we explore how the neural matching kernel captures fine-grained relationships between each piece of evidence and increases the interpretability of table-based fact verification.

A. DATASETS

Consistent with existing researches on table-based fact verification task, we evaluate the DGMFP model on TABFACT [13] dataset that has already been divided. The dataset contains 118K statements and 16K tables, and each sample is marked ENTAILED or REFUTED, indicating statements are correct/incorrect through the given semi-structured table. The TABFACT dataset is roughly divided into train, validation (val), and test sets at a ratio of 8:1:1 by stratified sampling to ensure that the samples in the divided train, val, and test sets have similar distributions. Besides the standard test, train and val sets, the dataset provides multiple subsets. To distinguish the difficulty of the evaluation, the test set is split into the complex and simple test sets. In addition, a small test set is used to compare human evaluation and machine evaluation. Table 1 shows the statistics of TABFACT and lists the number of tables, statements, and labels for different sets. In the train set, the number of positive samples is slightly more than the negative samples. The val and test sets both have rather balanced distributions on positive and negative samples. Therefore, this work only uses the official accuracy metric based on the previous work [13].

TABLE 1. Statistics of the TABFACT dataset.

Datasets	Splits					
TABFACT	Train	Val	Test	Test (simple)	Test (complex)	Small
# Table	13182	1696	1695	833	862	298
# Statement	92283	12792	12779	4171	8608	1998
# Label: 1	50820	6478	6425	2120	4305	989
# Label: 0	41463	6314	6354	2051	4303	1009

B. EXPERIMENTAL SETTINGS

According to Chen et al. [13], we use BERT [12] as the backbone to build our model. In our experiments, we use Adam optimizer with a weight decay 2e−4 and a warmup rate of 0.1. We run 20 epochs with a maximum sequence length of 512, a batch size of 8, as well as an initial learning rate of 1e−5. The size of all hidden layers is 768, the same as BERT-base model. We set the probability θ to 0.3. According to the existing work [52], we set the kernel size to 21. DGMPF is optimized using cross entropy loss. Our experiments are run on a workstation equipped with 80GB of INTEL and 2 A40 GPUs.

C. BASELINE SYSTEMS

We describe all advanced baselines for comparison with our model.

- BERT-only [13]: The easiest way to infer using only statements to train a BERT classifier;
- Table-BERT [13]: A BERT-based model which considers the table-based fact verification as a NLI task, using BERT to encode the statement and table to predict the label;
- LPA [13]: A weakly supervised method which generates the suitable programs for statements and sorts the candidate programs based on the transformer [51];
- LogicalFactChecker [14]: A graph network model which uses different semantic parsers to generate programs and construct a heterogeneous graph to represent the programs;
- HeterTFV [15]: A heterogeneous graph-based reasoning approach which jointly encodes the statement, table and logical forms into a heterogeneous graph to fuse different information;
- SAT [34]: A structure-aware transformer [53] model which masks partial tokens in the self-attention layers;
- ProgVGAT [16]: A graph network model which uses a marginal loss-based program selection module to generate optimal logical forms and employs GAT [23] for reasoning to predict the label;
- SASP [35]: A structure-aware semantic parsing model which defines an operation-oriented tree mining structure features and integrates structure features into program generation;
- LERGV [36]: A graph-based reasoning network model which views programs as additional evidence and employs GAT [23] to perform reasoning.

D. EXPERIMENTAL RESULTS

Table 2 reports the number of parameters for each model and lists our experimental results, where numbers in bold represent optimal performance. Compared to all baseline models, the number of parameters of DGMFP is moderate. DGMFP surpasses all baseline models with significant improvements, achieving 76.1% accuracy on the test set.[2]

[2]We do not directly compare with TAPAS-based methods, because DGMFP is evidence-centric, not table-centric. TAPAS-based methods cannot take evidence-statement pairs as input.

TABLE 2. Experimental results on TABFACT. Horizontal and Vertical represent the scanning strategies for table linearization. T and F represent the table and statement, respectively. Template stands for the templates used to concatenate the table cells.

