# Heterogeneous Graph Reasoning for Fact Checking over Texts and Tables

**Haisong Gong[1,2], Weizhi Xu[3], Shu Wu[1,2*], Qiang Liu[1,2], Liang Wang[1,2]**
[1]Center for Research on Intelligent Perception and Computing
State Key Laboratory of Multimodal Artificial Intelligence Systems
Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]ByteDance Inc.
gonghaisong2021@ia.ac.cn, jason_xu1998@163.com, {shu.wu, qiang.liu, wangliang}@nlpr.ia.ac.cn

## Abstract

Fact checking aims to predict claim veracity by reasoning over multiple evidence pieces. It usually involves evidence retrieval and veracity reasoning. In this paper, we focus on the latter, reasoning over unstructured text and structured table information. Previous works have primarily relied on fine-tuning pretrained language models or training homogeneous-graph-based models. Despite their effectiveness, we argue that they fail to explore the rich semantic information underlying the evidence with different structures. To address this, we propose a novel word-level Heterogeneous-graph-based model for Fact Checking over unstructured and structured information, namely HeterFC. Our approach leverages a heterogeneous evidence graph, with words as nodes and thoughtfully designed edges representing different evidence properties. We perform information propagation via a relational graph neural network, facilitating interactions between claims and evidence. An attention-based method is utilized to integrate information, combined with a language model for generating predictions. We introduce a multitask loss function to account for potential inaccuracies in evidence retrieval. Comprehensive experiments on the large fact checking dataset FEVEROUS demonstrate the effectiveness of HeterFC. Code will be released at: https://github.com/Deno-V/HeterFC.

## Introduction

Fact checking, or fact verification, predicts claim veracity using evidence. This task has practical applications in various domains like politics (Liu et al. 2023; Xu et al. 2022), news media (Zellers et al. 2019), public health (Naeem and Bhatti 2020; Krause et al. 2020), and science (Wright et al. 2022; Wadden et al. 2020), attracting extensive research. Prior efforts predominantly address unstructured fact checking, handling evidence and claims as plain text (Thorne et al. 2018; Wang 2017; Xu et al. 2023). However, real-world contexts often involve structured data like tables, creating a pressing need for fact checking across unstructured and structured information.

Fact checking mainly involves evidence retrieval and veracity reasoning, which are two independent tasks (Aly et al. 2021). The aim of evidence retrieval is to retrieve as much the claim-related evidence (golden evidence) as possible; the target of veracity reasoning is to precisely predict the veracity of claim based on the retrieved evidence from the former stage. In this paper, we mainly focus on the design of veracity reasoning model.

Previous veracity reasoning models (Aly et al. 2021; Kotonya et al. 2021; Funkquist 2021; Hu et al. 2022; Bouziane et al. 2021; Zhao et al. 2020; Wu et al. 2022) have primarily relied on fine-tuning pretrained language models or training homogeneous-graph-based models. In the fine-tuning approach, they firstly transform tables into sentences via some heuristic linearizing rules. Then, a pretrained language model (PLM), such as RoBERTa (Liu et al. 2019), is fine-tuned by concatenating all pieces of evidence as the input. In the homogeneous-graph-based approach, they construct a homogeneous fully-connected evidence graph where each node is treated as a piece of evidence. After that, a graph neural network (GNN) is utilized to propagate neighborhood information, which enables the semantic representations of different pieces of evidence to be aggregated.

While effective, existing approaches exhibit two key weaknesses. *Firstly*, fact checking necessitates capturing semantics among various evidence pieces, demanding intricate modeling of evidence relationships. Transformer-based methods often fall short as they merely concatenate evidence or deal with point-wise claim-evidence pairs, insufficiently exploring complex evidence interconnections. *Secondly*, prevalent graph-based methods construct sentence-level graphs with claim-evidence pairs as nodes, employing Pre-trained Language Models (PLMs) for node representations (Zhou et al. 2019; Liu et al. 2020; Kotonya et al. 2021). Although these models perform well in conventional fact checking, they falter in scenarios involving both structured and unstructured information. This is due to the limitations of sentence-level graphs in capturing fine-grained details such as entities and time phrases. Furthermore, assuming uniform relationships between node pairs overlooks the diverse properties inherent in table and sentence evidence.

To tackle the aforementioned problems, we propose a novel word-level **Heter**ogeneous-graph-based model for **F**act **C**hecking over unstructured and structured information, HeterFC for brevity. Specifically, we first construct a graph where nodes represent words in all pieces of evidence, thereby achieving a granularity at word-level. Then, to capture the different relationships in structured and unstructured

*Corresponding Author

information, we specially design three kinds of connections on the graph, namely intra-sentence edges, intra-table edges, and inter-evidence edges. In detail, intra-sentence edges and intra-table edges are added between each word and its local contextual words in a fix-sized sliding window. Inter-evidence edges are between the same keyword appearing in several pieces of evidence, which allows the important information be aggregated across the evidence via these edges. We employ the relational graph convolutional network (R-GCN) to perform neighborhood propagation and readout the representations of each evidence, followed by an attention mechanism to obtain information from all pieces of evidence. Combined with a language model, we get the final veracity prediction. To train the model, in addition to cross-entropy loss for the claim veracity, we propose a multitask loss to assist the model in discerning between valid and invalid evidence, thereby enhancing the overall performance of the model.

In a nutshell, our main contributions can be listed as follows,

- We figure out the inapplicability of previous homogeneous-graph-based methods in the traditional unstructured fact checking and analyze the underlying possible reasons.
- We propose a novel word-level heterogeneous-graph-based model, namely HeterFC, which is specially designed for fact checking over unstructured and structured information.
- Extensive experiments on the large-scale FEVEROUS fact-checking dataset, which includes both structured and unstructured information, have demonstrated the effectiveness of our proposed model over several baselines.

