BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 09/25 pp682–698
DOI: 10.26599/BDMA.2024.9020021
Volume 7, Number 3, September 2024

Collaborative Knowledge Infusion for Low-Resource Stance Detection

Ming Yan*, Tianyi Zhou Joey, and W. Tsang Ivor

Abstract: Stance detection is the view towards a specific target by a given context (e.g. tweets, commercial reviews). Target-related knowledge is often needed to assist stance detection models in understanding the target well and making detection correctly. However, prevailing works for knowledge-infused stance detection predominantly incorporate target knowledge from a singular source that lacks knowledge verification in limited domain knowledge. The low-resource training data further increase the challenge for the data-driven large models in this task. To address those challenges, we propose a collaborative knowledge infusion approach for low-resource stance detection tasks, employing a combination of aligned knowledge enhancement and efficient parameter learning techniques. Specifically, our stance detection approach leverages target background knowledge collaboratively from different knowledge sources with the help of knowledge alignment. Additionally, we also introduce the parameter-efficient collaborative adaptor with a staged optimization algorithm, which collaboratively addresses the challenges associated with low-resource stance detection tasks from both network structure and learning perspectives. To assess the effectiveness of our method, we conduct extensive experiments on three public stance detection datasets, including low-resource and cross-target settings. The results demonstrate significant performance improvements compared to the existing stance detection approaches.

Key words: parameter-efficient learning; low-resource stance detection; knowledge infusion

1 Introduction

Stance detection is the view towards a specific target with a given context, such as tweets or commercial reviews. Typically, those given contexts in stance detection tasks are mostly short-length contexts, which makes it challenging to predict the target's stance for the data-driven detection models with such limited information. Large Pretrained Language Models (PLMs) are becoming the default backbone to enhance the stance detection model with learned commonsense knowledge, leading to great success in this field[1, 2]. To further enrich the knowledge of specific targets, the straightforward approach is to incorporate the target-related background knowledge as extra supplementary knowledge for the pretrained stance detection model, which has been shown to substantially improve model performance[3, 4]. In detail, those works infuse explicitly knowledge individually through knowledge graph[5, 6], Wikipedia[7, 8], generative knowledge[9, 10], leveraging PLMs' knowledge feature learning and representation capability by finetuning entire models' parameters. However, those knowledge-infuse solutions are quite inefficient in finetuning large PLM backbones on the limited training data. For instance, the few-shot or zero-shot stance detection dataset

• Ming Yan, Tianyi Zhou Joey, and W. Tsang Ivor are with Centre for Frontier AI Research, and Institute of High-Performance Computing, Agency for Science Technology and Research, Singapore 138632, Singapore. E-mail: mingy@cfar.a-star.edu.sg; Joey_Zhou@cfar.a-star.edu.sg; Ivor_Tsang@cfar.a-star.edu.sg.
* To whom correspondence should be addressed.
Manuscript received: 2023-08-13; revised: 2024-02-17; accepted: 2024-03-22

VAST[11] has very limited training data, and even some stance detection target has no training data. Besides the low-resource challenge, unbalanced dataset distribution is another challenge for the stance detection task, leading the training trajectory to fall into the local minima. Last but not least, we find some background knowledge is not always infused correctly in the knowledge infusion process. This is because a single knowledge source in previous works cannot fully cover and support enough knowledge for diverse targets[4]. For example, the target 'breaking the law' on Wikipedia is erroneously linked to a heavy metal music song rather than its ground truth definition of engaging in activities contrary to the law.

To address the aforementioned challenges, we propose a novel collaborative knowledge-infused stance detection method for training the large detection model in the low-resource setting efficiently. Specifically, we introduce a retrieval-based knowledge verifier that mitigates incorrect knowledge infusion by selecting the rich-semantic background knowledge from different knowledge sources, rather than relying on a single knowledge source. Furthermore, we present a trainable collaborative adaptor integrated into PLMs to enable efficient parameter learning in low-resource stance detection tasks. Concretely, the collaborative adaptor freezes the parameter weights of large PLM and finetunes the parameter-efficient adaptor only, which alleviates the overfitting effects on large PLM in low-resource scenarios. However, we empirically find that intuitively adding adaptors into PLM may lead to unstable training in the new stance detection tasks. This because that the initialized weights of the collaborative adaptor cannot work well with the pretrained PLMs in the early finetuning stage of new tasks. Moreover, the unbalanced data distribution further impacts the stable training. Thereby, we design a staged optimization algorithm for the adaptive model training in unbalanced distributions. The primary objective of the first optimization stage is to prevent the training trajectory from converging to a local minimum leading to unexpected performance. In the second stage, our model introduces a weighted cross-entropy loss to balance the biased stance categories and further improve the model performance in low-resource stance detection tasks. In other words, we progressively use label smooth (Stage 1) and weighted loss (Stage 2) separately to reduce the overfitting effects in our low-resource stance detection tasks, which is different from traditional optimization paradigms using those two without dynamic adjustments.

We conduct extensive experiments on three public stance detection datasets, encompassing the low-resource stance detection, and cross-target stance detection tasks. Experimental results demonstrate the superior performance of our method compared to the state-of-the-art approaches across all stance detection tasks. The contributions of our work are summarized as follows.