Model	Number of parameters (Million)	Val (%)	Test (%)	Test (simple) (%)	Test (complex) (%)	Small (%)
BERT classifier w/o Table	110	50.9	50.5	51.0	50.1	50.4
Table-BERT-Vertical-T+F-Template	177.85	56.7	57.0	60.6	54.3	55.5
Table-BERT-Horitonzal-T+F-Template	177.85	66.1	65.1	79.1	58.2	68.1
LPA	39.48	65.2	65.0	78.4	58.5	68.6
LogicalFactChecker	-	71.8	71.7	85.4	65.1	74.3
HeterTFV	-	72.5	72.3	85.9	65.7	74.2
SAT	177.85	73.3	73.2	85.5	67.2	-
ProgVGAT	473.47	74.9	74.4	88.3	67.6	76.2
SASP	178.26	75.0	74.9	87.6	68.8	-
LERGV	180.97	75.6	75.5	87.9	69.5	77.8
DGMFP (ours)	182.14	**75.9**	**76.1**	**88.5**	**70.2**	**79.1**
DGMFP (10-fold CV)	182.14	-	**76.31 ± 0.17**	-	-	-
Human Performance	-	-	-	-	-	92.1

Table 2 shows that DGMFP is superior to LPA, Table-BERT, and SAT by large margins, illustrating the benefits of evidence that fully contains semantic and symbolic information. We can also see that compared with the approaches based on the semantic parser, namely, LogicalFactChecker, HeterTFV, ProgVGAT and SASP, DGMFP's performance improves by 0.6%-4.4%. This indicates the effectiveness of DGMFP, demonstrating its ability to capture fine-grained relationships between and within the evidence to better understand the semi-structured tables. In addition, our model outperforms LERGV by nearly 1 point on complex set, demonstrating DGMFP's ability to handle complex statements. DGMFP also reaches competitive performance on small set, reducing the gap between human and machine evaluation to 13%. Therefore, the above conclusions prove the usefulness of DGMFP for table-based fact verification.

Subsequently, from Table 2, we zoom in all baseline models' performance. For models that do not use logical forms, Table-BERT (65.1%) is the first baseline to linearize the table and statement through BERT. Through masking, SAT (73.2%) only considers the corresponding row and column in the representation of each cell, suggesting that understanding the table structure is critical for the BERT-based approaches. For models that use logical forms, LPA (65.0%) is the first baseline to generate lots of logical forms through the semantic parser, and the performance is almost equivalent to Table-BERT (65.1%). Other models perform joint reasoning on logical forms, as well as the corresponding statement and table, such as LogicalFactCachecker (71.7%), HeterTFV (72.3%), ProgVGAT (74.4%), SASP (74.9%) and LERGV (75.5%), which outperform LPA by large margins. LERGV is the optimal approach in all baseline models, suggesting that the generated logical forms as additional evidence are important to improve the evaluation capabilities for table-based fact verification models. We also observe that DGMFP outperforms these models, demonstrating its ability to capture more fine-grained relationships between and within the evidence.

TABLE 3. Ablation study on TABFACT (%).

Model	Val	Test	Test (simple)	Test (complex)	Small
DGMFP (ours)	**75.9**	**76.1**	**88.5**	**70.2**	**79.1**
w/o GAT	75.4	75.2	87.7	69.4	78.2
w/o KGAT	74.9	74.5	87.1	68.6	77.3
w/o PEvi	75.2	74.9	87.7	69.0	78.6
w/o FEvi	70.8	71.2	82.7	65.8	74.4

We use K-fold cross-validation for model evaluation and report the model's performance along with standard deviation. We set $K = 10$ and use stratified sampling to randomly and uniformly divide the dataset into 10 parts. Since it will be costly to redivide the Test (Simple), Test (Complex) and Small sets for each test set required for 10-fold cross-validation, we only report the performance on the complete test set. Table 2 shows the mean (76.31%) and standard deviation (0.17%) of the 10-fold cross-validation. The results indicate that the performance of the model using 10-fold cross-validation is slightly better.

E. ABLATION STUDY

We remove additional corpus of GAT (expressed as w/o GAT), KGAT (w/o KGAT), filtering evidence (w/o FEvi) and program-like evidence (w/o PEvi) to analyse the effect of different modules. Table 3 lists the detailed performance of removing different subsets on TABFACT. In general, performance variations on different sets show the differences of these data. DGMFP achieves the best accuracy in all sets, especially on complex set containing the most challenging samples in the TABFACT dataset.