## Related Work

### Fact Checking Over Unstructured Information

Fact checking, a form of natural language inference (NLI), involves predicting claim veracity by reasoning over multiple evidence pieces. Existing methods fall into two categories. The first category uses pretrained language models (PLMs), fine-tuning them for fact checking. They organize the input by concatenating all evidence and claim into a single sentence (Aly et al. 2021), or processing each evidence separately with the claim using aggregation techniques (Soleimani, Monz, and Worring 2020; Gi, Fang, and Tsai 2021). The second category employs graph neural networks (GNNs) to capture complex semantic interactions. GEAR (Zhou et al. 2019) constructs sentence-level fully-connected evidence graphs with GNNs. Zhao et al. (2020) use Transformer-XH for graph representation, while Liu et al. (2020) introduce KGAT with node and edge kernels. DREAM (Zhong et al. 2020) incorporates semantic role labeling for fine-grained semantic graphs. Chen et al. (2022) propose EvidenceNet with symmetrical interaction attention and gating on sentence-level evidence graphs.

### Fact Checking Over Mixed-type Information

Unlike unstructured fact checking, fact checking over both structured and unstructured information requires handling a combination of structured tables and unstructured text. Existing methods include table linearization, where tables are converted into text, potentially losing structural information (Gi, Fang, and Tsai 2021; Kotonya et al. 2021; Malon 2021). Another approach is to combine sentence and table evidence using models like TAPAS (Funkquist 2021; Bouziane et al. 2021; Hu et al. 2022). In graph-based methods, previous works focused on homogeneous sentence-level evidence graphs (Kotonya et al. 2021). In contrast, our approach introduces word-level nodes and heterogeneous relations, making it more suitable for fact checking over structured and unstructured information.

### Heterogeneous Graph Neural Networks

Heterogeneous graph neural networks (heterGNNs) are specialized GNNs designed for neighborhood propagation on graphs with different types of edges. R-GCN (Schlichtkrull et al. 2018) is a representative heterGNN that assigns trainable weight matrices to each relation. HeterGNNs have been successfully applied in various domains including recommender systems (Fan et al. 2019; Zhao et al. 2017; Yan et al. 2021) and question answering (Yu et al. 2019; Sun et al. 2018).

## Method

In this section, we introduce the proposed method HeterFC in details. The overall framework is shown in Figure 1.

### Task Formulation

The aim of multi-structured fact checking is to predict the veracity of a claim according to several pieces of evidence, which contains both tables and texts. Mathematically, given a claim $c$ and a retrieved evidence set $\mathcal{E} = \{e_1, e_2, \ldots, e_M\}$, where each piece of evidence $e_i$ represents either a sentence or a table cell, we need to propose a model $\hat{\mathbf{p}} = f(c, \mathcal{E})$ to output the predicted veracity $\hat{\mathbf{p}}$.

### Word-level Evidence Graph Construction

In this part, we elaborate the design of nodes and edges on the evidence graph.

**Node Construction** We treat each word in the evidence as a node on the evidence graph, since it contains more fine-grained semantic information than the sentence.[1] To achieve this, we employ a PLM to process each claim-evidence pair to build the initial node representations.

The sentence evidence can be directly treated as input to a PLM, however, since tables have a distinct structure from sentences, table evidence can't be directly processed by PLM. To address this, we adopt the idea of cell linearization proposed in (Hu et al. 2022). Specifically, each piece of table evidence is transformed into either "*<column header> for <row header> is <cell value>* " if it belongs to a general table or "*<column header> :<row header> of <title> is <cell value>* " if it is from an infobox-type table.

[1]We have tried token-level node construction, which is more fine-grained than the word-level nodes. However, it is less effective and more analysis can be seen in the experimental section.

