CORRECT: Context- and Reference-Augmented Reasoning and Prompting for Fact-Checking

Delvin Ce Zhang
The Pennsylvania State University
delvincezhang@gmail.com

Dongwon Lee
The Pennsylvania State University
dongwon@psu.edu

Abstract

Fact-checking the truthfulness of claims usually requires reasoning over multiple evidence sentences. Oftentimes, evidence sentences may not be always self-contained, and may require additional contexts and references from elsewhere to understand coreferential expressions, acronyms, and the scope of a reported finding. For example, evidence sentences from an academic paper may need contextual sentences in the paper and descriptions in its cited papers to determine the scope of a research discovery. However, most fact-checking models mainly focus on the reasoning within evidence sentences, and ignore the auxiliary contexts and references. To address this problem, we propose a novel method, Context- and Reference-augmented Reasoning and Prompting. For evidence reasoning, we construct a three-layer evidence graph with evidence, context, and reference layers. We design intra- and cross-layer reasoning to integrate three graph layers into a unified evidence embedding. For verdict prediction, we design evidence-conditioned prompt encoder, which produces unique prompt embeddings for each claim. These evidence-conditioned prompt embeddings and claims are unified for fact-checking. Experiments verify the strength of our model. Code and datasets are available at https://github.com/cezhang01/correct.

1 Introduction

The proliferation of misinformation has posed growing challenge in the realm of information reliability. There is a need to develop automated fact-checking methods (Guo et al., 2022) to verify the truthfulness of real-world claims using evidence.

Existing fact-checking models (Zhou et al., 2019; Liu et al., 2020) have shown promise in aggregating and reasoning over multiple evidence sentences to verify a claim. However, the evidence sentences retrieved from a large corpus may contain incomplete information when they are taken out-of-corpus. We need to refer to additional contexts and references from elsewhere to understand coreferential expressions, acronyms, and the scope of a reported finding. For example, Fig. 1(a) illustrates context-dependent evidence, where undefined acronym "MNC" in evidence sentence from a paper abstract requires additional context from the abstract to jointly interpret the meaning of acronym "MNC". Fig. 1(b) presents reference-dependent evidence, where we need to check the cited paper to understand that "SARS-CoV-2 infection" and "COVID-19 infection" are coreferential expressions, so that we could accurately fact-check the claim. Such scenario also exists in general domain where evidence sentences from a Wikipedia page may need contextual sentences in the same page and text in the hyperlinked pages to complement the insufficient information in the evidence.

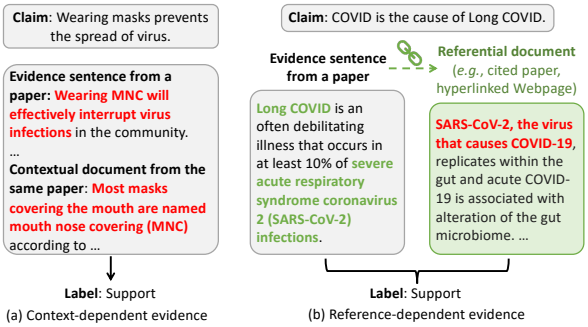


Figure 1: Illustration of (a) context-dependent and (b) reference-dependent evidence from BearFact dataset.

Challenges and Approach. To overcome the limitations of existing methods, we propose **Co**ntext- and **R**eference-augmented **Re**asoning and prompting for fa**ct**-checking (CORRECT), to address two open questions.

First, *how to aggregate both contextual and referential documents into evidence reasoning?* Some models are proposed to capture contextual documents, e.g., MultiVerS (Wadden et al., 2022). Some others are designed for referential documents, e.g.,

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 3007–3019
April 29 - May 4, 2025

Transformer-XH (Zhao et al., 2020) and HESM (Subramanian and Lee, 2020). However, they incorporate either contextual or referential documents, failing to aggregate both of them into unified evidence embedding. Moreover, most of them simply concatenate evidence with contextual or referential documents, and inefficiently input the long text to language models for evidence encoding. Though they have shown that modeling either contexts or references helps fact-checking, integrating both of them for evidence reasoning is still unexplored. In our model, we construct a three-layer graph with evidence, context, and reference layers. We design intra- and cross-layer reasoning to aggregate three graph layers into unified evidence embedding.

Second, *how to integrate evidence reasoning and claim for accurate verdict prediction?* Previous fact-checking methods, e.g., ProToCo (Zeng and Gao, 2023), rely on natural language as input prompt to language model for claim verification. However, discrete natural language prompts are difficult to design and may result in suboptimal results (Zhou et al., 2022b). Recently, prompt tuning (Lester et al., 2021) uses continuous and learnable prompt embeddings to replace discrete prompt and has achieved decent result, but no one has explored its design for claim verification. We propose evidence-conditioned prompt encoder, which takes evidence embedding as input, and produces unique prompt embeddings for each claim. We combine prompt embeddings with claim token embeddings to unify evidence and claim for verdict prediction.

Contributions. First, we propose a novel model, Context- and Reference-augmented Reasoning and Prompting (CORRECT), to integrate both contextual and referential documents into evidence reasoning. Second, we design a three-layer evidence graph, and propose intra- and cross-layer reasoning to learn unified evidence embedding. Third, we propose evidence-conditioned prompt embeddings, which are combined with claims to integrate evidence reasoning with claim for fact-checking.

2 Related Work

Multi-hop fact-checking. Complex claims usually require reasoning over multiple evidence sentences. Many methods are based on Language Models (Vaswani et al., 2017; Devlin et al., 2019) and Graph Neural Networks (Hamilton et al., 2017), such as GEAR (Zhou et al., 2019), KGAT (Liu et al., 2020), DREAM (Zhong et al., 2020), SaGP (Si et al., 2023), DECKER (Zou et al., 2023), CausalWalk (Zhang et al., 2024a), etc. However, they mainly focus on the reasoning within evidence sentences. They ignore the auxiliary contextual and referential documents. Methods incorporating contextual documents are proposed, e.g., ParagraphJoint (Li and Peng, 2021), ARSJoint (Zhang et al., 2021), MultiVerS (Wadden et al., 2022), etc. Some others integrating referential documents include Transformer-XH (Zhao et al., 2020) and HESM (Subramanian and Lee, 2020). However, they incorporate either contextual or referential documents, but not both. In contrast, we construct a three-layer evidence graph to model evidence sentences, contexts, and references. There are fake news detection models where auxiliary graph with Wikidata is used (Hu et al., 2021; Whitehouse et al., 2022). Fake news detection aims to detect the whole article with meta-data, while fact-checking focuses on claim sentences with retrieved evidence.

Some fact-checking works are based on retrieval-augmented generation (Zeng and Gao, 2024). They unify evidence retrieval and claim verification as a joint approach, while our model mainly focuses on verification, and relies on external tool for evidence retrieval. Our setting is consistent with existing works (Wadden et al., 2022; Zhang et al., 2024a).

Prompt-based fact-checking. Some models verify claims by prompting LLMs (Achiam et al., 2023). ProToCo (Zeng and Gao, 2023) inputs both evidence sentences and claim to T5 (Raffel et al., 2020). ProgramFC (Pan et al., 2023) decomposes complex claims into simpler sub-tasks and uses natural language to prompt LLMs. Varifocal (Ousidhoum et al., 2022) formulates fact-checking as question generation and answering. They rely on handcrafted natural language as prompt. The performance heavily relies on the choice of prompt, and it is difficult to design a prompt that produces a decent result, as shown in (Zhou et al., 2022b). Our model is designed with learnable prompt embeddings where the prompting instruction is naturally learned by embeddings through optimization.