- We introduce a collaborative knowledge verification module to assist the detection model in selecting high related semantic knowledge from different knowledge sources. To the best of our knowledge, this is the first work to infuse verified knowledge into the knowledge enhancement stance detection task.
- We introduce a collaborative adaptor in an efficient way for the low-resource setting. It contains three sub-components, which are architecturally located in different positions of the backbone model, learning different features collaboratively.
- To alleviate the unbalanced effects of low-resource stance detection tasks, we also provide a staged optimization algorithm to improve the training efficiency in large PLMs. Experiments show the superiority of our method in different low-resource settings and outperforms state-of-the-art approaches on three public stance detection datasets.

2 Related Work

2.1 Knowledge enhancement

Knowledge enhancement increases the capabilities in thinking, understanding, and reasoning for the data-driven models beyond the original training data. In recent years, there has been a growing trend in infusing external-specific knowledge as complementary knowledge to the large pretrained models[12]. Depending on the infused knowledge, knowledge infusion methods can be broadly categorized into structured-knowledge infusion (e.g., knowledge graph) and unstructured-knowledge infusion (e.g., Wikipedia).

Domain-specific experts typically collect structured knowledge, encompassing well-organized and rich knowledge. For instance, CKE-Net[5] utilizes the structured knowledge base (ConceptNet) to enhance its model's common-sense knowledge in zero-shot or few-

shot stance detection tasks. Similarly, K-BERT[13] incorporates domain knowledge through entity triplets obtained from the knowledge graph. Other methods, like JAKET[14], ERNIE[15] and, Entity-as Experts[16], also infuse knowledge from knowledge bases through grounding knowledge with entity linking technologies. Structured knowledge provides well-organized and domain-specific knowledge for specific targets in stance detection tasks. However, its utility is limited by the pre-defined scope of available knowledge, which may not cover all targets encountered in practical scenarios.

In contrast, unstructured knowledge offers more flexibility and can be easily collected from a wide range of diverse domains. For instance, the VAST[11] dataset introduces thousands of diverse targets that mostly cannot be found in the well-constructed structured knowledge. To incorporate unstructured knowledge into the stance detection models, WS-BERT[4] directly infuses external knowledge from Wikipedia as its inputs to pretrained models for stance detection in the VAST dataset. Another knowledge infusion paradigm is to finetune PLMs on the specific domain corpus to embed the domain-specific knowledge, as demonstrated by SciBERT[17], Bio-BERT[18], and BERTweet[19]. In addition to domain-specific finetuning, Self-talk[20] offers another interesting solution by exploring knowledge from its own training corpus with hand-crafted prompts, enhancing language model learning with task-related knowledge. Furthermore, DDP[21] and K-Former[22] present the retrieval-based knowledge infusion methods by retrieving knowledge from feature pools and online websites, respectively. Nevertheless, more efforts are still needed to collaborate structured and unstructured knowledge together in a correct and efficient manner for large PLM-based models, particularly in low-resource tasks.

2.2 Stance detection

Stance detection refers to the identification of attitudes toward a specific context or topic, typically framed as a stance classification problem (positive, negative, and neutral) for the neural network-based models. Stance detection encompasses various tasks depending on the specific topics involved, including rummer stance detection[23], fake news stance detection[24], disinformation or misinformation stance detection[25], multi-language and cross-language stance detection[26, 27], and zero-shot stance detection[11], etc. In this study, we focus on in stance detection tasks of background knowledge infusion and low-resource training. To infuse background knowledge, existing approaches try to incorporate knowledge from different sources. For instance, CKE-Net[5] introduces target-related knowledge from ConceptNet, which is trained on the common sense knowledge graph. Similarly, BS-GGCN[28] simplifies the whole concept-net graph to a compact sentence related graph, enabling more efficient knowledge embedding for stance detection. Moreover, WS-BERT[4] leverages the background knowledge from Wikipedia pages as its additional input to improve the model performance in stance detection. Regarding the low-resource challenge, STCC[29] employs contrastive learning to enhance target representation in low-resource stance detection tasks. Different from STCC building contrastive examples in the existence of the target, JointCL[30] builds contrastive examples from the prototype graph representation of the target's link, which further improves the model performance on the unseen targets. While most of those works solve the knowledge infusion and low-resource task separately, it is essential to consider two challenges together to improve the performance of stance detection models.

3 Methodology

Before delving into our methodology, we define the notations of the stance detection task as follows.

Given a context set C with element c_i, where $i = 1, 2, \ldots, n_c$, and a target set $\mathcal{T}$ with element t_j, where $j = 1, 2, \ldots, n_t$. The stance detection task is formulated to predict the stance y that maximizes the prediction possibility of $P(y|\{C,\mathcal{T}\})$, where the stance set is $Y = \{\text{Positive, Negative, Neutral}\}$. Regarding the knowledge enhancement stance detection task, its objective is defined as maximizing $P(y|\{C,\mathcal{T},\mathcal{K}\})$, where $\mathcal{K}$ denotes the infused knowledge that assists in the stance detection task.

The overview of our proposed methodology, as illustrated in Fig. 1, contains three modules:

(1) Knowledge alignment, which aims to collaboratively select semantic target knowledge from structured and unstructured knowledge sources.

(2) Parameter-efficient learning, which involves collaborative adaptor and knowledge augmentation to enhance model performance in low-resource settings.