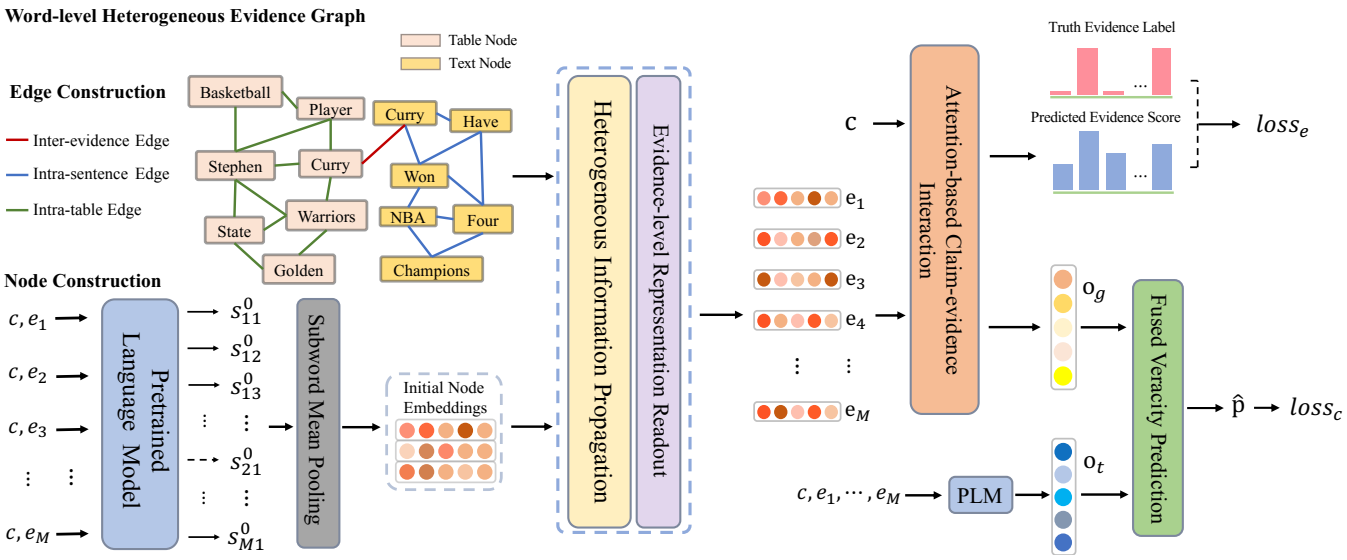


Figure 1: Architecture of HeterFC. Inputs are claim $c$ and an evidence set $\mathcal{E}$. There are five main parts: 1) *Word-level evidence graph construction.* Initial node embeddings are obtained by using PLM and subword mean pooling. Three types of edges are designed for the heterogeneous connection. 2) *Heterogeneous information propagation.* R-GCN is used to perform neighborhood aggregation on the word-level evidence graph. 3) *Evidence-level representation readout.* Evidence representations are obtained by pooling over subgraphs corresponding to each piece of evidence. 4) *Attention-based claim-evidence interaction.* Graph representation $\mathbf{o}_g$ is generated by claim-guided evidence combination. A supervised loss item, $Loss_e$, is computed based on the attention assignment. 5) *Fused veracity prediction.* The claim and evidence are concatenated and fed into PLM to obtain $\mathbf{o}_t$, which, when combined with the graph representation $\mathbf{o}_g$, forms the final representation. A fully-connected network takes the representation as input and generates the prediction $\hat{\mathbf{p}}$. HeterFC is trained using classification loss $Loss_c$ and assisted $Loss_e$.

By applying cell linearization technique to table evidence, we are able to feed each claim-evidence pair to a PLM and obtain the embedding of each token (subword) in both table evidence and sentence evidence. It is noteworthy that though claim-evidence pair is fed into PLM, the subwords in claim are excluded and only the subwords in evidence are kept. This can be expressed mathematically as follows:

$$\mathbf{S}_i^0 = \text{PLM}([c, e_i]) \tag{1}$$

$$\mathbf{S}_i^0 := \mathbf{s}_{i0}^0 \,||\, \mathbf{s}_{i1}^0 \,||\, \ldots \,||\, \mathbf{s}_{ij}^0 \tag{2}$$

where PLM is a RoBERTa model here, following previous works (Aly et al. 2021). $c$ denotes the claim and $e_i$ is the $i$-th evidence. $\mathbf{s}_{ij}^0$ represents the embedding of the $j$-th subword in the $i$-th evidence and $||$ is the concatenation of vectors.

Then, we generate the embedding of a whole word via computing the mean of the embeddings of its corresponding subwords. By following this approach, we can obtain the word embedding matrix $\mathbf{H}^0$ for the all pieces of evidence by the mentioned way, where $\mathbf{H}^0$ is the initial node representations since we treat each word in the evidence as a node.

**Edge Construction** After constructing word-level nodes, the next step is to design the connections among them. The simplest way is to construct a fully-connected graph, where each node shares an edge with every other node on the graph. However, this approach may bring too much noise since only part of information is related and beneficial for a node. Especially, the input evidence inevitably contains some unrelated information due to the error of the retrieval model. Therefore, a more elaborate design is required in this task.

We carefully design three types of edges to capture the heterogeneous information among several pieces of evidence. Specifically, the three types of edges are named inter-evidence edges, intra-sentence edges and intra-table edges. The illustration of such edges is shown in Figure 1 where different type of edge is illustrated with different color in the graph, and we introduce them in detail as follows,

**Inter-evidence edges $r_e$.** Aggregating relevant information from multiple pieces of evidence is crucial for accurate claim veracity prediction. In fact verification scenarios involving both texts and tables, it is common for the same entity to be referenced in different types of evidence. Thus, integrating information from both sources is necessary for a comprehensive understanding. To capture the multi-hop relationship between evidence, we construct edges connecting the same word in different pieces of evidence. By propagating information along these edges, we can capture the flow of information between related evidence. To ensure the quality of inter-evidence edges, we filter out stop words like "is," "of," and "the" to prevent constructing edges between frequently used but insignificant words.

**Intra-sentence edges $r_s$.** A word in the sentence is usually associated with its local context for understanding the semantics (Mikolov et al. 2013). Therefore, we adopt this traditional technique and employ a sliding window with a fix size $w$ to cover the local context. In this way, each word in the center of the window has edges with the rest of words

in the window, through which each word is connected with its context on the graph. Thus, the contextual information can be aggregated via a one-layer GNN, which is beneficial for learning the sentence-level semantics.

**Intra-table edges $r_t$.** Tables have a completely different structure compared with sentences. The header and the cell form a key-value relationship. This structure is distinct from that of a sentence, which contains many stop words (is, the, etc.) to ensure fluency. Based on the analysis above, we assign edges among the cell, its row header, column header and its page title. To achieve this, we reuse the cell linearization method mentioned in the node construction section to transform each table cell into sequence and utilize the fix-sized window again to connect words, just like building intra-sentence edges.

Eventually, we construct a word-level evidence graph $\mathcal{G}$ via the aforementioned design, which involves three different relations $\mathcal{R} = \{r_s, r_t, r_e\}$. Next, we introduce the main model architecture.