1) VALIDATION OF EVIDENCE

w/o FEvi and w/o PEvi reduce the accuracy of the val set by 0.7%-5.1%, indicating that the evidence that adequately contains semantic and symbolic information has a great effect on the proposed method. The filtering evidence and program-like evidence plays an important role

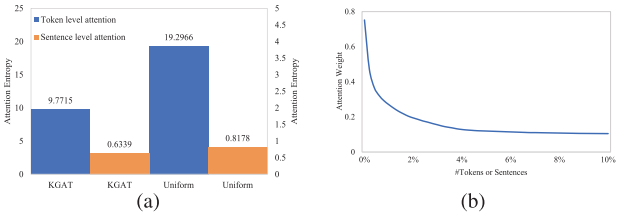

FIGURE 5. Effectiveness of kernel in KGAT. (a). Attention weight entropy from KGAT. For comparison, the entropy of uniform attention weights is also shown. The smaller entropy indicates more focused attention. (b). Attention weight distribution from KGAT, which includes both token level attention weight and sentence level attention weight. Only the top 10% tokens are displayed, and the remaining 90% of tokens follow standard long tail distributions.

in better understanding the semi-structured table. w/o FEvi only retains the program-like evidence, and w/o PEvi only retains the filtering evidence. Compared with w/o PEvi, after deleting the filtering evidence, our model only retains logical forms to be executed, resulting in a more significant performance degradation, that is, a 5.1% decrease in the val set. This result shows that although providing effective additional information to tables is beneficial to improve performance, information about the tables themselves are more important.

2) VALIDATION OF TWO GRAPHS

w/o KGAT or w/o GAT indicates using the concatenation of the model input and graph node representation to replace KGAT or GAT. This operation results in a 0.9%-1.6% drop on the val set, demonstrating the effectiveness of KGAT and GAT in capturing the fine-grained relationships between the pieces of evidence and within the program-like evidence. w/o KGAT only relies on GAT to present the performance of DGMFP. Compared with w/o GAT, deleting KGAT module results in a more significant performance degradation, that is, a 1.6% decrease in the val set. This result further demonstrates the advantage of the neural matching kernel for evidence representation learning in table-based fact verification.

TABLE 4. Comparison of different heads of GAT with respect to their effectiveness.

Number of GAT heads	Val
0	75.40
1	75.62
2	75.65
3	75.79
4	**75.92**

To further verify the effectiveness of GAT, this work compares different heads of GAT and sets them to 0-4 heads. The case with 0 heads in GAT is equivalent to removing the GAT module from DGMFP. Table 4 lists the experimental results for each case. It can be observed that DGMFP with GAT modules consistently performed better than without GAT modules, with the best performance being the four heads improving the results by 0.52%.

F. EFFECTIVENESS OF KERNEL

This set of experiments further illustrates the effectiveness of kernel in KGAT [48]. Kernel attentions are used to integrate evidence cues in evidence graph. We study kernel attentions through the entropy, which indicates the attention weight is focused or dispersed. Considering the size of token level attention and sentence level attention, we replace the token level attention to a uniform distribution that obeys [0, 0.005] and sentence level attention to a uniform distribution that obeys [1,0]. Figure 5(a) shows token level attention' entropy and sentence level attention' entropy in KGAT. It also shows the entropy of replacing token level attention or sentence level attention with uniform attention. Compared to the uniform distribution, the token and sentence level attentions focus on fewer tokens and have a smaller token attention entropy, illustrating that KGAT can assign more weights to some important tokens based on the kernel.

We also study attention weight distribution of the kernel, including token level attention weights and sentence level