Prompt learning. Prompting (Brown et al., 2020) uses natural language as the input to language models to fulfill certain tasks. Many prompting models have been proposed, including natural language prompt (Gao et al., 2021; Shin et al., 2020) and prompt embeddings (Lester et al., 2021; Liu et al., 2023, 2022; Li and Liang, 2021). Prompting also benefits many tasks (Zhou et al., 2022a; Tan et al., 2022). However, no one has explored

Table 1: Summary of mathematical notations.

Notation	Description
$\mathcal{D}$	a fact-checking dataset
$\mathcal{X}$	a set of $N = \|\mathcal{X}\|$ claims
$\mathcal{E}$	a corpus of evidence sentences
$\mathcal{C}$	a set of contextual documents
$\mathcal{R}$	a set of referential documents
$\mathcal{N}_{\text{ref}}(e)$	evidence sentence e's referential documents
$\mathcal{N}_{\text{evid}}(x)$	claim x's retrieved evidence sentences
$\mathcal{Y}$	a set of labels
$\mathbf{h}^{(l)}_{e,\text{CLS}}$	evidence sentence e's [CLS] token embedding
$\hat{\mathbf{h}}^{(l)}_{c}$	aggregated contextual document embedding
$\hat{\mathbf{h}}^{(l)}_{r}$	aggregated referential document embedding
$\widehat{\mathbf{H}}^{(l)}_{e}$	evidence sentence e's augmented embedding matrix
$\boldsymbol{\pi}_{m,y}$	the m-th prompt embedding for class y
$\mathbf{h}_{m,y}$	the m-th base prompt embedding for class y

prompt embeddings for fact-checking.

Text-attributed graph. Texts are usually connected in a graph structure, termed text-attributed graph (Zhang et al., 2024b). Various methods have been developed to learn text embeddings in an unsupervised manner (Zhang and Lauw, 2020, 2023, 2021; Zhang et al., 2023; Yang et al., 2021; Jin et al., 2023; Yang et al., 2024). Though both our model and these works construct a text-attributed graph, our work is different from them, since our model is a supervised model for fact-checking.

3 Model Architecture

We introduce **Co**ntext- and **R**eference-augmented **Re**asoning and prompting for fa**ct**-checking (CORRECT). Table 1 summarizes math notations.

3.1 Problem Formulation

We are given a fact-checking dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{E}, \mathcal{C}, \mathcal{R}\}$. Claim set $\mathcal{X} = \{x_i\}_{i=1}^{N}$ contains a set of N claims. Evidence set $\mathcal{E} = \{e_j\}_{j=1}^{E}$ is a corpus of E evidence sentences. For each evidence sentence $e \in \mathcal{E}$, we have its contextual document $c \in \mathcal{C}$. Usually, an evidence sentence has only one contextual document, from which this sentence is retrieved. We also have e's referential documents $\mathcal{N}_{\text{ref}}(e) = \{r_{e,n}\}_{n=1}^{R_e} \subset \mathcal{R}$. Here R_e is the number of e's referential documents. Evidence sentence e may have multiple referential documents, such as papers cited by e's paper or Webpages hyperlinked by e. We use $\mathcal{N}_{\text{ref}}(e)$ to represent the set of e's referential documents. We use $\mathcal{N}_{\text{evid}}(x) \subset \mathcal{E}$ to denote the set of evidence sentences for a claim x.

Given $\mathcal{D}$ as input, we design a model that uses evidence sentences from $\mathcal{E}$ together with their contextual documents in $\mathcal{C}$ and referential documents in $\mathcal{R}$ to verify claims. Eventually, for each claim $x \in \mathcal{X}$, we output its predicted label $\hat{y} \in \mathcal{Y} = \{\text{SUPPORT, REFUTE, NEI}\}$, indicating whether the evidence supports, refutes, or does not have enough information to verify the claim.

As shown in Fig. 2, CORRECT has two modules: (a-c) context- and reference-augmented evidence reasoning on three-layer graph, (d) evidence-conditioned prompting for claim verification.

3.2 Three-layer Evidence Graph Reasoning

Graph construction. For each claim $x \in \mathcal{X}$ and its evidence sentences $\mathcal{N}_{\text{evid}}(x) \subset \mathcal{E}$, we construct a three-layer graph with evidence, context, and reference layers in Fig. 2(a). We consider evidence sentences, contextual documents, and referential documents as three types of vertices. Each type of vertices reside on their own layer. Cross-layer links between evidence layer and context layer connect each evidence sentence with its contextual document. Each evidence sentence and its referential documents are connected by cross-layer referential links. Green links in Fig. 2(a) are cross-layer links. For multi-evidence reasoning, we add intra-layer links on evidence layer where evidence sentences of a claim are fully connected, shown by black links in Fig. 2(a). The purpose of constructing three layers instead of mixing all vertices into one layer is to better differentiate three types of vertices.

Intra-layer reasoning. Evidence reasoning includes intra- and cross-layer reasoning. We first show intra reasoning (orange arrows in Fig. 2(b)).

For each evidence sentence $e \in \mathcal{N}_{\text{evid}}(x)$, we let $\mathbf{H}^{(l)}_{e} = [\mathbf{h}^{(l)}_{e,\text{CLS}}, \mathbf{h}^{(l)}_{e,1}, \mathbf{h}^{(l)}_{e,2}, ...]$ denote the output from the l-th Transformer step. Note that previous works call it the l-th layer, but to distinguish it from our three-layer graph, we instead call it the l-th step. $\mathbf{h}^{(l)}_{e,i} \in \mathbb{R}^d$ is d-dimensional token embedding. We use graph neural network to aggregate different evidence sentences of a claim. For each evidence sentence e, we first project it by

$$\tilde{\mathbf{h}}^{(l)}_{e,\text{CLS}} = \mathbf{W}_1 \mathbf{h}^{(l)}_{e,\text{CLS}}. \tag{1}$$

The [CLS] token is taken as the evidence sentence embedding, and $\mathbf{W}_1 \in \mathbb{R}^{d \times d}$ is type-specific parameter. We design type-specific attention.

$$a_{e,e'} = \text{softmax}\Big(\text{LeakyReLU}(\mathbf{b}_1^\top [\tilde{\mathbf{h}}^{(l)}_{e,\text{CLS}} || \tilde{\mathbf{h}}^{(l)}_{e',\text{CLS}}])\Big). \tag{2}$$

$e' \in \mathcal{N}_{\text{evid}}(x) \backslash e$ is another evidence sentence for the same claim x, $[\cdot||\cdot]$ is concatenation, and $\mathbf{b}_1 \in$

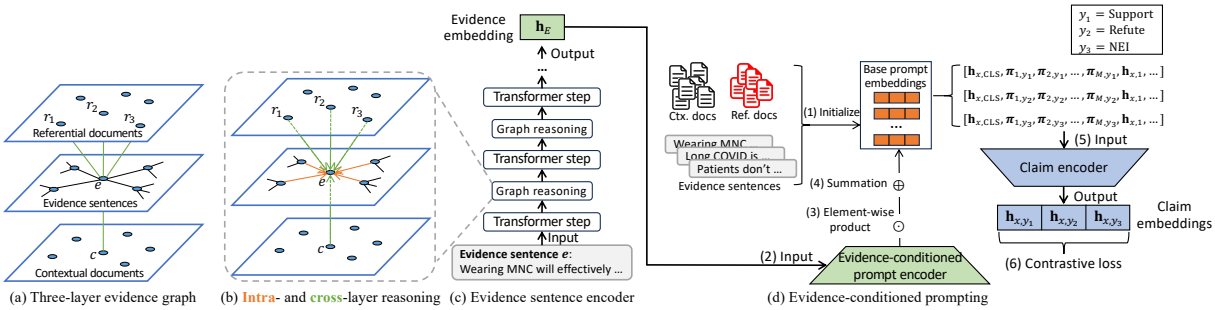


Figure 2: Model architecture. (a) A three-layer graph for a claim. (b) Intra- and cross-layer reasoning. (c) A nested architecture with language model and graph reasoning for evidence encoding. (d) Evidence-conditioned prompting.