## Heterogeneous Information Propagation

The constructed evidence graph includes various edges, making homogeneous GNNs unsuitable. Thus, we employ relational graph convolutional networks (R-GCN) to capture distinct node relations. Formally, it can be written as,

$$\mathbf{h}_i^{l+1} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_{i,r}}\mathbf{W}_r^l\mathbf{h}_j^l + \mathbf{W}_0^l\mathbf{h}_i^l\right) \tag{3}$$

where $\mathcal{N}_i^r$ denotes the one-hop neighbors of $i$-th node that have edges of relation $r$ and $\mathbf{h}_i^l = \mathbf{H}^l[i,:] \in \mathbb{R}^{1*d^l}$ ($d^l$ is the embedding dimension). $c_{i,r} = |\mathcal{N}_i^r|$ is the normalized term, $\sigma$ is the Sigmoid activation function, and $\mathbf{W}_*^l$ are learnable weight matrices in the $l$-th layer.

After one-step neighborhood propagation via Eq. (3), we obtain the contextual information in one-hop neighborhood. By stacking $k$ layers of R-GCN, we can aggregate the information from $k$-hop neighborhood, where $k$ is a hyperparameter that will be discussed in the experimental section. We denote the final output of $k$-layer R-GCN as $\mathbf{H} \in \mathbb{R}^{N*d}$, where $N$ and $d$ are the number of nodes and the embedding dimension, respectively.

## Evidence-level Representation Readout

The word-level node representations $\mathbf{H}$ are processed using a readout module to generate evidence-level embeddings. Inspired by the work in the graph classification (Ying et al. 2018), we employ both max pooling and mean pooling over the words within each evidence to produce its representation.

$$\mathbf{e}_m = \max(\mathbf{H}_{i:j}) \,||\, \text{mean}(\mathbf{H}_{i:j}) \tag{4}$$

where $\mathbf{H}_{i:j} \in \mathbb{R}^{(j-i+1)*d}$ denotes the segment of $\mathbf{H}$ spanning rows $i$ to $j$, corresponding to nodes from the $m$-th evidence. The pooling strategies are applied along the first dimension so as to obtain the embedding of each evidence $\mathbf{e}_m \in \mathbb{R}^{1*2d}$.

In contrast to the graph classification context where pooling is performed over all graph nodes, this evidence-wise readout scheme proves beneficial for our task due to the varying significance of different evidence pieces.

## Attention-based Claim-evidence Interaction

Evidence representations $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M\}$ have varying significance for claim verification. For example, the imperfect upstream evidence retrieval model may recall some redundant, claim-unrelated evidence. Such evidence should be ignored in the reasoning model. According to this observation, we introduce an attention-based claim-evidence interaction module. In detail, we compute the importance score $\alpha_m$ for the $m$-th evidence regarding the claim based on an attention mechanism,

$$\mathbf{c} = \text{PLM}(c) \tag{5}$$

$$g_m = \mathbf{W}_{a1}\left(\text{ReLU}\left(\mathbf{W}_{a0}\left(\mathbf{c}\|\mathbf{e}_m\right)\right)\right) \tag{6}$$

$$\alpha_m = \text{softmax}\left(g_m\right) = \frac{\exp\left(g_m\right)}{\sum_{i=1}^{M}\exp\left(g_i\right)} \tag{7}$$

where PLM denotes a RoBERTa model encoding the claim into embeddings. PLMs in this module and node construction share the same weights for efficiency. We take the [CLS] representation as the claim embedding $\mathbf{c} \in \mathbb{R}^{1*d}$. We then obtain the graph representation of the whole evidence set $\mathbf{o}_g \in \mathbb{R}^{1*2d}$ via the attention-weighted summation of all evidence.

$$\mathbf{o}_g = \sum_{i=1}^{M}\alpha_i\mathbf{e}_i \tag{8}$$

## Fused Veracity Prediction

The graph construction method utilized here effectively captures the interaction between evidence, but it also weakens the integrity of the claim and evidence paragraphs. We found that relying solely on the graph representation may cause the model to overlook phrases such as negation words. Therefore, in addition to solely using the graph representation, we generate an assisted representation $\mathbf{o}_t$ by feeding the linearized claim and evidence sequences directly to a PLM. Then, the graph representation $\mathbf{o}_g$ is concatenated with the assisted representation $\mathbf{o}_t$ and fed into a multi-layer fully-connected network, followed by softmax normalization, to produce the final prediction $\hat{\mathbf{p}} \in \mathbb{R}^{1*C}$, where $C$ denotes the number of class.

$$\mathbf{o}_t = \text{PLM}([c, e_1, e_2, \cdots, e_M]) \tag{9}$$

$$\hat{\mathbf{p}} = \text{softmax}(\text{MLP}(\mathbf{o}_g\|\mathbf{o}_t)) \tag{10}$$

## Model Training

The cross-entropy objective is utilized to compute the veracity classification loss,

$$Loss_c = -\sum_{c=1}^{C}\mathbf{y}_c\log\left(\hat{\mathbf{p}}_c\right) \tag{11}$$

where $\mathbf{y}$ denotes the one-hot label vector (e.g., [1, 0, 0] represents that the ground truth is the first class).