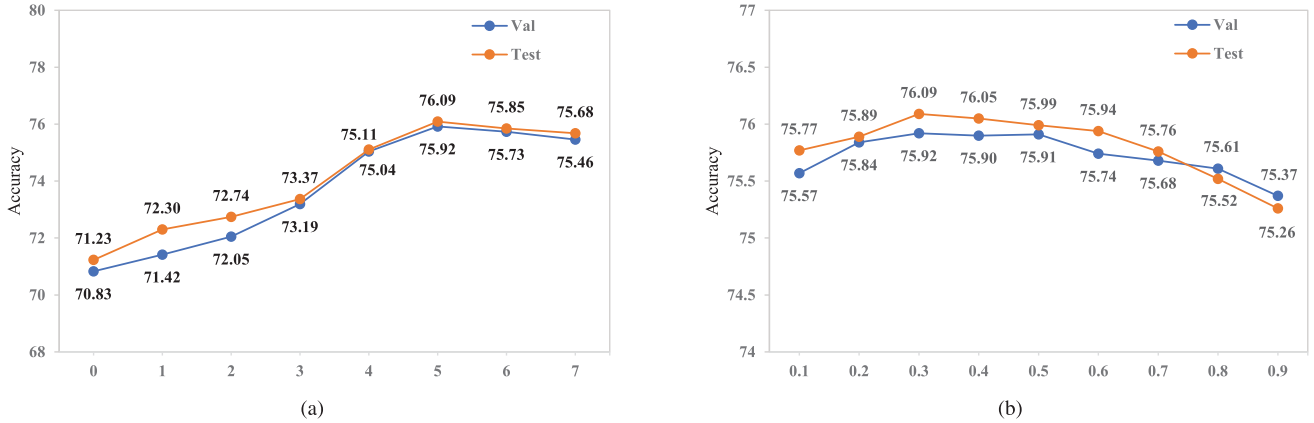

FIGURE 6. Graphs of sensitivity analysis. (a). The accuracy of val and test sets under different amounts of filtering evidence. (b). The accuracy of val and test sets under different probability θ values.

TABLE 5. An example statement whose verification requires multiple pieces of evidence. The table caption is "Swimming at the 2008 summer olympics-women's 100 metre backstroke".

Statement: reiko nakamura was in lane 4 and ranked 2.	Sentence level attention weight
(1)[Table caption] rank, lane, name, nationality, time.	$\beta^{(1\rightarrow1)}(0.2318)$
(2)[Table caption] 2, 4, reiko nakamura, japan, 59.64.	$\beta^{(2\rightarrow1)}(0.2441)$
(3)[Table caption] 4, 6, hanae ito, japan, 1:00.13.	$\beta^{(3\rightarrow1)}(0.0945)$
(4)[Table caption] 6, 2, julia wilkinson, canada, 1:00.60.	$\beta^{(4\rightarrow1)}(0.0915)$
(5)[Table caption] count: japan, 2, greatbritain, 2.	$\beta^{(5\rightarrow1)}(0.1046)$
(6)[Table caption] eq { 2 ; hop { filter_eq { filter_eq { all_rows ; name ; reiko nakamura } ; lane ; 4 } ; rank } }, eq { 2 ; hop { filter_eq { all_rows ; name ; reiko nakamura } ; rank } } }, eq { 2 ; hop { filter _eq { all_rows ; lane ; 4 } ; rank } } } , ...	$\beta^{(6\rightarrow1)}(0.2435)$
Label: ENTAILED.	

attention weights. As shown in Figure 5(b), the kernel attentions are focused on fewer words, rather than being distributed almost evenly across all words. When combining evidence cues from multiple pieces, the kernel provides the fine-grained and intuitive attention pattern in the quality analysis.

V. DATA ANALYSIS

A. PARAMETER SENSITIVITY

The experimental results of this work show that the filtering evidence improves the performance for table-based fact verification. Since the amount of filtering evidence can be selected, it is necessary to evaluate its sensitivity for different parameter values. Considering that tables in TABFACT [13] dataset have at least 6 rows, and accounting for the addition of one summary row to each table, we set the parameter values of filtering evidence to 0-7. Figure 6(a) shows that different amounts of filtering evidence affect the final experimental results. It can be seen that for both the val and test sets of table-based fact verification task, the optimal amount of filtering evidence is between 5 and 6. As the amount of filtering evidence increases, DGMPF's performance on val and test sets rises and then decreases. This suggests that adding a certain amount of filtering evidence can improve the performance of DGMFP. We note that the model retaining less filtering evidence performs more erratically than those that retain more filtering evidence, which can infer that less filtering evidence is not sufficient for graph reasoning.

A parameter θ has been introduced in the program-like evidence graph for node pruning. Although the experimental results in this work show that it improves the performance of DGMFP, it is necessary to evaluate its sensitivity for different parameter values. Figure 6(b) shows the accuracy of the val and test sets, with probability θ values within the range of [0.1, 0.9]. It can be seen that for both the val and test sets of table-based fact verification task, the optimal value for θ is between 0.3 and 0.4. The result indicates that the use of node pruning method for many semantically independent nodes in the program-like evidence graph can improve the performance of DGMFP. It can be observed that setting the value of probability θ too high or too low is not conducive to improving the performance of DGMFP. Particularly, when the probability θ value is 0.9, the performance of DGMFP will even be lower than that of the baseline model LERGV [36].