$\mathbb{R}^{2d}$ is learnable parameter. Finally, we aggregate evidence sentences to e by mean pooling.

$$\hat{\mathbf{h}}_e^{(l)} = \text{mean}\Big(\tilde{\mathbf{h}}_{e,\text{CLS}}^{(l)}, \sum_{e' \in \mathcal{N}_{\text{evid}}(x) \backslash e} a_{e,e'} \tilde{\mathbf{h}}_{e',\text{CLS}}^{(l)}\Big). \tag{3}$$

The aggregated sentence embedding $\hat{\mathbf{h}}_e^{(l)}$ captures information of both itself and other evidence sentences. To summarize Eqs. 1–3, we have

$$\hat{\mathbf{h}}_e^{(l)} = f_{\text{GNN}}\Big(\mathbf{h}_{e,\text{CLS}}^{(l)}, \{\mathbf{h}_{e',\text{CLS}}^{(l)} | e' \in \mathcal{N}_{\text{evid}}(x) \backslash e\}; \mathbf{W}_1, \mathbf{b}_1\Big). \tag{4}$$

To integrate intra-layer aggregation into the encoding of each evidence sentence, we introduce a *virtual token* to represent the aggregated sentence embedding $\hat{\mathbf{h}}_e^{(l)}$. For evidence sentence e, we concatenate $\hat{\mathbf{h}}_e^{(l)}$ with e's text token embeddings by $\widehat{\mathbf{H}}_e^{(l)} = \hat{\mathbf{h}}_e^{(l)} || \mathbf{H}_e^{(l)}$. After concatenation, $\widehat{\mathbf{H}}_e^{(l)}$ contains information of both evidence sentence e's text and the aggregated embedding from e's intra-layer neighbors. We aim to propagate the aggregated sentence embedding to other text tokens of sentence e, so that the text tokens can fully unify other sentences for multi-evidence reasoning. We will introduce *asymmetric* multi-head self-attention to achieve this goal. But before that, we first discuss cross-layer reasoning.

Cross-layer reasoning. We present cross-layer reasoning, which aggregates contextual and referential documents into evidence sentences (green arrows in Fig. 2(b)). The aggregation from referential documents to evidence sentence is similarly defined by Eq. 5. We use reference-specific parameters, $\mathbf{W}_2$ and $\mathbf{b}_2$, to preserve graph heterogeneity.

$$\hat{\mathbf{h}}_r^{(l)} = f_{\text{GNN}}\Big(\mathbf{h}_{e,\text{CLS}}^{(l)}, \{\mathbf{h}_{r,\text{CLS}}^{(l)} | r \in \mathcal{N}_{\text{ref}}(e)\}; \mathbf{W}_2, \mathbf{b}_2\Big). \tag{5}$$

Each referential document $r \in \mathcal{N}_{\text{ref}}(e)$ is also encoded, and its [CLS] token is passed to Eq. 5 for aggregation. Similarly, we have $\hat{\mathbf{h}}_c^{(l)}$ as contextual document embedding. To integrate both embeddings into evidence sentence for cross-layer reasoning, we introduce two more *virtual tokens*.

$$\widehat{\mathbf{H}}_e^{(l)} = \hat{\mathbf{h}}_c^{(l)} || \hat{\mathbf{h}}_r^{(l)} || \hat{\mathbf{h}}_e^{(l)} || \mathbf{H}_e^{(l)}. \tag{6}$$

The augmented embedding matrix, i.e., $\widehat{\mathbf{H}}_e^{(l)}$, contains both intra-evidence reasoning as well as cross-layer context and reference augmentation.

To fully unify all three graph layers into evidence sentence e, we input $\widehat{\mathbf{H}}_e^{(l)}$ at Eq. 6 to the $(l+1)$-th Transformer step with our proposed *asymmetric* multi-head self-attention (MSA^{asy}).

$$\text{MSA}^{\text{asy}}(\mathbf{H}_e^{(l)}, \widehat{\mathbf{H}}_e^{(l)}, \widehat{\mathbf{H}}_e^{(l)}) = \text{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\Big)\mathbf{V},$$
$$\mathbf{Q} = \mathbf{H}_e^{(l)} \mathbf{W}_Q^{(l)}, \quad \mathbf{K} = \widehat{\mathbf{H}}_e^{(l)} \mathbf{W}_K^{(l)}, \quad \mathbf{V} = \widehat{\mathbf{H}}_e^{(l)} \mathbf{W}_V^{(l)}. \tag{7}$$

Keys $\mathbf{K}$ and values $\mathbf{V}$ are augmented with virtual tokens, but queries $\mathbf{Q}$ are not, to avoid context and reference embeddings being overwritten by evidence sentence embedding. The result of asymmetric MSA is passed to a multi-layer perceptron and layer normalization (Vaswani et al., 2017). Finally, we obtain the output from the $(l+1)$-th step, $\mathbf{H}_e^{(l+1)}$, integrating evidence sentence e, other evidence sentences of the same claim, e's contextual and referential documents, see Fig. 2(c).

We conduct such intra-layer and cross-layer reasoning inside each Transformer step to allow different graph layers to fully communicate with each other. We repeat such nested and graph-augmented encoding for L times, and obtain $\mathbf{h}_e = \mathbf{h}_{e,\text{CLS}}^{(L)}$ as the graph-augmented embedding for evidence sentence e. This nested architecture is shown by Fig. 2(c). For claim x, we have $\{\mathbf{h}_e\}_{e \in \mathcal{N}_{\text{evid}}(x)}$, a set of graph-augmented embeddings for its evidence

sentences. Finally, we aggregate them by mean pooling and obtain a single evidence embedding.

$$\mathbf{h}_E = \text{mean}(\mathbf{h}_e | e \in \mathcal{N}_{\text{evid}}(x)). \quad (8)$$

3.3 Evidence-conditioned Prompting

Now we integrate evidence reasoning into claim embedding to fully integrate their information for fact-checking. Prompting (Liu et al., 2023) is a powerful method in fact-checking (Zeng and Gao, 2023). However, existing models are mainly based on natural language as input prompt to language models for verdict prediction. Handcrafted discrete prompt has two disadvantages: First, it is difficult to manually design a prompt that provides a decent performance. Previous works (Zhou et al., 2022b) have shown that the change of a single word in the prompt may lead to significant deterioration of the results, and it is time-consuming to enumerate every prompt. Second, discrete natural language prompt is difficult to optimize, since language models are intrinsically continuous.

To mitigate these problems, we explore learnable and continuous prompt embedding. Below we design a prompt encoder, which takes evidence embedding $\mathbf{h}_E$ as input, and produces evidence-conditioned prompt embeddings. See Fig. 2(d).

Evidence-conditioned prompt encoder. We consider below continuous embeddings as prompt.

$$\mathbf{P}_x = [\mathbf{h}_{x,\text{CLS}}, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, ..., \boldsymbol{\pi}_M, \mathbf{h}_{x,1}, \mathbf{h}_{x,2}, ...]. \quad (9)$$

Here $\{\boldsymbol{\pi}_m\}_{m=1}^M$ where $\boldsymbol{\pi}_m \in \mathbb{R}^d$ is a set of M learnable evidence-conditioned prompt embeddings to be explained shortly, and M is a hyperparameter, indicating the number of prompt embeddings. Each $\mathbf{h}_{x,i} \in \mathbb{R}^d$ is a d-dimensional embedding of the i-th text token in claim x. In language models, there is an embedding look-up table before language model encoder. In this look-up table, input text tokens are first mapped to the vocabulary to obtain their token embeddings, which are then summed up with positional encodings. $\mathbf{h}_{x,i}$ in Eq. 9 is obtained by this look-up table.