To counteract the effects of the upstream evidence retrieval model's imperfections, we utilize the advantageous

attributes of attention mechanisms in the attention-based claim-evidence interaction module. Specifically, we calculate a predicted evidence score $s_m$ for each evidence $e_m$ using the sigmoid function, and then compare it with the truth evidence label to generate a binary classification loss. The truth evidence label indicates whether an evidence is relevant for verifying the claim. The assisted loss is obtained by averaging all binary classification losses over the evidence set $\mathcal{E}$:

$$s_m = \text{sigmoid}(g_m) \tag{12}$$

$$Loss_e = \frac{1}{M}\sum_{i=1}^{M}[t_m \log(s_m) + (1-t_m)\log(1-s_m)] \tag{13}$$

where $g_m$ is a intermediate result in computing attention scores in Equation 6, $t_m$ denotes the truth evidence label for evidence $e_m$.

For training the whole model, we combined the loss item $loss_c$ and $loss_e$ with a hyperparameter $\beta$, which will be discussed in the experimental section.

$$Loss = Loss_c + \beta Loss_e \tag{14}$$

## Experiment

In this section, we conduct comprehensive experiments to answer the following research questions:

- RQ1: How does the proposed method HeterFC perform compared with existing baselines?
- RQ2: How is the word-level graph compared with the sentence-level graph and the token-level graph?
- RQ3: How does the model perform with different design of edges?
- RQ4: Which part of the model contributes most to the final result apart from the graph construction?
- RQ5: How does the model performance change with different values of hyper-parameters?
- RQ6: How does the model perform with different retrieval model?

### Experimental Setups

**Dataset** Following prior research, we utilize the extensive FEVEROUS dataset in our experiments (Aly et al. 2021). The FEVEROUS task involves finding relevant evidence before claim verification. Each claim is manually annotated with labels: Supported, Refuted, or Not Enough Information, and paired with a corresponding golden evidence set. The dataset is divided into a training set of 71,291 claims, a development set of 7,890 claims, and a blind test set available on an online judging system[2]. The evidence sets include 38,941 with sentences only, 30,574 with tables only, and 25,395 with both sentences and tables.

[2]https://eval.ai/web/challenges/challenge-page/1091/overview

**Metrics** Two metrics gauge the model's performance: the Feverous score and label accuracy. Label accuracy solely evaluates claim veracity classification accuracy, whereas the Feverous score assesses verdict prediction accuracy and correct retrieval of the golden evidence set. This score quantifies instances where the golden evidence set is successfully retrieved and the verdict is correctly predicted. The Feverous score is a comprehensive metric that assesses both the retrieval system and veracity reasoning model's performance.

**Baselines**

- **RoBERTa-Pair**$_{\text{mean/max}}$ Utilizes RoBERTa as backbone. Concatenates claim with each evidence separately to form sentences. Mean or max pooling is applied over embeddings of all evidence for final prediction.
- **RoBERTa-Concat** Concatenates claim with all evidence using [SEP] as separator. [CLS] token's representation is used for classification.
- **GEAR** (Zhou et al. 2019) Homogeneous coarse-grained graph-based method. Treats each claim-evidence pair as node in a fully-connected evidence graph. Graph convolutional network and evidence aggregator are used.
- **KGAT** (Liu et al. 2020) Kernel graph attention model. Similar graph construction as GEAR. Employs node and edge kernels for fine-grained evidence propagation.
- **DCUF** (Hu et al. 2022) Dual-channel approach. Converts evidence to sentence form and table form. Utilizes RoBERTa for sentence form, and TAPAS for table form. Integrates both channels for veracity prediction.

**Evidence Retrieval** The primary objective of our paper is to present a model for veracity reasoning. For equitable comparison, we employ the evidence retrieval model from (Hu et al. 2022) for all tested models. We retrieve up to 150 relevant Wikipedia pages per claim using entity-matching and TF-IDF. The top 5 pages are selected based on SBERT[3] and BM25 rankings. Tables within these pages are flattened, and up to 5 sentences and 3 tables are selected per claim using DrQA (Chen et al. 2017). Relevant table cells are identified using a cell selector. Each claim's retrieved evidence set comprises up to 5 sentences and 25 table cells, as FEVEROUS task requirements.

**Implementation Details** To boost our model's ability to extract relevant details from noisy evidence, we augment each claim with two sets of evidence: the golden evidence set and the retrieved evidence set. Claims with the golden evidence set have all truth evidence labels set to positive. while for claims with the retrieved evidence set, only evidence shared with the golden evidence set receives a positive label. We consistently apply this augmentation to all baseline models. This approach differs from GEAR and KGAT's original technique for the FEVER task (Thorne et al. 2018), which focuses on fact verification over unstructured text evidence. We use the Adam optimizer (Kingma and Ba 2015) with learning rates of 1e-5 for language model parameters

[3]https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

| | Model | Dev | | Test | |
|---|---|---|---|---|---|
| | | Feverous | Label Accuracy | Feverous | Label Accuracy |
| Transformer-based | RoBERTa-Pair$_{\text{mean}}$ | 0.3452 | 0.7117 | 0.3231 | 0.6074 |
| | RoBERTa-Pair$_{\text{max}}$ | 0.3550 | 0.7207 | 0.3347 | 0.6190 |
| | RoBERTa-Concat | 0.3549 | 0.7221 | 0.3344 | 0.6194 |
| | DCUF (Hu et al. 2022) | 0.3577 | 0.7291 | 0.3397 | 0.6321 |
| Graph-based | GEAR (Zhou et al. 2019) | 0.2640 | 0.5859 | 0.2483 | 0.4964 |
| | KGAT (Liu et al. 2020) | 0.3293 | 0.6844 | 0.3043 | 0.5797 |
| | HeterFC | **0.3714** | **0.7352** | **0.3476** | **0.6329** |

Table 1: Comparison of models on Feverous task

and 1e-3 for others, employing a linear scheduler with a 20% warm-up rate. RoBERTa-Large serves as the PLM, with window size $w$ and R-GCN layer count $k$ set to 2. All experiments run on a server with an AMD EPYC 7742 (256) @ 2.250GHz CPU and one NVIDIA A100 GPU.