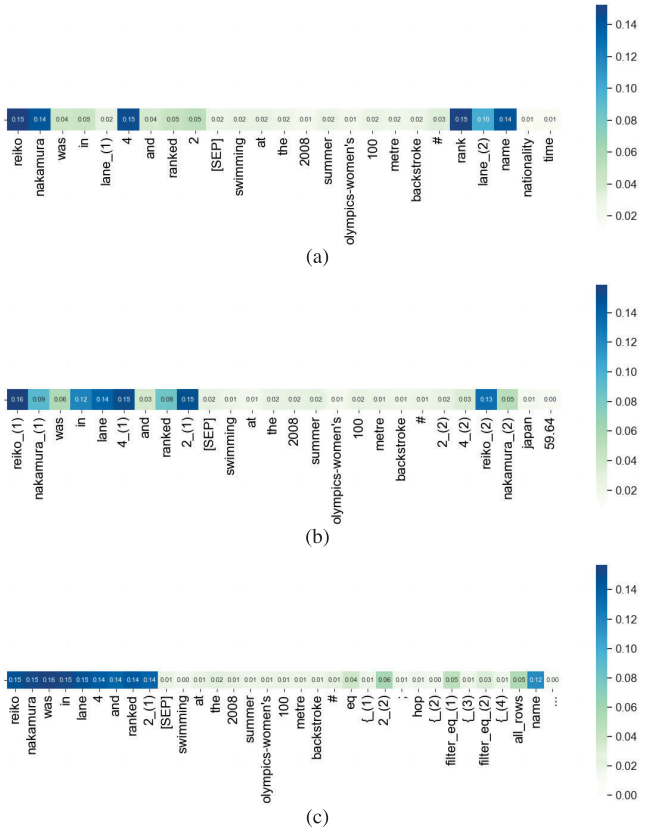

FIGURE 7. Kernel attention weights on evidence tokens. (a). The token attention distribution from the first evidence to the first evidence ($\alpha_i^{1\rightarrow1}$). (b). The token attention distribution from the first evidence to the second evidence ($\alpha_i^{2\rightarrow1}$). (c). The token attention distribution from the first evidence to the sixth evidence ($\alpha_i^{6\rightarrow1}$). Due to space limitations, we have omitted some of the evidence in the figure.

B. QUALITY ANALYSIS

We randomly select and introduce an example statement and the evidence retrieved in relation to the statement. Based on

the first evidence, the sentence level attention weights for six pieces of evidence in Table 5 show that the pieces of evidence (1), (2), and (6) are necessary for the given statement because they have more than two times higher sentence level attention weights than others. We mark the necessary evidence words in red. Figure 7 shows the token attention distribution from the first evidence to the first evidence ($\alpha_i^{1\to1}$), the first evidence to the second evidence ($\alpha_i^{2\to1}$), and the first evidence to the sixth evidence ($\alpha_i^{6\to1}$) in kernels. To get a graph of kernel attention weights on evidence tokens, we sort the same words, e.g. lane_(1), lane_(2). Although the same words exist in the statement and evidence, they represent different meanings.

The first evidence confirms the correlation among *reiko nakamura*, *rank*, *lane*, *name* and 4. The edge kernels from KGAT [48] accurately obtain the additional information from evidence (2): *reiko nakamura was in lane* 4, and the number 2, as well as supplementary information from evidence (6): *reiko nakamura ranked* 2. This evidence effectively fills the missing information needed to complete the entire reasoning. Interestingly, *reiko nakamura* and number 4 also receive increased attention in the evidence (2), thereby verifying that information in evidence (2) is related to the right person and number 4. The *reiko nakamura*, *in lane* 4 and number 2 also received increased attention in evidence (6), verifying that information in evidence (6) is related to the right person, *in lane* 4, and number 2. The kernel attention pattern is more intuitive and effective.

VI. DISCUSSION AND CONCLUSION

Table-based fact verification is an extremely demanding work because it essentially requests parsing table and statement structure and performing numerical and logical reasoning. This work provides some contributions. Firstly, some efforts have been made to utilize GNNs to construct graphs of tables and statements in table-based fact verification task [15], [16], [36]. However, there are no studies that attempt to construct both statement-evidence pair graphs and program-like evidence graphs, which can incorporate the semantic and symbolic information of the evidence and capture fine-grained relationships between and within the evidence. Secondly, to the best of our knowledge, no studies are made to solve the critical evidence selection problem by the neural matching kernel for this task. This work is inspired by the reasoning process of KGAT [48], we construct an evidence graph with statement-evidence pairs as nodes to conduct more fine-grained joint reasoning. Finally, this work improves the interpretability of table-based fact verification by employing neural matching kernels to select crucial evidence, rendering the reasoning process of table-based fact verification more specific.