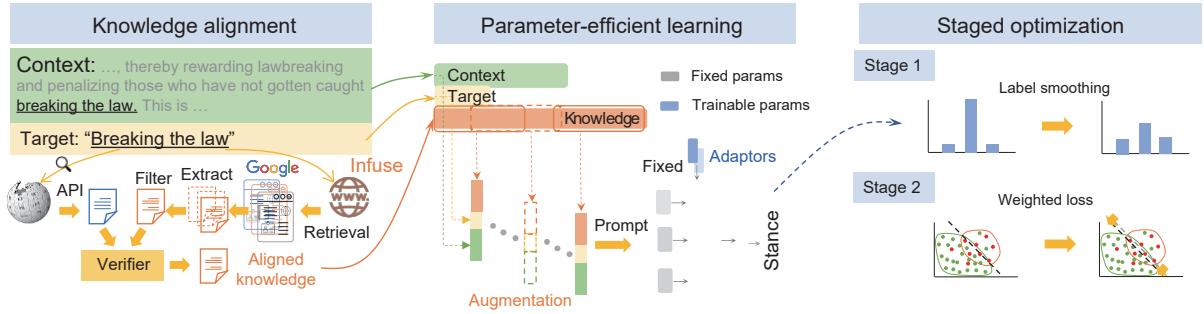


Fig. 1 Overview of our stance detection architecture: Knowledge alignment, parameter-efficient learning, and staged optimization. Knowledge alignment collaboratively selects semantic similar knowledge of the target from different knowledge sources. Parameter-efficient learning introduces the collaborative adaptor and knowledge augmentation into the stance detection model to perform low-resource learning. Staged optimization algorithm optimizes the classifier with label smoothing, then pushes the classifier edge aligning to data distribution with the help of weighted loss.

(3) Staged optimization algorithm, which further refines the adaptive model through strategies, such as label smoothing and weighted loss.

3.1 Knowledge alignment

We introduce a knowledge alignment module to assist the stance detection model infuse target-related background knowledge correctly, particularly for the targets that lack matching items in singular knowledge source (e.g., Wikipedia). To help the unmatched targets infuse knowledge out of Wikipedia, we incorporate retrieval knowledge from the Internet, specifically through Google search, as an additional knowledge source. Thus, our approach adopts a multi-source knowledge infusion paradigm across structured Wikipedia and unstructured Internet, which selects the semantic similar knowledge as the collaborative knowledge $\mathcal{K}$ to align to the target $\mathcal{T}$ from multiple knowledge sources. Figure 2 illustrates our knowledge alignment module.

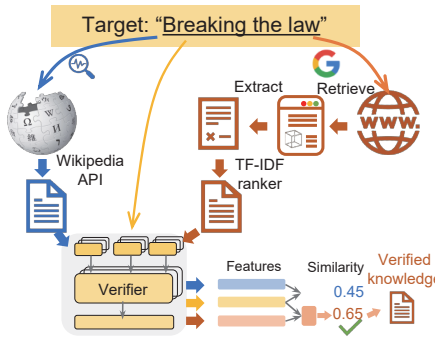


Fig. 2 Knowledge alignment. The target's collaborative knowledge is the knowledge with a higher semantic similarity score from Wikipedia or the Internet. The Wikipedia knowledge is obtained by Wikipedia's API. The Internet knowledge is obtained by Google retrieval.

In the paradigms of knowledge enhancement stance detection, detection models mostly infuse extra knowledge through the target ($\mathcal{T}$) rather than the given context (C). There are two reasons. Firstly, the stance detection task aims to identify the stance (Y) of the target, which may not be explicitly mentioned in the given context. Consequently, the target contains more information compared to the given context. Secondly, the context is typically long and complex, making it challenging to locate target-related information for infusing background knowledge ($\mathcal{K}$). In our proposed collaborative knowledge infusion approach (see Fig. 2), we collaboratively incorporate the background knowledge into the detection model by retrieving the target-related knowledge from Wikipedia and the Internet.

For structured Wikipedia knowledge, we utilize the target as the keywords to retrieve background knowledge through Wikipedia's API (https://github.com/goldsmith/Wikipedia). This API returns a summary of the matched Wikipedia page. In cases where there is no match for a target, we follow the setting of Ref. [4] and consider the target itself as the knowledge without introducing any additional information.

For unstructured Internet knowledge, we retrieve the target-related web pages by using the searching prompt "What is the definition of TARGET ($\mathcal{T}$)?" as the search term for the Google search engine. Subsequently, we select the top three pages from the Google search results and employ BeautifulSoup (https://git.launchpad.net/beautifulsoup) to parse the HTML contents of these pages into candidate passage

lists ($\mathcal{D}$). The next step involves filtering out unrelated contexts from the candidate passage lists, as web pages often contain noise and extraneous information. To accomplish this, we utilize Term Frequency Inverse Document Frequency (TF-IDF) ranker to identify and exclude noisy passages from a long list of candidate passages,

$$\text{TF}-\text{IDF} = \text{TF} \times \text{IDF} \tag{1}$$

where TF is term frequency, and IDF is inverse document frequency. Once the knowledge related to the target has been collected from Wikipedia and the Internet, we introduce the knowledge verifier to select more accurate knowledge from multiple sources as the infused knowledge. Knowledge verification involves feature encoding and feature similarity comparison, that selects semantic similar knowledge among different knowledge sources. Concretely, we employ Sentence-BERT[31] to encode the target $\mathcal{T}$ and its corresponding knowledge (Wikipedia knowledge $\mathcal{K}_{\text{w}}$ and Internet knowledge $\mathcal{K}_{\text{g}}$) into embedding features ($\mathcal{T}^{\text{em}}, \mathcal{K}_{\text{w}}^{\text{em}}, \mathcal{K}_{\text{g}}^{\text{em}}$),