## Overall Performance (RQ1)

We compare our proposed HeterFC against various baselines, spanning transformer-based and graph-based models. In Table 1, HeterFC consistently outperforms all strong baselines across metrics and datasets, highlighting its superiority. Key observations from these results are as follows:

- Among RoBERTa-based models, RoBERTa-Pair$_{\text{mean}}$ lags, while the other two exhibit similar performance levels. This may stem from RoBERTa-Pair's limited ability to capture diverse relationships by processing evidence pieces separately. RoBERTa-Concat processes evidence together, but struggles to distinguish between golden and noisy evidence, affecting its performance. Thus, direct PLM use falls short, emphasizing the need for task-specific design.
- DCUF, the top-performing transformer-based method, incorporates RoBERTa-Concat alongside TAPAS, contributing to its superior performance.
- Graph-based models GEAR and KGAT, stemming from FEVER task, exhibit suboptimal performance due to task and data differences. Comparing them with HeterFC$_{\text{graph}}$ (see Table 2), which is a purely graph-based model, HeterFC$_{\text{graph}}$ outperforms substantially, affirming the efficacy of our hybrid fact verification design.

## Ablation Study and Model Variants

**Comparison of the Graph Granularity (RQ2)** To validate our word-level graph design, we experiment with HeterFC against its sentence-level graph variant (**HeterFC$_{\text{sent}}$**) and token-level graph variant (**HeterFC$_{\text{token}}$**). For sentence-level graphs, each claim-paired sentence (tables linearized into sentences) constitutes a node. Nodes from the same claim interconnect to form a fully-connected evidence graph, initialized with PLM [CLS] token embeddings. For token-level graphs, we omit subword mean pooling, resulting in nodes representing subwords.

From Figure 2, HeterFC surpasses both model variants across metrics. HeterFC$_{\text{sent}}$ significantly lags behind HeterFC and HeterFC$_{\text{token}}$. This gap suggests that HeterFC$_{\text{sent}}$'s coarser granularity struggles to capture nuanced semantic relationships among evidence pieces. Comparing HeterFC with HeterFC$_{\text{token}}$, HeterFC maintains better Feverous score and slightly edges in label accuracy. This suggests that the word-level graph in HeterFC enhances performance due to more precise inter-evidence connections than the token-level graph. In token-level graphs, shared subwords may lead to extraneous inter-evidence links among unrelated words (e.g., "interesting" and "thing" sharing "ing"). This results in noisy connections due to non-keyword shared subwords, undermining overall performance.

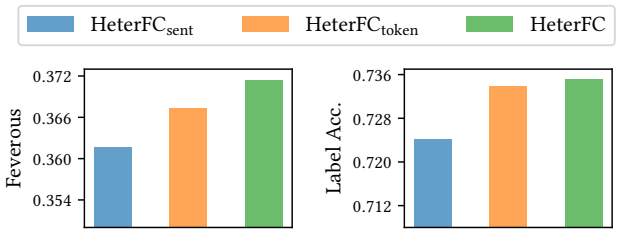


Figure 2: The performance comparison among models with different graph granularity. HeterFC$_{\text{sent}}$ represents the variant with sentence-level graph, while HeterFC$_{\text{token}}$ represents the variant with token-level graph.

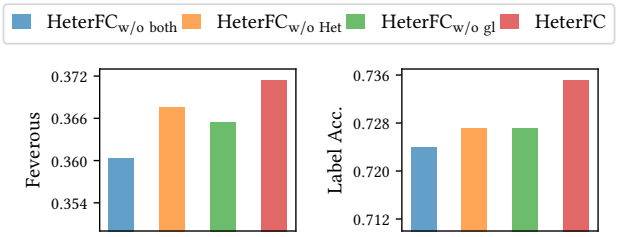


Figure 3: The performance comparison between the proposed model HeterFC and its three variants: HeterFC$_{\text{w/o gl}}$ ignores global and local connection designs. HeterFC$_{\text{w/o Het}}$ ignores heterogeneous relations. HeterFC$_{\text{w/o both}}$ removes all special designs for a fully-connected homogeneous graph.

**Comparison of Different Edge Construction Strategies (RQ3)** In HeterFC, unique edge constructions include het-

| Model | Feverous | Label Accuracy |
|---|---|---|
| HeterFC | **0.3714** | **0.7352** |
| HeterFC$_{\text{mean}}$ | 0.3613 | 0.7238 |
| HeterFC$_{\text{max}}$ | 0.3620 | 0.7226 |
| HeterFC$_{\text{graph}}$ | 0.3640 | 0.7243 |
| HeterFC$_{\text{single}}$ | 0.3641 | 0.7257 |

Table 2: Comparison between HeterFC and its variants: HeterFC$_{\text{mean/max}}$ substitutes attention-based claim-evidence interaction with mean/max pooling. HeterFC$_{\text{graph}}$ uses only graph representation for a purely graph-based model. HeterFC$_{\text{single}}$ ignores the assisted loss.

erogeneous relations, inter-evidence global connections, and intra-evidence local connections. This section ablates these strategies through model variants to assess their impact. For **HeterFC**$_{\text{w/o Het}}$, homogeneous edges replace heterogeneous relations, making all edges identical. **HeterFC**$_{\text{w/o gl}}$ disregards specific local and global connection designs, yielding a fully-connected graph while retaining heterogeneous relations. **HeterFC**$_{\text{w/o both}}$ discards all special designs, leading to a fully-connected and homogeneous graph.