This work proposes a double GAT [23] reasoning method based on filtering and program-like evidence for table-based fact verification, taking full advantage of more fine-grained reasoning and performing interpretability during the reasoning process. In particular, we retrieve the filtering evidence by using the table itself as part of the evidence and obtain the program-like evidence from logical forms according to a rule-based method, which aims to incorporate the semantic and symbolic information of evidence. Subsequently, we construct a fully-connected evidence graph with statement-evidence pairs as nodes and use the kernel in GNN to carry out feature propagation between nodes, which aims to conduct more fine-grained joint reasoning and increase the interpretability of table-based fact verification. We also construct a connected graph with all entities and functions in the program-like evidence as a node and use GAT to learn the importance between different nodes, thereby capturing more fine-grained relationships within the program-like evidence. Experimental results on TABFACT [13] show that DGMFP outperforms all baselines. Ablation studies and analyses further indicate the effect of DGMFP. We will further investigate how to introduce intermediate evidence into model to enrich evidence information. Applying the idea of the proposed model to tasks related to other semi-structured information, e.g., INFOTABS [31] and Tabular QA [54], is also a meaningful direction.

REFERENCES

[1] H. Saleh, A. Alharbi, and S. H. Alsamhi, "OPCNN-FAKE: Optimized convolutional neural network for fake news detection," *IEEE Access*, vol. 9, pp. 129471–129489, 2021.

[2] Q. Liao, H. Chai, H. Han, X. Zhang, X. Wang, W. Xia, and Y. Ding, "An integrated multi-task model for fake news detection," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5154–5165, Nov. 2022.

[3] N. Seddari, A. Derhab, M. Belaoued, W. Halboob, J. Al-Muhtadi, and A. Bouras, "A hybrid linguistic and knowledge-based analysis approach for fake news detection on social media," *IEEE Access*, vol. 10, pp. 62097–62109, 2022.

[4] Y. Xiao, Q. Yang, C. Sang, and Y. Liu, "Rumor diffusion model based on representation learning and anti-rumor," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 3, pp. 1910–1923, Sep. 2020.

[5] L. Ding, P. Hu, Z.-H. Guan, and T. Li, "An efficient hybrid control strategy for restraining rumor spreading," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 11, pp. 6779–6791, Nov. 2021.

[6] D. Wright, D. Wadden, K. Lo, B. Kuehl, A. Cohan, I. Augenstein, and L. Lu Wang, "Generating scientific claims for zero-shot scientific fact checking," 2022, *arXiv:2203.12990*.

[7] C. Chen, F. Cai, X. Hu, J. Zheng, Y. Ling, and H. Chen, "An entity-graph based reasoning method for fact verification," *Inf. Process. Manage.*, vol. 58, no. 3, May 2021, Art. no. 102472.

[8] W. Zhong, J. Xu, D. Tang, Z. Xu, N. Duan, M. Zhou, J. Wang, and J. Yin, "Reasoning over semantic-level graph for fact checking," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Seattle, WA, USA, Jul. 2020, pp. 6170–6180.

[9] C. Shao, P.-M. Hui, P. Cui, X. Jiang, and Y. Peng, "Tracking and characterizing the competition of fact checking and misinformation: Case studies," *IEEE Access*, vol. 6, pp. 75327–75341, 2018.

[10] C. Chen, F. Cai, X. Hu, W. Chen, and H. Chen, "HHGN: A Hierarchical Reasoning-based Heterogeneous Graph Neural Network for fact verification," *Inf. Process. Manage.*, vol. 58, no. 5, Sep. 2021, Art. no. 102659.

[11] Y. Wang, C. Xia, C. Si, B. Yao, and T. Wang, "Robust reasoning over heterogeneous textual information for fact verification," *IEEE Access*, vol. 8, pp. 157140–157150, 2020.

[12] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language under-standing," in *Proc. 57th Annu. Meet. Assoc. Comput. Linguist. Conf. (ACL)*, Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.

[13] W. Chen, H. Wang, J. Chen, Y. Zhang, H. Wang, S. Li, X. Zhou, and Y. W. Wang, "TabFact: A large-scale dataset for table-based fact verification," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, Addis Ababa, Ethiopia, Apr. 2020, pp. 1–14.