$$\left\{\mathcal{T}^{\text{em}}, \mathcal{K}_{\text{g}}^{\text{em}}, \mathcal{K}_{\text{w}}^{\text{em}}\right\} = F\left(\left\{\mathcal{T}, \mathcal{K}_{\text{w}}, \mathcal{K}_{\text{g}}\right\}\right) \tag{2}$$

where $\mathcal{F}(\cdot)$ is the embed function. Then, we compute the semantic similarity between the stance target ($\mathcal{T}^{\text{em}}$) and different knowledge sources $\mathcal{K}_{\text{g}}^{\text{em}}$ and $\mathcal{K}_{\text{w}}^{\text{em}}$ using the classical cosine similarity as follow:

$$S\left(\mathcal{T}^{\text{em}}, \mathcal{K}_{\text{w}}^{\text{em}}\right) = \frac{\mathcal{T}^{\text{em}} \mathcal{K}_{\text{w}}^{\text{em}}}{|\mathcal{T}^{\text{em}}| \times \left|\mathcal{K}_{\text{w}}^{\text{em}}\right|} \tag{3}$$

$$S\left(\mathcal{T}^{\text{em}}, \mathcal{K}_{\text{g}}^{\text{em}}\right) = \frac{\mathcal{T}^{\text{em}} \mathcal{K}_{\text{g}}^{\text{em}}}{|\mathcal{T}^{\text{em}}| \times \left|\mathcal{K}_{\text{g}}^{\text{em}}\right|} \tag{4}$$

where $|\cdot|$ is the L1 norm. Finally, we select the knowledge with the highest semantic similarity as the collaborative knowledge $\mathcal{K}$ to be infused into our model, which is expressed as follows,

$$K = \text{argmax}\left\{S\left(\mathcal{T}^{\text{em}}, \mathcal{K}_{\text{w}}^{\text{em}}\right), S\left(\mathcal{T}^{\text{em}}, \mathcal{K}_{\text{g}}^{\text{em}}\right)\right\} \tag{5}$$

By collaborative integration of this verified knowledge, our stance detection model allows for the inclusion of reliable knowledge in stance detection tasks. This knowledge infusion manner expands the scope of target knowledge by incorporating information from both structured and unstructured knowledge sources. As a result, it addresses the limitations associated with relying on a single knowledge source, including the issues of out-of-scope knowledge and false infusions.

3.2 Efficient parameter learning

By collaborative integration of this verified knowledge, our stance detection model allows for the inclusion of reliable knowledge into stance detection tasks. This knowledge infusion manner expands the scope of target knowledge by incorporating information from both structured and unstructured knowledge sources. As a result, it addresses the limitations associated with relying on a single knowledge source, including the issues of out-of-scope knowledge and false infusions. But those training data are still the low-resource data for the large PLM finetuning. In this section, we proposed a new efficient parameter learning method.

Suppose our stance detection model's backbone is the pretrained PLM with collaborative adaptors. Specifically, the PLM with fixed parameter (W) and adaptors with trainable parameter (ΔW). The backbone model can be a generic language model BERT with Transformer architecture. In our low-resource stance detection task, the objective of the knowledge-infused stance detection task is formulated as follows:

$$J = \min \sum_{T, C}^{Y} -\log\left(P(Y|W, \Delta W, \{C, \mathcal{T}, \mathcal{K}\})\right) \tag{6}$$

where T is the target set, C is the content set, and Y is the stance set. During the training process, our approach keeps the parameters of the pretrained model fixed and trains the collaborative adaptor parameters only on the low-resource stance detection dataset. The fixed-weight setting prevents the catastrophic forgetting problem and mitigates the challenges associated with training large models on limited training data (C). Moreover, the collaborative adaptor has significantly fewer parameters than the pretrained PLM's parameters, $|\Delta W| \ll |W|$ (e.g., the prefix-tuning adaptor of our collaborative adaptor only has approximately 0.01% of the parameters W in BERT). This reduction in parameters greatly alleviates the data dependency for large model training.

The motivation of our collaborative adaptor is to introduce multiple adaptive modules that collaboratively work together to provide a stronger feature representation capability than the individual adaptors. Figure 3 presents the overview of our collaborative adaptor, which consists of gated low-rank adaptation, prefix-tuning, and attentive fusion modules.

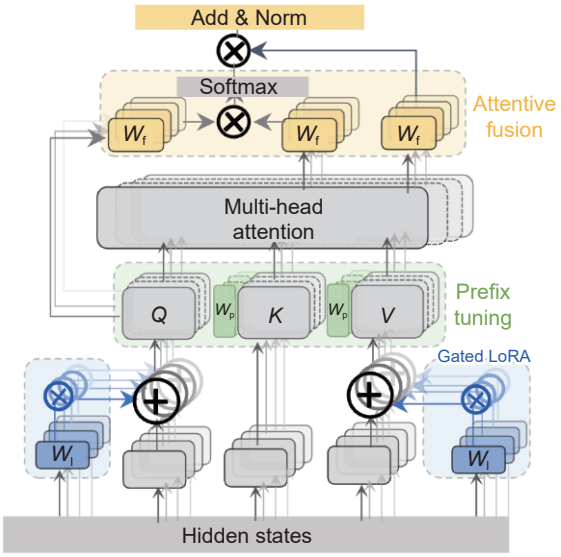


Fig. 3 Overview of collaborative adaptor in efficient parameter learning. *Q*, *K* ,*V* are the query, key, and value of the Transformer module, respectively.

Those adaptive modules are hierarchically incorporated into different levels of the Transformer architecture.