Now we explain prompt embeddings $\{\boldsymbol{\pi}_m\}_{m=1}^M$, output from an evidence-conditioned prompt encoder. We first initialize M *base* prompt embeddings, $\{\mathbf{h}_m\}_{m=1}^M$. We then project evidence embedding $\mathbf{h}_E$ in Eq. 8 to the prompt embedding space, followed by element-wise product and summation.

$$\boldsymbol{\alpha}_x = \tanh\Big(\frac{\mathbf{W}_\alpha \mathbf{h}_E + \mathbf{b}_\alpha}{\tau}\Big), \quad \boldsymbol{\beta}_x = \tanh\Big(\frac{\mathbf{W}_\beta \mathbf{h}_E + \mathbf{b}_\beta}{\tau}\Big), \quad (10)$$

$$\boldsymbol{\pi}_m = \mathbf{h}_m \odot (\boldsymbol{\alpha}_x + \mathbf{1}) + \boldsymbol{\beta}_x. \quad (11)$$

$\odot$ is element-wise product, and $\mathbf{1} \in \mathbb{R}^d$ is a vector of ones to ensure that the scaling of $\mathbf{h}_m$ is centered around one. τ is a temperature to scale the shape of tanh function. $\boldsymbol{\pi}_m$ is thus conditioned on evidence embedding, and different claims with their own evidence sentences should have their unique claim-specific prompt embeddings, shown by Fig. 2(d).

Given the label set $\mathcal{Y}$ ={SUPPORT, REFUTE, NEI}, which usually has three types of labels, we apply above evidence-conditioned prompt encoder and correspondingly obtain three sets of prompt embeddings, $\{\boldsymbol{\pi}_{m,y}\}_{m=1}^M$ where $y \in \mathcal{Y}$. As in Eq. 9, we concatenate each set of prompt embeddings with token embeddings of claim x, and obtain three sets of inputs $\{\mathbf{P}_{x,y}\}_{y\in\mathcal{Y}}$ to claim encoder.

$$\begin{aligned}\mathbf{H}_{x,y}^{(L)} &= f(\mathbf{P}_{x,y}) \\ &= f([\mathbf{h}_{x,\text{CLS}}, \boldsymbol{\pi}_{1,y}, \boldsymbol{\pi}_{2,y}, ..., \boldsymbol{\pi}_{M,y}, \mathbf{h}_{x,1}, \mathbf{h}_{x,2}, ...])\end{aligned} \quad (12)$$

$\mathbf{H}_{x,y}^{(L)}$ is the output from the claim encoder, and its [CLS] token is taken as claim embedding $\mathbf{h}_{x,y} = \mathbf{h}_{x,y,\text{CLS}}^{(L)}$. Claim encoder shares parameters with evidence encoder. Due to contextualized modeling, claim token embeddings and evidence-conditioned prompt embeddings fully exchange information, and the output claim embedding captures both claim x and evidence reasoning for fact-checking.

Finally, we use contrastive loss function to predict the veracity of claim x by

$$\mathcal{L} = -\sum_{x\in\mathcal{X}_{\text{train}}} \log \frac{\exp(\mathbf{h}_{x,y}^\top \mathbf{h}_E)}{\exp(\mathbf{h}_{x,y}^\top \mathbf{h}_E) + \sum_{y'\in\mathcal{Y}\setminus y} \exp(\mathbf{h}_{x,y'}^\top \mathbf{h}_E)}. \quad (13)$$

$\mathbf{h}_E$ is evidence embedding of claim x obtained by Eq. 8. $\mathcal{X}_{\text{train}}$ is a set of training claims. Though we use three types of labels in $\mathcal{Y}$, more types of labels in $\mathcal{Y}$ can also be modeled. Algorithm 1 summarizes the learning process.

Initialization of base prompt embeddings. Previous works (Zhou et al., 2022b) have shown the importance of the initialization of *base* prompt embeddings $\{\mathbf{h}_{m,y}\}_{m=1}^M$ where $y \in \mathcal{Y}$. Some of them randomly initialize the embeddings, while others use word embeddings of discrete prompts. Random initialization presents unstable optimization (Wen and Fang, 2023), while it is difficult to choose the right discrete prompts for initialization. We solve these problems by using the three-layer graph.

For a claim x, the vertices on its three-layer graph consistently carry the signal of claim x's veracity due to semantic relatedness. Thus, for

Table 2: Dataset statistics.

Name	#Claims		#Contextual	#Referential
	Train	Test	Documents	Documents
FEVEROUS-S	23,912	5,978	19,546	21,579
BearFact	1,158	290	1,166	12,938
Check-COVID	1,275	229	347	3,132
SciFact	809	300	1,189	9,617

each label in the label set $y \in \mathcal{Y}$, we have training claims belonging to this label $\mathcal{X}_{\text{train},y} = \{x \in \mathcal{X}_{\text{train}} | y_x = y\}$. For each of these claims, we truncate its evidence sentences, contextual and referential documents to M words, and obtain their M word embeddings in the look-up table of language model. We then take mean pooling for evidence sentences, contextual and referential documents, and obtain M pooled word embeddings for each claim. Finally, we average all training claims belonging to the same label $\mathcal{X}_{\text{train},y}$, and obtain M word embeddings, which are used to initialize M base prompt embeddings $\{\mathbf{h}_{m,y}\}_{m=1}^{M}$. They are derived from training claims of the same label, thus provide a more informative starting point than random initialization for verdict prediction. We repeat this process for every label $y \in \mathcal{Y}$, and obtain initialization for each set of base prompt embeddings.

4 Experiments

We conduct extensive experiments and ablation analysis to evaluate the effectiveness of the proposed model CORRECT.

Datasets. We use 4 datasets in Table 2. FEVEROUS (Aly et al., 2021) is a general-domain dataset. Each claim is annotated in the form of sentences and/or cells from tables in Wikipedia pages. Since we focus on textual fact-checking, we follow (Pan et al., 2023) and select claims that only require sentences as evidence. We call this subset **FEVEROUS-S**. **BearFact** (Wuehrl et al., 2024) is a biomedical dataset with sentences from papers as evidence. Its original dataset does not have evidence for claims in NEI class. We follow (Zeng and Gao, 2023) and select sentences that have the highest *tf-idf* similarity with those claims as evidence. **Check-COVID** (Wang et al., 2023) contains claims about COVID-19. **SciFact** (Wadden et al., 2020) is a dataset with sentences in papers as evidence. As in its original paper, for claims in NEI class, we choose sentences from the cited abstract with top-3 highest *tf-idf* similarity with the claim as evidence. Appendix B contains data preprocessing details.

Baselines. We have 4 categories of baselines.

***i*) Multi-hop fact-checking**, KGAT (Liu et al., 2020), HESM (Subramanian and Lee, 2020), Transformer-XH (Zhao et al., 2020), MultiVerS (Wadden et al., 2022), and the recent CausalWalk (Zhang et al., 2024a). MultiVerS models contextual documents, and HESM and Transformer-XH incorporate referential documents. By comparing to them, we highlight the advantage of three-layer graph for modeling both contextual and referential documents. Since our model is built on Transformer-XH, we further extend it by modeling both contextual and referential documents, and name it Transformer-XH++. The comparison showcases the effect of evidence-conditioned prompting.

***ii*) Few-shot fact-checking**, GPT2-PPL (Lee et al., 2021), ProToCo (Zeng and Gao, 2023), and ProgramFC (Pan et al., 2023). They are mainly designed for few-shot setting. By increasing their training set, we could also compare to them on fully supervised setting. ProToCo and ProgramFC are proposed with handcrafted natural language prompt. By comparison, we verify the usefulness of our evidence-conditioned prompt embedding.