[14] W. Zhong, D. Tang, Z. Feng, N. Duan, M. Zhou, M. Gong, L. Shou, D. Jiang, J. Wang, and J. Yin, "LogicalFactChecker: Leveraging logical operations for fact checking with graph module network," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Seattle, WA, USA, Jul. 2020, pp. 6053–6065.

[15] Q. Shi, Y. Zhang, Q. Yin, and T. Liu, "Learn to combine linguistic and symbolic information for table-based fact verification," in *Proc. 28th Int. Conf. Comput. Linguistics*, Barcelona, Spain, Dec. 2020, pp. 5335–5346.

[16] X. Yang, F. Nie, Y. Feng, Q. Liu, Z. Chen, and X. Zhu, "Program enhanced fact verification with verbalization and graph attention network," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Punta Cana, Dominican Republic, Nov. 2020, pp. 7810–7825.

[17] F. Wang, K. Sun, J. Pujara, P. Szekely, and M. Chen, "Table-based fact verification with salience-aware learning," in *Proc. Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, Pontacana, Dominican Republic, Nov. 2021, pp. 4025–4036.

[18] X. Yang and X. Zhu, "Exploring decomposition for table-based fact verification," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, Pontacana, Dominican Republic, Nov. 2021, pp. 1045–1052.

[19] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, and J. Eisenschlos, "TaPas: Weakly supervised table parsing via pre-training," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Seattle, WA, USA, Jul. 2020, pp. 4320–4333.

[20] J. Eisenschlos, S. Krichene, and T. Müller, "Understanding tables with intermediate pre-training," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, Punta Cana, Dominican Republic, Nov. 2020, pp. 281–296.

[21] Q. Liu, B. Chen, J. Guo, Z. Lin, and J. Lou, "TAPEX: Table pre-training via learning a neural SQL executor," in *Proc. 10th Int. Conf. Learn. Represent. (ICLR)*, Vienna, Austria, Apr. 2022, pp. 1–18.

[22] G. Zhao, P. Yang, and Y. Yao, "RERG: Reinforced evidence reasoning with graph neural network for table-based fact verification," *Appl. Intell.*, vol. 53, no. 10, pp. 12308–12323, May 2023.

[23] C. Xiong, Z. Dai, J. Callan, Z. Liu, and R. Power, "End-to-end neural ad-hoc ranking with kernel pooling," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Tokyo, Japan, Aug. 2017, pp. 55–64.

[24] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, Apr. 2018, pp. 1–12.

[25] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced LSTM for natural language inference," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Vancouver, BC, Canada, Jul. 2017, pp. 1657–1668.

[26] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol. (NAACL-HLT)*, New Orleans, LA, USA, Jun. 2018, pp. 2227–2237.

[27] Y. Tay, A. T. Luu, and S. C. Hui, "Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Brussels, Belgium, Oct. 2018, pp. 1565–1575.

[28] R. Ghaeini, S. A. Hasan, V. Datla, J. Liu, K. Lee, A. Qadir, Y. Ling, A. Prakash, X. Fern, and O. Farri, "DR-BiLSTM: Dependent reading bidirectional LSTM for natural language inference," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol. (NAACL-HLT)*, New Orleans, LA, USA, Jun. 2018, pp. 1460–1469.

[29] Q. Chen, X. Zhu, Z.-H. Ling, D. Inkpen, and S. Wei, "Neural natural language inference models enhanced with external knowledge," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics Conf. (ACL)*, Melbourne, VIC, Australia, Jul. 2018, pp. 2406–2417.

[30] S. Jo, I. Trummer, W. Yu, and N. Mehta, "AggChecker: A fact-checking system for text summaries of relational data sets," in *Proc. 45th Very Large Data Bases Conf. (VLDB)*, Los Angeles, CA, USA, Aug. 2019, pp. 1938–1941.

[31] V. Gupta, M. Mehta, P. Nokhiz, and V. Srikumar, "INFOTABS: Inference on tables as semi-structured data," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Seattle, WA, USA, Jul. 2020, pp. 2309–2324.

[32] K. Zhang, G. Lv, L. Wu, E. Chen, Q. Liu, H. Wu, X. Xie, and F. Wu, "Multilevel image-enhanced sentence representation net for natural language inference," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 6, pp. 3781–3795, Jun. 2021.