In detail, the gated low-rank adaptor (with parameter W_{l}) is inserted after the bottom hidden state layer of Transformer architecture. It maps the selective embedding features from previous layers. In the intermediate level of the Transformer architecture, the prefix-tuning (with parameter W_{p}) introduces additional trainable prefix tokens before the key K and value V to incorporate new task-specific information into the PLMs. At the top of Transformer architecture, we introduce the attentive fusion module (with parameter W_{f}) to further select task-related features from value V. Therefore, all adaptive modules collaborate to learn task-specific information and feature representations in a hierarchical manner.

More specifically, the gated low-rank adaptation not only maintains the parameter efficiency of the Low-Rank Adaptor (LoRA)[32], but also introduces a gate function to selectively incorporate the learned features from LoRA. This gate function empowers the vanilla LoRA with a similar attentive capability to the Transformers module, by passing through the learned features selectively. Here, we default set the gated function Φ to Sigmoid function. Mathematically, if we denote the output of fully connection layer with normalization as the hidden state H, the low-rank downscale metric W_{down} and upscale metric W_{up} map features in an efficient computing manner. Our gated LoRA is defined as follows:

$$I^{\mathrm{l}} = \Phi\left(W_{\mathrm{up}}^{\mathrm{T}} \cdot W_{\mathrm{down}} \cdot H\right) + W_{\mathrm{I}} \cdot H \tag{7}$$

where T denotes the transpose of matrix, $\mathrm{l} = \exists\{V, Q\}$, and I^{l} denotes the selected features by our gated LoRA. The LoRA features are then collaborative with the Transformer's features to form the key K, value V, and query Q through an additive operation correspondingly. To enhance the feature learning capacity of our detection model, we introduce an alternative approach where the mapped embeddings $\langle K, V, Q\rangle$ are not directly fed into the multi-head computation. Instead, we integrate prefix tokens T_Q and T_V along with selected key and value features into new configurations: $\langle T_Q; I^Q\rangle$ and $\langle T_V; I^v\rangle$. This modification aims to enrich the model's feature processing capabilities. The details of the multi-head attention computation are provided as follows:

$$Z_i = \mathrm{Attention}_i\left(K, \langle T_k; \langle T_Q; I^Q\rangle, \langle T_V; I^v\rangle\right) \tag{8}$$

where Z_i is the i-th output of the multi-head attention computation ($\mathrm{Attention}_i$). The prefix-tuning adaptors are introduced at the intermediate level of every Transformer, with significantly fewer trainable parameters compared to full parameter finetuning. To leverage the attentive mechanism, we introduce attentive fusion at the top of the Transformer, selectively activating the prefix-tuning features. As an attentive network, the key (K') and value (V') are derived from the outputs (Z_i) of the multi-head attention layer, and the query (Q') is obtained from the previous layer's query (Q) through a residual connection. The attentive fusion is performed using dot production ⊗. All computations of attentive fusion are listed as follows:

$$Z = \mathrm{Softmax}\left(Q' \otimes K'\right) \otimes V' \tag{9}$$

In this way, our collaborative adaptor performs efficient parameter learning across bottom embedding layers, middle of Transformer, and top feature fusion. All the efficient modules collaborate with each other to learn a generic representation for the low-resource stance detection tasks. The rest computing is performed in traditional manner with dense layer and classifier. More detailed computation please find in Algorithm 1.

Besides the collaborative adaptor in efficient parameter learning, we also introduce knowledge

Algorithm 1 Staged optimization algorithm

Initialize:

Target's knowledge $\mathcal{K}$, collaborative adaptor weights $W_* = \{W_\mathrm{l}, W_\mathrm{P}, W_\mathrm{f}\}$, total steps N, Stage 1 steps N_{s_1}, Stage 2 steps N_{s_2}, content $\mathcal{C}$, prompt $\mathcal{P}_\mathrm{t}$, label $Y_\mathrm{c} \in Y$, learning rate α_0, and loss weights θ_a and θ_b

Ensure:

Set input $< \mathcal{C}, \mathcal{T}, \mathcal{K} >$

Repeat
  for index $= 1$ to N **do**
    Fetch train data batch $< \mathcal{C}, \mathcal{T}, Y >$;
    Set augmentation $\mathcal{K}_\mathrm{sub} \leftarrow < \mathcal{T}, \mathcal{K} >$;
    Set prompt $\mathcal{P} \leftarrow \mathcal{P}_\mathrm{t}$;
    Stage 1:
    if index $\leqslant N_{s_1}$ **then**
      Set smooth factor ε;
    else
      $\varepsilon \leftarrow 0$;
    end if
    Stage 2:
    if index $\geqslant N_{s_2}$ **then**
      $\theta \leftarrow \theta_\mathrm{a}$;
    else
      $\theta \leftarrow \theta_\mathrm{b}$;
    end if
    Feed forward computing:
      $\tilde{Y} \leftarrow \mathrm{model}\ (\mathcal{C}, \mathcal{P}_\mathrm{t}, \mathcal{K}_\mathrm{sub} | W, W_*)$;
      $\tilde{Y}_\mathrm{c} \leftarrow \mathrm{Softmax}\ (\tilde{Y})$;
    Compute cost function:
      $\mathrm{CE}\ (Y, \tilde{Y}) \leftarrow -\sum_{c=1}^{c} \theta \cdot Y_\mathrm{c} \log(\tilde{Y}_\mathrm{c})$;
      $J(Y, \{\tilde{Y}\}) \leftarrow -(1-\varepsilon)\log(1-\mathrm{CE}(Y, \{\tilde{Y}\}) - \varepsilon \cdot \log(\mathrm{CE}(Y, \{\tilde{Y}\}))$;
    Compute gradient:
      $$\Delta W \leftarrow \frac{\partial J(Y, \tilde{Y})}{\partial W_*};$$
    Update gradient:
      $$W_* \leftarrow W_* + \alpha \Delta W;$$
  end for
until convergence

augmentation to facilitate efficient parameter learning. In the knowledge infusion module, the crawled knowledge obtained from the Internet often exceeds the maximum input length of backbone PLMs. Our knowledge augmentation approach involves slicing the lengthy knowledge content into properly segmented parts to help the detection model capture the complete semantics of the infused knowledge.