***iii*) Prompt tuning** is not for fact-checking. But for completeness, we convert P-Tuning v2 (Liu et al., 2022), a continuous prompting, to our task.

***iv*) Retrieval-augmented generation for fact-checking**. Though our model is not designed with retrieval-augmented generation, we still compare to JustiLM (Zeng and Gao, 2024) for completeness.

Implementation details. Following (Vaswani et al., 2017), we set L to 12 and d to 768. Number of prompt embeddings M is 8. Temperature τ in Eq. 10 is 100. For both our model and language model-based baselines, we initialize the model with pre-trained parameters in biomedical domain (Gu et al., 2021) for scientific datasets, and in general domain (Devlin et al., 2019) for FEVEROUS-S. Each result is obtained by 5 independent runs. Experiments are done on 4 NVIDIA A100 80GB GPUs. More details are in Appendix C.

We present two experimental settings below.

Fully supervised v.s. Few-shot. For fully supervised setting, we train the model on the training set. If the dataset provides data split, we follow the split and obtain training and test sets. Otherwise, we split the dataset into 80:20 for training and test, respectively. Among training set, we further reserve 10% for validation. For few-shot setting, we report 5-shot experiments as the main results, i.e., for each class in the label set $y \in \mathcal{Y}$, we randomly sample

Table 3: Verdict prediction results on *fully supervised* setting with *Macro F1* score. Results are in percentage.

Model	BearFact		Check-COVID		SciFact		FEVEROUS-S	
	Gold	Retrieved	Gold	Retrieved	Gold	Retrieved	Gold	Retrieved
KGAT	53.11±2.25	36.55±1.95	71.97±1.31	75.83±0.74	70.23±1.08	59.83±0.68	86.10±0.32	67.76±0.93
HESM	44.90±2.20	42.93±0.27	62.85±0.59	71.58±1.98	68.66±0.69	50.91±2.57	83.12±0.80	67.43±0.81
Transformer-XH	45.28±1.08	38.39±0.80	67.81±0.93	76.51±2.09	72.01±0.86	56.26±0.64	85.44±0.75	68.13±0.52
Transformer-XH++	46.81±1.52	41.06±1.70	70.52±0.55	78.49±0.52	73.92±0.58	57.82±2.29	85.35±0.45	69.76±0.73
MultiVerS	51.56±1.30	38.71±1.96	66.32±1.27	70.01±2.23	81.33±1.63	**62.30±0.98**	78.14±1.31	65.29±0.36
CausalWalk	45.52±1.99	34.15±0.97	71.49±1.65	71.55±2.46	71.27±2.48	57.05±0.62	80.65±0.10	71.22±1.74
GPT2-PPL	25.94±1.00	25.58±0.31	28.84±0.14	29.00±0.42	27.69±1.56	30.35±1.24	54.17±0.05	54.14±0.01
ProToCo	42.63±1.62	21.51±1.22	36.68±0.80	27.76±1.35	52.94±2.54	26.75±0.91	40.12±0.51	30.78±0.85
ProgramFC	46.04±1.42	32.12±0.76	62.49±1.74	71.63±0.91	60.17±3.34	53.67±1.92	86.84±0.84	69.41±2.07
P-Tuning v2	52.54±0.55	36.94±0.13	73.03±1.76	75.60±3.01	76.56±1.77	55.48±2.04	87.01±0.36	68.87±0.76
JustiLM	47.33±3.81	33.27±1.98	58.75±3.08	60.03±1.60	69.63±1.53	51.78±0.80	81.33±1.97	65.49±0.65
CORRECT	**59.88±2.03**	**44.25±1.73**	**75.34±1.02**	**80.59±1.00**	**83.20±0.80**	60.26±1.31	**88.41±0.19**	**74.95±0.38**

Table 4: Verdict prediction results on *fully supervised* setting with *Micro F1* score. Results are in percentage.

Model	BearFact		Check-COVID		SciFact		FEVEROUS-S	
	Gold	Retrieved	Gold	Retrieved	Gold	Retrieved	Gold	Retrieved
KGAT	69.42±0.87	57.36±0.52	72.05±1.58	76.47±0.65	74.44±0.96	62.33±0.88	86.21±0.28	67.99±0.78
HESM	63.68±1.39	58.62±0.32	63.47±0.25	71.90±1.85	72.44±0.77	53.36±2.33	83.30±0.75	68.36±0.86
Transformer-XH	61.26±0.72	56.55±1.50	68.56±0.87	76.91±1.51	75.89±0.51	58.67±1.53	85.61±0.80	69.78±0.41
Transformer-XH++	64.02±1.44	58.39±1.11	70.60±0.67	78.65±0.38	77.78±0.69	60.56±1.83	85.52±0.39	70.37±0.77
MultiVerS	62.93±1.17	50.69±1.46	66.65±1.71	70.70±1.73	83.68±1.40	**66.77±0.14**	83.57±1.54	67.66±1.65
CausalWalk	69.31±1.69	60.00±0.69	71.86±1.54	71.68±2.48	77.34±2.30	59.00±1.20	86.42±0.92	71.51±1.66
GPT2-PPL	40.00±2.43	39.49±0.73	32.75±0.62	32.54±0.93	31.50±0.71	31.24±1.56	54.33±0.06	54.23±0.01
ProToCo	56.03±0.24	35.57±2.35	37.12±1.10	32.75±2.31	60.00±1.53	31.17±0.71	54.21±0.67	44.71±1.95
ProgramFC	62.00±2.83	54.63±3.66	65.50±2.12	72.36±1.85	65.40±3.78	59.76±3.54	86.48±0.33	69.50±2.12
P-Tuning v2	70.69±0.49	60.34±0.15	72.93±1.85	77.32±1.96	80.34±0.94	57.44±1.83	87.16±0.33	70.56±0.54
JustiLM	62.41±1.25	48.49±2.21	59.71±2.56	61.71±2.05	72.20±1.85	54.74±0.82	81.60±0.33	68.38±1.45
CORRECT	**74.60±1.11**	**61.84±0.11**	**75.33±0.93**	**80.83±0.76**	**85.17±0.71**	63.50±1.17	**88.51±0.19**	**75.35±0.28**

5 instances from training set, obtaining $5 \times |\mathcal{Y}|$ training instances. This setting is consistent with existing work (Zeng and Gao, 2023). For a fair comparison, we sample instances using 5 random seeds. We keep the same sampling for our model and baselines. We report the results on test set.

Gold v.s. Retrieved evidence. For gold evidence setting, we observe the ground-truth evidence sentences, and we verify the claim based on the gold sentences. For retrieved evidence setting, we do not observe any evidence sentences, and retrieve sentences from an evidence corpus, based on which we make prediction. We follow (Pan et al., 2023) and use BM25 (Robertson et al., 2009) to retrieve top-3 evidence sentences for each claim. In the original Check-COVID dataset, if a claim is labeled as REFUTE based on the evidence, this claim is *reused* in NEI class with another random evidence. Thus, there are two claims with the same content, but different evidence and labels. However, in our retrieved evidence setting, both claims will receive the same retrieved evidence, but they are labeled differently, making model training inconsistent. Thus, for retrieved evidence setting, we remove claims in NEI class for Check-COVID.