[33] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi, "A corpus for reasoning about natural language grounded in photographs," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Florence, Italy, Jul. 2019, pp. 6418–6428.

[34] H. Zhang, Y. Wang, S. Wang, X. Cao, F. Zhang, and Z. Wang, "Table fact verification with structure-aware transformer," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Punta Cana, Dominican Republic, Nov. 2020, pp. 1624–1629.

[35] S. Ou and Y. Liu, "Learning to generate programs for table fact verification via structure-aware semantic parsing," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Dublin, Ireland, May 2022, pp. 7624–7638.

[36] Q. Shi, Y. Zhang, Q. Yin, and T. Liu, "Logic-level evidence retrieval and graph-based verification network for table-based fact verification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Punta Cana, Dominican Republic, Nov. 2021, pp. 175–184.

[37] J. Yang, A. Gupta, S. Upadhyay, L. He, R. Goel, and S. Paul, "TableFormer: Robust transformer modeling for table-text encoding," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Dublin, Ireland, May 2022, pp. 528–537.

[38] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[39] B. M. Oloulade, J. Gao, J. Chen, T. Lyu, and R. Al-Sabri, "Graph neural architecture search: A survey," *Tsinghua Sci. Technol.*, vol. 27, no. 4, pp. 692–708, Aug. 2022.

[40] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 109–127, Apr. 2021.

[41] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, Apr. 2021.

[42] M. Yang, R. Wang, Y. Shen, H. Qi, and B. Yin, "Breaking the expression bottleneck of graph neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5652–5664, Jun. 2023.

[43] X. Wang, Y. Wu, A. Zhang, F. Feng, X. He, and T.-S. Chua, "Reinforced causal explainer for graph neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2297–2309, Feb. 2023.

[44] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–14.

[45] S. Ma, J.-W. Liu, X. Zuo, and W.-M. Li, "Heterogeneous graph gated attention network," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Shenzhen, China, Jul. 2021, pp. 1–6.

[46] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf.*, Taiwan, Apr. 2020, pp. 2704–2710.

[47] X. Fu, J. Zhang, Z. Meng, and I. King, "MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proc. Web Conf.*, Taiwan, Apr. 2020, pp. 2331–2341.

[48] Z. Liu, C. Xiong, M. Sun, and Z. Liu, "Fine-grained fact verification with kernel graph attention network," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Seattle, WA, USA, Jul. 2020, pp. 7342–7351.

[49] Z. Dai, C. Xiong, J. Callan, and Z. Liu, "Convolutional neural networks for soft-matching n-grams in ad-hoc search," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Marina Del Rey, CA, USA, Feb. 2018, pp. 126–134.

[50] S. MacAvaney, A. Yates, A. Cohan, and N. Goharian, "CEDR: Contextualized embeddings for document ranking," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Paris, France, Jul. 2019, pp. 1101–1104.

[51] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Jun. 2018. *Improving Language Understanding by Generative Pre-Training*. [Online]. Available: https://openai.com/research/language-unsupervised

[52] Y. Qiao, C. Xiong, Z. Liu, and Z. Liu, "Understanding the behaviors of BERT in ranking," 2019, *arXiv:1904.07531*.

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008.

[54] W. Chen, H. Zha, Z. Chen, W. Xiong, H. Wang, and W. Y. Wang, "HybridQA: A dataset of multi-hop question answering over tabular and textual data," in *Proc. Findings Assoc. Comput. Linguistics, (EMNLP)*, Punta Cana, Dominican Republic, Nov. 2020, pp. 1026–1036.

HONGFANG GONG received the M.E. degree in computer application and the Ph.D. degree in computer science and technology from Hunan University, China, in 2004 and 2018, respectively. He is currently a Full Professor of mathematics and statistics with the Changsha University of Science and Technology, Changsha, China. His current research interests include machine learning, bigdata analysis, autonomous driving, and cyber-physical systems (CPS).

CAN WANG received the B.S. degree in mathematics and statistics from the Changsha University of Science and Technology, Changsha, China, in 2021, where she is currently pursuing the M.S. degree with the School of Mathematics and Statistics. Her current research interests include natural language processing, deep learning, and machine learning.

XIAOFEI HUANG received the B.S. degree in mathematics and statistics from the Changsha University of Science and Technology, Changsha, China, in 2020, where he is currently pursuing the M.S. degree with the School of Mathematics and Statistics. His current research interests include computerized medical diagnosis, pattern recognition, deep learning, and machine learning.