From Figure 3, HeterFC outperforms its variants, with HeterFC$_{\text{w/o both}}$ exhibiting the weakest performance. Furthermore, HeterFC$_{\text{w/o gl}}$ and HeterFC$_{\text{w/o Het}}$ show marked performance drops, emphasizing the effectiveness of both heterogeneous edges and local/global connection designs.

**Investigating Attention Module, Veracity Prediction and Losses (RQ4)** In addition to the discussed graph construction, we further evaluate the Attention-based Claim-evidence Interaction model, Fused Veracity Prediction, and assisted loss designs through several HeterFC variants:

- **HeterFC**$_{\text{mean/max}}$: Substitutes the original Attention-based Claim-evidence Interaction with a mean/max pooling layer for evidence representation aggregation. No assisted loss is computed due to the absence of predicted evidence scores.
- **HeterFC**$_{\text{graph}}$: Excludes $o_t$ in Fused Veracity Prediction, relying solely on graph representation $o_g$ for a purely graph-based model.
- **HeterFC**$_{\text{single}}$: Using only the veracity classification loss $Loss_c$, ignoring the assisted loss $Loss_e$.

Table 2 displays development set results for these variants. Notably, HeterFC surpasses all variants. HeterFC$_{\text{mean/max}}$ perform worse than HeterFC$_{\text{single}}$. These three variants lack assisted loss during training, highlighting the importance of the Attention-based Claim-evidence Interaction model. Comparing HeterFC$_{\text{single}}$ and HeterFC$_{\text{graph}}$, assisted loss and Fused Veracity Prediction contribute similarly to outcomes. The pivotal role of the Attention-based Claim-evidence Interaction module is evident.

## Hyperparameter Sensitivity Analysis (RQ5)

We study the impact of two key hyperparameters: the number of R-GCN layers $k$ determining information aggregation range and the parameter $\beta$ for balancing $Loss_e$ and $Loss_c$.

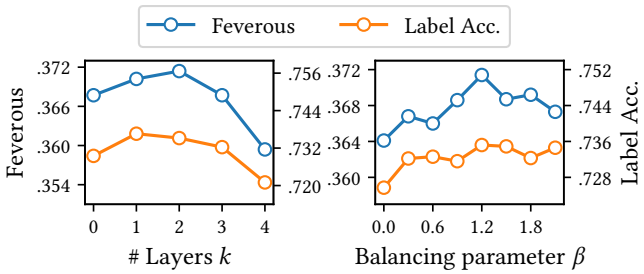


Figure 4: The influence of hyper-parameters on HeterFC's performance on the development set.

| Model | Feverous | Label Acc. |
|---|---|---|
| RoBERTa-Pair$_{\text{mean}}$ | 0.2272 | 0.6689 |
| RoBERTa-Pair$_{\text{max}}$ | 0.2379 | 0.6829 |
| RoBERTa-Concat | 0.2352 | 0.6627 |
| DCUF | 0.2380 | 0.6895 |
| GEAR | 0.1783 | 0.5786 |
| KGAT | 0.2110 | 0.6117 |
| HeterFC | **0.2427** | **0.6977** |

Table 3: Comparison of model performances on the development set under different retrieval method.

**Number of layers $k$.** In the left part of Figure 4, increasing GNN layers enhances metrics, but beyond $k = 2$, metrics drop due to over-smoothing. We select $k = 2$ for HeterFC.

**Balancing parameter $\beta$.** The right part of Figure 4 shows Feverous score peaks at $\beta = 1.2$ before a slight decline, while label accuracy increases steadily with larger $\beta$, stabilizing eventually. We select $\beta = 1.2$ as optimal for HeterFC.

## Evaluating the Robustness of HeterFC (RQ6)

We extend our analysis beyond the retrieval method by (Hu et al. 2022) to include a simpler approach by (Aly et al. 2021). Despite less effective retrieval, it simulates scenarios with limited, noisy evidence. Results in Table 3 align with those in Table 1, showing that HeterFC maintains superior performance over baseline models, underscoring its robustness even with less optimal retrieval techniques.

## Conclusion

In this paper, we introduce HeterFC, a novel word-level heterogeneous-graph-based model for fact checking that effectively combines unstructured and structured information. Our model employs a carefully designed graph structure with word-level nodes and diverse edge types. We integrate a heterogeneous information propagation module with attention-based claim-evidence interaction to capture the semantic relationships between claims and evidence. Additionally, we introduce an assisted loss based on attention scores to differentiate valid and invalid evidence. Extensive experiments confirm the superiority of HeterFC over diverse baseline models.

## Acknowledgements

This work is jointly sponsored by National Natural Science Foundation of China (U19B2038, 62372454, 62206291, 62236010).

## References

Aly, R.; Guo, Z.; Schlichtkrull, M. S.; Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; Cocarascu, O.; and Mittal, A. 2021. FEVEROUS: Fact Extraction and VERification Over Unstructured and Structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Bouziane, M.; Perrin, H.; Sadeq, A.; Nguyen, T.; Cluzeau, A.; and Mardas, J. 2021. FaBULOUS: Fact-checking Based on Understanding of Language Over Unstructured and Structured information. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, 31–39. Association for Computational Linguistics.

Chen, D.; Fisch, A.; Weston, J.; and Bordes, A. 2017. Reading Wikipedia to Answer Open-Domain Questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1870–1879. Association for Computational Linguistics.

Chen, Z.; Hui, S. C.; Zhuang, F.; Liao, L.; Li, F.; Jia, M.; and Li, J. 2022. EvidenceNet: Evidence Fusion Network for Fact Verification. In *Proceedings of the ACM Web Conference 2022*, 2636–2645.