4.1 Empirical Evaluation

Fully supervised setting. We follow (Wadden et al., 2022) and report Macro F1 score for both gold and retrieved evidence settings in Table 3. We also show Micro F1 score in Table 4. Transformer-XH++ consistently outperforms Transformer-XH, verifying that contextual and referential documents bring useful information. By comparing CORRECT to Transformer-XH++, we design evidence-conditioned prompting to integrate evidence and claim embeddings, and further improve the performance. Models with handcrafted prompt do not predict verdict as accurately as our model, which showcases the advantage of continuous prompt embeddings. Overall, the results on gold evidence setting are higher than on retrieved evidence setting, because the retrieved evidence sentences may not be always correct and may contain noisy information. The only exception is Check-COVID, because the retrieved evidence setting has only two labels, making the prediction task easier. MultiVerS is slightly better than CORRECT on SciFact, because the evidence sentences in SciFact contain sufficient information for fact-checking as shown in (Wadden et al., 2020, 2022), and referential documents do

Table 5: Verdict prediction results on *5-shot* setting with *Macro F1* score. Results are in percentage.

Model	BearFact		Check-COVID		SciFact		FEVEROUS-S	
	Gold	Retrieved	Gold	Retrieved	Gold	Retrieved	Gold	Retrieved
KGAT	36.62±2.28	29.92±3.99	35.65±4.56	47.81±2.40	39.07±2.06	35.12±2.79	50.21±0.95	50.68±1.21
HESM	35.40±3.77	26.00±2.34	35.41±4.78	42.82±6.50	38.87±1.69	34.03±5.61	51.36±0.35	51.92±0.39
Transformer-XH	29.45±2.49	31.69±2.06	40.48±2.73	49.24±1.60	47.65±3.99	33.47±1.11	52.45±2.71	49.41±1.93
Transformer-XH++	31.34±4.07	29.74±1.30	38.73±1.35	50.56±0.64	47.53±0.65	33.79±1.87	58.19±0.75	52.78±1.60
MultiVerS	24.34±3.12	20.92±0.48	32.16±2.50	50.80±1.78	**52.29±1.92**	29.64±1.53	38.26±0.18	38.82±0.37
CausalWalk	32.01±3.35	31.10±1.56	31.73±5.13	43.79±3.25	39.48±5.51	34.95±5.28	59.46±1.78	55.37±5.55
GPT2-PPL	24.99±0.90	26.28±0.18	25.05±4.47	23.89±2.65	27.69±0.41	27.45±0.66	51.33±2.55	51.54±2.50
ProToCo	35.11±0.40	21.51±0.78	35.62±5.32	29.72±3.85	48.68±3.38	25.93±5.60	40.48±0.88	31.00±0.57
ProgramFC	31.42±1.20	30.88±1.98	36.17±0.73	49.06±1.14	48.69±0.46	33.18±0.89	49.13±2.57	51.62±0.62
P-Tuning v2	35.68±2.36	31.86±0.33	38.90±4.81	50.63±4.22	43.94±0.54	33.33±2.48	56.70±1.82	48.53±2.23
JustiLM	31.38±2.07	26.01±2.08	36.48±2.78	44.39±2.41	44.42±2.08	31.04±1.47	45.35±1.18	42.48±1.02
CORRECT	**40.91±1.42**	**33.47±0.46**	**40.77±1.19**	**52.40±1.21**	49.12±0.30	**35.30±1.05**	**61.00±1.95**	**57.04±0.68**

Table 6: Verdict prediction results on *5-shot* setting with *Micro F1* score. Results are in percentage.

Model	BearFact		Check-COVID		SciFact		FEVEROUS-S	
	Gold	Retrieved	Gold	Retrieved	Gold	Retrieved	Gold	Retrieved
KGAT	44.66±0.25	36.78±3.86	37.55±2.86	50.98±1.13	42.22±3.06	36.78±3.52	51.13±1.55	51.66±1.38
HESM	48.85±2.42	28.97±3.26	36.68±5.15	50.11±3.22	39.56±2.45	35.89±4.67	56.33±0.91	53.68±0.37
Transformer-XH	32.53±3.13	40.80±3.32	41.67±2.19	51.63±1.16	48.89±2.84	35.11±1.95	52.94±2.65	51.56±0.73
Transformer-XH++	37.93±4.10	35.06±3.66	41.40±1.78	52.41±1.22	50.00±0.85	36.67±1.65	59.97±1.50	53.23±1.19
MultiVerS	40.86±0.74	39.49±1.70	41.01±1.29	49.82±1.56	**54.99±1.90**	43.33±1.74	51.39±1.33	51.84±1.54
CausalWalk	45.52±3.47	41.38±3.24	37.70±4.59	43.79±3.25	44.02±2.70	41.78±2.71	60.60±1.98	55.20±4.18
GPT2-PPL	36.38±4.14	40.69±0.49	33.19±0.67	34.62±0.72	29.44±2.50	29.00±1.46	53.02±1.11	53.11±1.12
ProToCo	51.03±2.44	33.80±2.44	41.05±2.47	34.50±2.31	51.55±2.27	36.78±1.35	54.72±1.32	42.45±2.73
ProgramFC	38.45±2.19	38.28±0.49	37.55±0.38	50.00±1.08	49.93±0.36	36.17±1.25	52.94±2.67	51.94±0.25
P-Tuning v2	48.70±1.95	41.90±3.10	41.05±3.88	52.07±3.09	45.78±1.07	37.67±1.85	58.02±1.68	52.06±0.76
JustiLM	42.70±1.64	37.72±5.08	40.82±2.20	46.73±1.53	47.41±1.07	33.00±1.58	52.54±1.71	49.38±1.60
CORRECT	**51.72±1.04**	**42.76±0.73**	**43.37±1.82**	**54.46±0.76**	53.00±1.30	**44.36±0.84**	**63.33±0.91**	**57.14±0.82**

not bring much additional benefit.

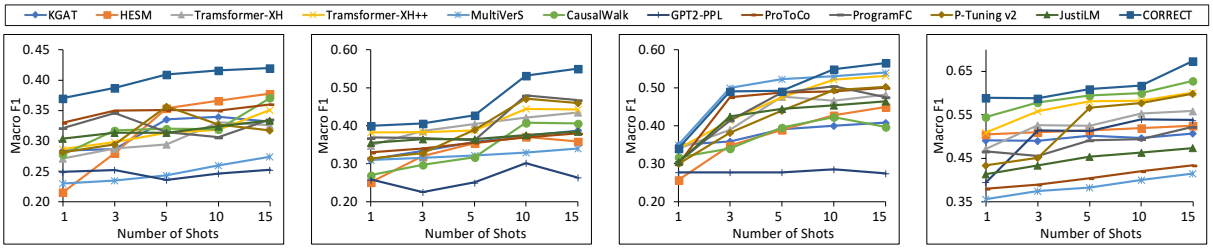
Few-shot setting. We report 5-shot results in Table 5 for Macro F1 score and Table 6 for Micro F1 score. Overall, HESM and Transformer-XH perform better than others, since referential documents contain useful information to complement evidence sentences for accurate prediction. Our model further improves them, verifying the strength of both contextual and referential documents. P-Tuning v2 outperforms models with handcrafted prompt, since continuous prompt embeddings can better adapt to the training data. By comparing to it, we design an evidence-conditioned prompt encoder to integrate contextual and referential documents into prompt embeddings, and produce more accurate results. We vary the number of shots in $\{1, 3, 5, 10, 15\}$ in Fig. 3. Though our model is competitive with MultiVerS on SciFact, we are still better than it on other datasets, due to the advantage of both contextual and referential documents.

4.2 Model Analysis

Effect of intra- and cross-layer reasoning. We respectively remove each graph layer from the complete model. Macro F1 score is shown in Fig. 4(a). The model with all three layers performs the best, indicating that all three layers bring useful information. Intra-layer reasoning on evidence sentence layer plays the most important role, since evidence sentences provide the most immediate information for verification. Contexts and references are important, and disregarding them deteriorates the results.