Unlike previous approaches in knowledge-infused stance detection that infuses knowledge ($\mathcal{K}$) following the paradigm:

$$[\mathcal{T}, \mathcal{C}, \langle\mathrm{SEP}\rangle, \mathcal{K}, \langle\mathrm{CLS}\rangle]$$

or

$$[\mathcal{T}, \langle\mathrm{SEP}\rangle, \mathcal{C}, \langle\mathrm{SEP}\rangle, \mathcal{K}, \langle\mathrm{CLS}\rangle],$$

where ⟨SEP⟩ and ⟨CLS⟩ are separate token and ending token for the PLMs, respectively. We reformulate our knowledge infusion paradigm into

$$[P_\mathrm{t}, \mathcal{C}, \langle\mathrm{SEP}\rangle, \mathcal{K}_\mathrm{sub}, \langle\mathrm{CLS}\rangle].$$

Our input paradigm employs the prompt P_t: [What's the stance of $\mathcal{T}$ in following context?] instead of the target $\mathcal{T}$ to fully leverage the reasoning capability of PLMs, which matches the pretrain input format of two sentences split by ⟨SEP⟩, as well as keeping the semantic integrity.

In detail, we conduct the knowledge augmentation by slicing the long collaborative knowledge content into sub-knowledge segments, each of which fits the maximum length requirement. This manner helps the stance detection model capture the entire background knowledge instead of the cropped knowledge with missing information, as in the previous knowledge enhancement paradigms. The sub-knowledge segment $\mathcal{K}_\mathrm{sub}$ is sampled from the collaborative knowledge $\mathcal{K}$ as follows:

$$\mathcal{K}_\mathrm{sub} = [\mathcal{K}_{i \cdot l/2}, \mathcal{K}_{(i+1) \cdot l/2}],$$

$$\text{s.t.,} \quad i \in (0, 1, \ldots, \lfloor \mathrm{Len}(\mathcal{K})/l \rfloor).$$

The collaborative adaptor and knowledge augmentation work together to optimize efficient parameter learning by reducing trainable parameters and addressing data limitations in low-resource stance detection tasks. The collaborative adaptor reduces the data reliance in training large-scale PLMs, while knowledge augmentation expands the training data to further improve training efficiency.

3.3 Staged optimization algorithm

To address the challenges of data discrepancy and domain gap between training data and pretrained models in the low-resource stance detection task, we propose a staged optimization algorithm that combines collaborative knowledge infusion and efficient parameter learning. However, the collaborative adaptor weights are initialized randomly, which may lead

unstable training with the pretrained backbone PLMs. Another issue is the unbalanced data distribution, which is often been overlooked in stance detection tasks. Our algorithm aims to mitigate these challenges in the low-resource stance detection task.

Algorithm 1 presents the details of staged optimization for the low-resource stance detection task. Once the collaborative knowledge $\mathcal{K}$ and adaptor initialization are prepared, we set the first stage for label smooth training and set the second stage with the weighted loss for unbalanced stance categories. Before training begins, we prepare the prompt $\mathcal{P}_{\mathrm{t}}$ and augmented knowledge $\mathcal{K}_{\mathrm{sub}}$ with the given stance target $\mathcal{T}$ as the training input triplet $\langle C, \mathcal{P}_{\mathrm{t}}, \mathcal{K}_{\mathrm{sub}}\rangle$. Then, we can get the output Y by feed forward computing $\mathrm{model}(C, \mathcal{P}_{\mathrm{t}}, \mathcal{K}_{\mathrm{sub}}|W, W_*)$ and stance prediction Y_{c} by $\mathrm{Softmax}\left(Y\right)$. The objective is defined by Cross-Entropy (CE),

$$\mathrm{CE}\left(Y, \tilde{Y}\right) = -\sum_{c=1}^{N_c} \theta \cdot Y_{\mathrm{c}} \log\left(\tilde{Y}_{\mathrm{c}}\right) \tag{10}$$

where N_c denotes the total number of targets. Our algorithm aims to enhance the model's capability to handle ambiguous and diverse inputs by adjusting the loss function weights (θ_{a}, θ_{b}) and introducing label smoothing (ϵ) during the training process. In the first stage, label smoothing is applied to soften the training targets, allowing the model to better handle uncertain data instances. Meanwhile, label smoothing helps the collaborative adapter convergence with newly initialed parameters. In the second stage, a weighted loss function is employed to assign different weights to different classes, thereby improving the model's ability to handle unbalanced datasets. Our algorithm provides a promising solution for tackling the challenges of data discrepancy and unbalanced data distribution in low-resource stance tasks.

4 Experiment

To evaluate the effectiveness of our proposed method, we conduct extensive experiments on three publicly available stance detection datasets and different low-resource settings. In this section, we provide a brief description of the datasets and the compared methods in our stance detection task. Finally, we summarize the results using F1-score as the default evaluation metric.