Fan, S.; Zhu, J.; Han, X.; Shi, C.; Hu, L.; Ma, B.; and Li, Y. 2019. Metapath-guided Heterogeneous Graph Neural Network for Intent Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 2478–2486. ACM.

Funkquist, M. 2021. Combining sentence and table evidence to predict veracity of factual claims using TaPaS and RoBERTa. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, 92–100. Association for Computational Linguistics.

Gi, I.-Z.; Fang, T.-Y.; and Tsai, R. T.-H. 2021. Verdict Inference with Claim and Retrieved Elements Using RoBERTa. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, 60–65. Association for Computational Linguistics.

Hu, N.; Wu, Z.; Lai, Y.; Liu, X.; and Feng, Y. 2022. Dual-Channel Evidence Fusion for Fact Verification over Texts and Tables. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5232–5242. Association for Computational Linguistics.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kotonya, N.; Spooner, T.; Magazzeni, D.; and Toni, F. 2021. Graph Reasoning with Context-Aware Linearization for Interpretable Fact Extraction and Verification. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, 21–30. Association for Computational Linguistics.

Krause, N. M.; Freiling, I.; Beets, B.; and Brossard, D. 2020. Fact-checking as risk communication: the multi-layered risk of misinformation in times of COVID-19. *Journal of Risk Research*, 23: 1052 – 1059.

Liu, Q.; Wu, J.; Wu, S.; and Wang, L. 2023. Out-of-distribution Evidence-aware Fake News Detection via Dual Adversarial Debiasing. *arXiv preprint arXiv:2304.12888*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692.

Liu, Z.; Xiong, C.; Sun, M.; and Liu, Z. 2020. Fine-grained Fact Verification with Kernel Graph Attention Network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7342–7351. Association for Computational Linguistics.

Malon, C. 2021. Team Papelo at FEVEROUS: Multi-hop Evidence Pursuit. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, 40–49. Association for Computational Linguistics.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 3111–3119.

Naeem, S. B.; and Bhatti, R. 2020. The Covid-19 'infodemic': a new front for information professionals. *Health Information and Libraries Journal*, 37: 233 – 239.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Berg, R. v. d.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, 593–607. Springer.

Soleimani, A.; Monz, C.; and Worring, M. 2020. Bert for evidence retrieval and claim verification. In *European Conference on Information Retrieval*, 359–366.

Sun, H.; Dhingra, B.; Zaheer, M.; Mazaitis, K.; Salakhutdinov, R.; and Cohen, W. W. 2018. Open domain question answering using early fusion of knowledge bases and text. *arXiv preprint arXiv:1809.00782*.

Thorne, J.; Vlachos, A.; Christodoulopoulos, C.; and Mittal, A. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 809–819. Association for Computational Linguistics.

Wadden, D.; Lin, S.; Lo, K.; Wang, L. L.; van Zuylen, M.; Cohan, A.; and Hajishirzi, H. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7534–7550. Association for Computational Linguistics.

Wang, W. Y. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 422–426. Association for Computational Linguistics.

Wright, D.; Wadden, D.; Lo, K.; Kuehl, B.; Cohan, A.; Augenstein, I.; and Wang, L. 2022. Generating Scientific Claims for Zero-Shot Scientific Fact Checking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2448–2460. Association for Computational Linguistics.

Wu, J.; Xu, W.; Liu, Q.; Wu, S.; and Wang, L. 2022. Adversarial contrastive learning for evidence-aware fake news detection with graph neural networks. *arXiv preprint arXiv:2210.05498*.

Xu, W.; Liu, Q.; Wu, S.; and Wang, L. 2023. Counterfactual Debiasing for Fact Verification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6777–6789.

Xu, W.; Wu, J.; Liu, Q.; Wu, S.; and Wang, L. 2022. Evidence-aware Fake News Detection with Graph Neural Networks. *Proceedings of the ACM Web Conference 2022*.

Yan, Q.; Zhang, Y.; Liu, Q.; Wu, S.; and Wang, L. 2021. Relation-aware Heterogeneous Graph for User Profiling. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 3573–3577.

Ying, Z.; You, J.; Morris, C.; Ren, X.; Hamilton, W. L.; and Leskovec, J. 2018. Hierarchical Graph Representation Learning with Differentiable Pooling. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 4805–4815.

Yu, W.; Zhou, J.; Yu, W.; Liang, X.; and Xiao, N. 2019. Heterogeneous Graph Learning for Visual Commonsense Reasoning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2765–2775.

Zellers, R.; Holtzman, A.; Rashkin, H.; Bisk, Y.; Farhadi, A.; Roesner, F.; and Choi, Y. 2019. Defending Against Neural Fake News. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 9051–9062.

Zhao, C.; Xiong, C.; Rosset, C.; Song, X.; Bennett, P. N.; and Tiwary, S. 2020. Transformer-XH: Multi-Evidence Reasoning with eXtra Hop Attention. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhao, H.; Yao, Q.; Li, J.; Song, Y.; and Lee, D. L. 2017. Meta-Graph Based Recommendation Fusion over Heterogeneous Information Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*, 635–644. ACM.

Zhong, W.; Xu, J.; Tang, D.; Xu, Z.; Duan, N.; Zhou, M.; Wang, J.; and Yin, J. 2020. Reasoning Over Semantic-Level Graph for Fact Checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6170–6180. Association for Computational Linguistics.

Zhou, J.; Han, X.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2019. GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 892–901. Association for Computational Linguistics.