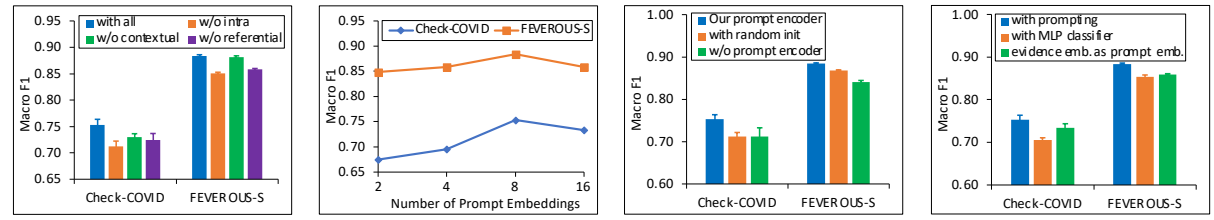
Different numbers of prompt embeddings M. We vary the number of prompt embeddings M in Fig. 4(b). When $M = 2$, we cannot fully capture the interaction between evidence and claims, causing a low accuracy. After we increase M, we observe an improvement. An overly high M hurts the result, because overfitting problem appears.

Prompt initialization and encoder. Our prompt encoder has both initialization of base prompt embeddings and evidence-conditioned prompt encoder. ***i***) To test the effect of initialization, we replace it with random initialization and report the results in Fig. 4(c). Our initialization produces better results, because evidence graph separates different sets of prompt embeddings and provides a more informative starting point. ***ii***) We remove



(a) BearFact (b) Check-COVID (c) SciFact (d) FEVEROUS-S

Figure 3: Few-shot veracity prediction with different number of shots.



(a) Effect of intra and cross reasoning (b) Number of prompt embeddings M (c) Prompt initialization and encoder (d) Effect of prompting

Figure 4: Model analysis on Check-COVID and FEVEROUS-S.

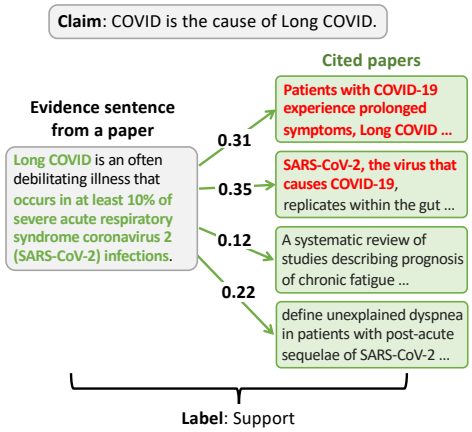


Figure 5: Case study on BearFact dataset.

prompt encoder from our model, and only retain base prompt embeddings. Fig. 4(c) shows that removing prompt encoder hurts the results, indicating that prompt encoder is necessary to combine evidence and claim for accurate prediction.

Effect of prompting. We design two ablated models. ***i***) We replace prompting with an MLP classifier, which concatenates evidence and claim embeddings as input, and produces predicted label. Here claim embedding is obtained without prompt embeddings. ***ii***) We directly consider evidence embedding as prompt embedding, and do not assume base prompt embeddings. Fig. 4(d) shows that prompting performs better than MLP classifier, because prompt embeddings and claim token embeddings are input to claim encoder together, and the contextualized encoding helps exchange information between evidence and claim for accurate prediction. Base prompt embeddings are also helpful, since they store general fact-checking knowledge and help generalize across different claims.

Case study. To intuitively understand that our model captures useful information in referential documents, we conduct a case study and visually show the attention values between an evidence sentence and its cited papers in graph neural networks. Fig. 5 shows that the highest attention scores appear between the evidence sentence and referential documents that indeed contain useful information. This visualization verifies that referential documents are crucial to improve claim verification.

5 Conclusion

We propose a context- and reference-augmented reasoning and prompting model for fact-checking. To model contextual and referential documents, we construct a three-layer graph with intra- and cross-layer reasoning. To integrate evidence into claims, we design evidence-conditioned prompting, which produces unique prompt embeddings for each claim. A future work is to extend three-layer graph to a multi-modal graph for fact-checking.

Acknowledgments

This work was in part supported by NSF awards #1934782 and #2114824. Some of the research results were obtained using computational resources provided by NAIRR award #240336.

Limitations

Here we identify two limitations of our work in terms of dataset and evidence type.

Dataset. Our model is proposed to incorporate contextual and referential documents of evidence sentences. We assume that the contextual and referential documents of evidence sentences are available in the dataset, or the dataset provides identifiers for evidence sentences, such as PubMed ID, so that we can use these identifiers to search their contextual and referential documents online. In Appendix B, we provide details on how to use identifiers to obtain contextual and referential documents. If the given dataset does not provide contextual or referential documents, or the identifiers of evidence sentences are not available, our model will reason within evidence sentences for fact-checking.

Evidence type. Following existing textual fact-checking models, we propose our model to reason over textual evidence sentences only. Our model is not proposed for tabular or multi-modal evidence, thus cannot reason over these types of evidence for fact-checking. One potential future work would be to extend our three-layer evidence graph to a multi-modal graph for evidence reasoning.

Ethics Statement

We do not foresee any undesired implications stemming from our work. Conversely, we hope that our work can advance AI Ethics research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact extraction and VERification over unstructured and structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 3816–3830.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Linmei Hu, Tianchi Yang, Luhao Zhang, Wanjun Zhong, Duyu Tang, Chuan Shi, Nan Duan, and Ming Zhou. 2021. Compare to the knowledge: Graph neural fake news detection with external knowledge. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 754–763.

Bowen Jin, Wentao Zhang, Yu Zhang, Yu Meng, Xinyang Zhang, Qi Zhu, and Jiawei Han. 2023. Patton: Language model pretraining on text-rich networks. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2021. Towards few-shot fact-checking via perplexity. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1971–1981.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 4582–4597.

Xiangci Li and Nanyun Peng. 2021. A paragraph-level multi-task learning model for scientific fact-verification.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 61–68.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. 2020. S2orc: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983.

Nedjma Ousidhoum, Zhangdie Yuan, and Andreas Vlachos. 2022. Varifocal question generation for fact-checking. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2532–2544.

Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, and Preslav Nakov. 2023. Fact-checking complex claims with program-guided reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6981–7004.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Jiasheng Si, Yingjie Zhu, and Deyu Zhou. 2023. Exploring faithful rationale for multi-hop fact verification via salience-aware graph learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13573–13581.

Shyam Subramanian and Kyumin Lee. 2020. Hierarchical evidence set modeling for automated fact extraction and verification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7798–7809.

Zhixing Tan, Xiangwen Zhang, Shuo Wang, and Yang Liu. 2022. Msp: Multi-stage prompting for making pre-trained language models better translators. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 6131–6142.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550.

David Wadden, Kyle Lo, Lucy Lu Wang, Arman Cohan, Iz Beltagy, and Hannaneh Hajishirzi. 2022. Multivers: Improving scientific claim verification with weak supervision and full-document context. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 61–76.

Gengyu Wang, Kate Harwood, Lawrence Chillrud, Amith Ananthram, Melanie Subbiah, and Kathleen Mckeown. 2023. Check-covid: Fact-checking covid-19 news claims with scientific evidence. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14114–14127.

Zhihao Wen and Yuan Fang. 2023. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 506–516.

Chenxi Whitehouse, Tillman Weyde, Pranava Madhyastha, and Nikos Komninos. 2022. Evaluation of fake news detection with knowledge-enhanced language models. In *Proceedings of the international AAAI conference on web and social media*, volume 16, pages 1425–1429.

Amelie Wuehrl, Yarik Menchaca Resendiz, Lara Grimminger, and Roman Klinger. 2024. What makes medical claims (un) verifiable? analyzing entity and relation properties for fact verification. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2046–2058.

Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh,

Guangzhong Sun, and Xing Xie. 2021. Graphformers: Gnn-nested transformers for representation learning on textual graph. In *Advances in Neural Information Processing Systems*, volume 34, pages 28798–28810. Curran Associates, Inc.

Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. 2024. Hypformer: Exploring efficient transformer fully in hyperbolic space. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3770–3781.

Fengzhu Zeng and Wei Gao. 2023. Prompt to be consistent is better than self-consistent? few-shot and zero-shot fact verification with pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4555–4569.

Fengzhu Zeng and Wei Gao. 2024. Justilm: Few-shot justification generation for explainable fact-checking of real-world claims. *Transactions of the Association for Computational Linguistics*, 12:334–354.

Ce Zhang and Hady W Lauw. 2020. Topic modeling on document networks with adjacent-encoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6737–6745.

Congzhi Zhang, Linhai Zhang, and Deyu Zhou. 2024a. Causal walk: Debiasing multi-hop fact verification with front-door adjustment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19533–19541.

Delvin Ce Zhang and Hady W Lauw. 2021. Semi-supervised semantic visualization for networked documents. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part III 21*, pages 762–778. Springer.

Delvin Ce Zhang and Hady W Lauw. 2023. Topic modeling on document networks with dirichlet optimal transport barycenter. *IEEE Transactions on Knowledge and Data Engineering*.

Delvin Ce Zhang, Menglin Yang, Rex Ying, and Hady W Lauw. 2024b. Text-attributed graph representation learning: Methods, applications, and challenges. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1298–1301.

Delvin Ce Zhang, Rex Ying, and Hady W Lauw. 2023. Hyperbolic graph topic modeling network with continuously updated topic tree. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3206–3216.

Zhiwei Zhang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. 2021. Abstract, rationale, stance: A joint model for scientific claim verification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3580–3586.

Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Transformer-xh: Multi-evidence reasoning with extra hop attention. In *International Conference on Learning Representations*.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6170–6180.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. Gear: Graph-based evidence aggregating and reasoning for fact verification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 892–901.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348.

Anni Zou, Zhuosheng Zhang, and Hai Zhao. 2023. Decker: Double check with heterogeneous knowledge for commonsense fact verification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11891–11904.

A Pseudo-code of Training Process

We summarize the training process at Algo. 1.

B Dataset Preprocessing Details

Here we present details of dataset preprocessing.

FEVEROUS[1] (Aly et al., 2021) is a general-domain dataset, and each claim is annotated in the form of sentences and/or cells from tables in Wikipedia pages. In this paper we mainly focus on textual evidence sentences, thus we follow ProgramFC (Pan et al., 2023) and obtain claims that only require textual evidence for verification, and name this subset **FEVEROUS-S**. Claims in this dataset have two labels only, SUPPORT and REFUTE. In the original dataset, each evidence sentence may contain hyperlinks to other Wikipedia pages, and such hyperlinks in sentences are indicated with double square brackets. We thus retrieve words or phrases inside double square brackets, and use them as entries to query Wikipedia dump

[1] https://fever.ai/dataset/feverous.html

Algorithm 1 Training Process of CORRECT

Input: A fact-checking dataset $\mathcal{D}$ with claims $\mathcal{X}$, evidence sentences $\mathcal{E}$, contextual documents $\mathcal{C}$, and referential documents $\mathcal{R}$. Number of prompt embeddings M and temperature τ.

Output: Predicted labels $\widehat{\mathcal{Y}}_{\text{test}}$ for test claims.

Initialize model with pre-trained parameters in biomedical domain or general domain.
while not converged **do**
  Construct three-layer evidence graph for each claim $x \in \mathcal{X}$.
  for evidence sentence $e \in \mathcal{N}_{\text{evid}}(x)$ **do**
    Initialization $\mathbf{H}_e^{(l=1)} = \text{TRM}(\mathbf{H}_e^{(l=0)})$.
    for $l = 1, 2, ..., L-1$ **do**
      // Evidence graph reasoning
      Intra-layer reasoning by Eqs. 1–4.
      Cross-layer reasoning by Eq. 5.
      // Asymmetric MHA step
      Virtual token concatenation Eq. 6.
      $\mathbf{H}_e^{(l+1)} = \text{TRM}^{\text{asy}}(\widehat{\mathbf{H}}_e^{(l)})$ by Eq. 7.
    end for
  end for
  Obtain an evidence embedding by Eq. 8.
  // Evidence-conditioned prompting
  Initialize $|\mathcal{Y}|$ sets of base prompt embeddings $\{\mathbf{h}_{m,y}\}_{m=1}^{M}$ where $y \in \mathcal{Y}$.
  Input evidence embedding $\mathbf{h}_E$ to evidence-conditioned prompt encoder and obtain $|\mathcal{Y}|$ sets of prompt embeddings $\{\boldsymbol{\pi}_{m,y}\}_{m=1}^{M}$ where $y \in \mathcal{Y}$ by Eqs. 10–11.
  Input $\{\mathbf{P}_{x,y}\}_{y\in\mathcal{Y}}$ to claim encoder and obtain $|\mathcal{Y}|$ claim embeddings $\{\mathbf{h}_{x,y}\}_{y\in\mathcal{Y}}$.
  Minimize loss function $\mathcal{L}$ in Eq. 13.
end while

to obtain their corresponding pages as referential documents. FEVEROUS uses the December 2020 dump, including 5.4 million full Wikipedia articles. If a Wikipedia page has overly many sentences, we reserve its top-20 sentences, since almost all the evidence sentences appear within top-20 sentences in FEVEROUS. Similarly, the full content of the Wikipedia page is contextual document of each evidence sentence. If a page has overly long content, we reserve its top-20 sentences.

BearFact[2] (Wuehrl et al., 2024) is a biomedical claim verification dataset. Evidence sentences are obtained from paper abstracts in PubMed database[3]. Original dataset does not provide evidence sentences for claims in NEI class. Thus we follow existing work (Zeng and Gao, 2023) and select evidence sentences that have the highest *tf-idf* similarity with claims as their evidence. We consider the full abstract as the contextual document for each evidence sentence, as in MultiVerS (Wadden et al., 2022). In addition, we use S2ORC (Lo et al., 2020) to obtain cited papers with abstracts as referential documents. Specifically, the original dataset provides PubMed ID for each evidence sentence. We use PubMed IDs as identifiers to search in S2ORC database and obtain cited papers. If a paper has overly many citations, we reserve its top-20 citations to avoid data redundancy.

Check-COVID[4] specifically focuses on COVID-19 claims taken from news articles. Each evidence sentence is from a paper abstract with CORD ID as identifier. We thus use CORD IDs to search in S2ORC database and obtain cited papers. Similarly, we consider the full abstract as contextual document. The original dataset provides sentences for claims in NEI class.

SciFact[5] (Wadden et al., 2020) is another biomedical fact-checking dataset with sentences in paper abstracts as evidence. Similarly, the full content of the abstract is considered as contextual document. In addition, each evidence sentence is coupled with S2ORC ID, which is used to obtain its citations using S2ORC database. The original dataset does not have sentences for claims in NEI class. Thus we follow the original paper (Wadden et al., 2020) and choose top-3 sentences in the same abstract with the highest *tf-idf* similarity with the claim as evidence.

C Experiment Environment

All the experiments were conducted on Linux server with 4 NVIDIA A100-SXM4-80GB GPUs. Its operating system is 20.04.5 LTS (Focal Fossa). We implemented our proposed model CORRECT using Python 3.9 as programming language and PyTorch 1.14.0 as deep learning library. Other frameworks include numpy 1.22.2, sklearn 0.24.2, and transformers 4.43.3.

[2] https://www.ims.uni-stuttgart.de/en/research/resources/corpora/bioclaim/

[3] https://pubmed.ncbi.nlm.nih.gov/

[4] https://github.com/posuer/Check-COVID/tree/main/Check-COVID

[5] https://github.com/allenai/scifact/tree/master