4.1 Datasets

(1) VAST[11] is a typical zero-shot and few-shot stance detection dataset that covers a wide range of over 6000 targets across various themes, including politics, sports, education, immigration, and public health, etc. The VAST dataset consists of 13 447, 2062, and 3006 examples in its training, validation, and test sets, respectively. Notably, the majority of targets in VAST are designed for zero-shot setting. It has an average of approximately 2.4 examples per target. This characteristic makes VAST particularly suitable for zero-shot and few-shot stance detection tasks.

(2) P-Stance[33] is stance detection dataset specific to the political domain. It contains in-target and cross-target settings with 21 574 labeled tweets on three specific targets: 'Biden', 'Sanders', and 'Trump'. In the in-target setting, the target and classifier are the same in both the training and evaluation sets. Conversely, in the cross-target setting, the targets are entirely different, allowing for the evaluation of the generalization performance.

(3) COVID-19-Stance[34] is stance detection dataset constructed from COVID-19-related tweets. It contains 6133 tweets with respect to four specific targets: 'Anthony S. Fauci, M. D. (Fauci)', 'Keep School Closed (School)', 'Stay at Home Order (Home)', and 'Wearing a Face Mask (Mask)'. COVID-19-Stance is an unbalanced dataset in terms of class distribution.

4.2 Compared methods

(1) TAN[35] is a classical attention-based method for the stance detection task. It contains a target-specific attention extractor and a long short-term memory network.

(2) BERT[36] is a well-known Transformer-based pretrained language model widely been used for various downstream tasks. We employ BERT as our baseline for reference in the stance detection task.

(3) WS-BERT-Dual[4] infuses target-related knowledge from extra Wikipedia to enhance background knowledge of PLMs in stance detection tasks. It utilizes two pretrained encoders to encode tweets and knowledge separately.

In addition to the shared baselines mentioned above, we also introduce other strong specific baselines for different stance datasets. For VAST task, we introduce graph convolution networks-based methods BERT-GCN, CKE-NET[9], and BSRGCN[28]. Those methods join large pretrained models with graphic convolution networks to leverage the learning capability of the models on heterogeneous data with structured graphic

representation. For the zero-shot setting in VAST task, we select BSRGCN and contrastive learning-based Joint-CL[30] as the compared methods.

For PStance task, we choose the bi-recurrent neural networks BiCE[37], and gated convolutional neural networks GCAE[38] and PGCNN[39] as the baselines. Specifically, the GCAE uses a Tanh (·) as the gate function to selectively output the sentiment futures according to the given aspect. Similarly, PGCNN also uses the parameterized filters as the gate function to effectively capture the aspect-specific features. Moreover, we also include BERT-Tweet[33] as the compared method, which is pretrained on the target Tweet domain data.

For COVID-19-Stance task, we choose CT-BERT[34] as the baseline, which is pretrained on the COVID-19-related tweet corpus to enhance task-specific domain knowledge. We also include CT-BERT-NS and CT-BERT-DAN[40], which incorporate self-training and domain adaption into CT-BERT to further improve model representation capability and reduce domain gap in the stance detection task.

4.3 Implementation details

We utilize RoBERTa as the backbone model for both VASE and PStance tasks. The batch size is set to 16, and the learning rates range from 1×10^{-5} to 5×10^{-5} in our experiments. Since COVID-19-Stance is a task related to the COVID-19 pandemic, we employ CT-BERT and BERT as the backbone models to leverage COVID-19-related knowledge from pretrained models. Due to our GPU memory limitations, the batch size is set to 8. Regarding the hyperparameters of the collaborative adaptor, we set the rank $r = 8$ for the low-rank adaptor. The prefix-taken is set to 100 with a dropout rate 0.2. The reducing factor for the feature fusion module is 16, and all gates are the ReLU function. The models are implemented using Pytorch, and the maximum input length is default set to 512 tokens. We train the models for a maximum of 30 epochs, and apply stopping with a patience of 10 epochs. The optimizer is AdamW with a weight decay of 1×10^{-5}. All the experiments are conducted with the same random seed on four NVIDIA RTX A5000 GPU cards.

4.4 Results

To verify the effectiveness of our proposed method, we evaluate its performance on three public stance detection datasets: VAST, PStance, and COVID-19-Stance. Firstly, we evaluate the proposed method on the VAST, a low-resource dataset with a significantly larger number of targets than the other two datasets. Additionally, we evaluate the method's performance on the PStance and COVID-19-Stance datasets. Following previous work in Ref. [4], all the datasets are evaluated using the macro-average F1-score as the standard metric. The overall performance is calculated as the average across all stances.

VAST dataset officially splits into two sub-tasks: zero-shot stance detection (600 targets) and few-shot stance detection (159 targets). Zero-shot setting does not include any targets in its training set, while the few-shot setting has very limited training samples (approximately 14.8 examples per target) in its training set. In contrast, the PStance and COVID-19-Stance datasets have over hundreds of training examples per target. Table 1 summarizes the evaluation results on the VAST dataset.

From the numbers presented in Table 1, we can observe that the baseline method BERT achieves clear improvements around 2% in the overall performance compared to the none pretrained baseline TAN, which indicates pretrained models have a stronger feature representation capability than none pretrained TAN. Building upon BERT, BERT-GCN incorporates Graphic Convolution Networks (GCNs) with BERT further improving the overall F1-score to 0.692. Similarly, the GCN-based methods CKE-Net and BSRGCN demonstrate progressive improvements by leveraging graph convolution networks, achieving an F1-score of 0.713. Specifically, BSRGCN performs better in the zero-shot setting, benefiting from the unsupervised training on the domain-specific corpus. Joint-CL further enhances model performance through contrastive learning, achieving an overall F1-score of

Table 1 Values of F1-score of zero-shot and few-shot on VAST.

Method	Zero-shot	Few-shot	Average
TAN	0.666	0.663	0.665
BERT	0.685	0.684	0.684
BERT-GCN	0.686	0.697	0.692
CKE-Net	0.702	0.701	0.701
BSRGCN	0.726	0.702	0.713
Joint-CL	0.723	0.716	0.723
WS-BERT-Dual	0.753	0.736	0.745
Ours	**0.819**	**0.796**	**0.807**

0.723.

However, all those solutions overlook the fact that VAST is a low-resource task, particularly for large pretrained models. Our method addresses this issue by incorporating efficient parameter learning and staged optimization for the low-resource task. Another neglect point in those solutions is that the infused target's knowledge should be the corrected knowledge. Our method addresses this issue by incorporating the collaborative knowledge infusion that introduces knowledge from multiple knowledge sources in a more accurate way. As a result, our method achieves the state-of-the-art performance on VAST, achieving an overall F1-score of 0.807. Interestingly, we find that the zero-shot settings achieve higher scores than the few-shot settings, especially in the pretrain-based methods. This difference can be attributed to the fact that the zero-shot and few-shot sets are two distinct subsets with completely different targets in the test set. Consequently, we can treat these two settings as two separate datasets.

Different from VAST with a large number of targets in a low-resource setting, PStance contains only 3 targets, and COVID-19-Stance contains 4 targets. Table 2 presents the evaluation results of compared methods on PStance. The classical recurrent neural network based TAN and BiCE obtain comparable performance to the GCN-based PGCNN and GCAE, yielding an F1-score around 0.75–0.76. Those results approach the performance of pretrained BERT. This similarity in performances suggests that rich training sources can benefit different types of models. Different from BERT which is pretrained on the generic corpus, BERT-Tweet enhances domain-specific knowledge by being pretrained on the Twitter corpus, resulting in a significant improvement of 4% in the overall F1-score. Building upon BERT-Tweet, WS-BERT-Dual further infuses the target background knowledge from Wikipedia, attaining an overall F1-score of 0.828. In light of WS-BERT-Dual, our method further optimizes knowledge augmentation through parameter-efficient learning, achieving the best performance across all the targets.

We also conduct evaluations of the proposed method on the domain specific COVID stance detection and present the results in Table 3. From the comparison of the results, we can clearly observe the performance gap between traditional gated-based methods (TAN, ATGRU, and GCAE) and pretrain-based models (CT-BERT and its variants). The gated-based methods, which only conduct finetuning on its rich training set, obtain low average F1-scores below 0.602, lacking any background knowledge specific to the target domain.

In contrast, pretrained models been trained on the COVID-related Twitters data exhibit good background knowledge and feature representation for COVID-19-Stance, resulting in a high average F1-score above 0.79. Meanwhile, with the help of self-training and domain adaptation techniques, CT-BERT has performance improvements in stance detection for 'Fauci' and 'Mask', but no substantial improvement in the overall F1-score. With the help of dual pretrained model encoders, WS-BERT-Dual further elevates the overall performance to 0.844. Similarly, our method achieves the best performance among the compared methods in COVID-19-Stance by leveraging collaborative adaptor and staged optimization. Based on the extensive experimental comparisons, we can conclude that our proposed method performs well not only in low-resource VAST stance detection task but also in rich-resource PStance and COVID-19-Stance tasks.

5 Discussion

5.1 Ablation study

We perform an ablation study on the main modules of

Table 2 F1-score on PStance.

Method	Trump	Biden	Sander	Average
TAN	0.771	0.776	0.716	0.751
BiCE	0.772	0.777	0.712	0.754
PGCNN	0.769	0.766	0.721	0.752
GCAE	0.790	0.780	0.718	0.763
BERT	0.783	0.787	0.725	0.765
BERT-Tweet	0.825	0.810	0.781	0.805
WS-BERT-Dual	0.858	0.835	0.790	0.828
Ours	**0.862**	**0.841**	**0.805**	**0.836**

Table 3 F1-score on COVID-19-Stance.

Method	Fauci	Home	Mask	School	Average
TAN	0.547	0.536	0.546	0.534	0.541
ATRGU	0.612	0.521	0.599	0.527	0.565
GCAE	0.640	0.645	0.633	0.490	0.602
CT-BERT	0.818	0.800	0.803	0.753	0.798
CT-BERT-NS	0.821	0.784	0.833	0.753	0.798
CT-BERT-DAN	0.832	0.787	0.825	0.717	0.790
WS-BERT-Dual	0.836	0.850	0.866	0.822	0.844
Ours	**0.8605**	**0.8676**	**0.8691**	**0.8333**	**0.8576**

the proposed method, namely collaborative knowledge infusion, efficient parameter learning, and staged optimization, using the VAST and PStance datasets. In addition to reporting the overall F1-score for zero-shot and few-shot settings in VAST, we also provide the detailed results for three specific stances: positive, negative, and neutral. For PStance, we report the average performance across the different targets.

Table 4 presents the results of the ablation study conducted on VAST, where RoBERTa serves as the backbone model. We study the impact of each module on the backbone performance. All individual modules that work with the backbone outperform the finetuning of the vanilla backbone. The collaborative Knowledge Infusion (KI) module, which includes knowledge verification and augmentation, facilitates the learning of target-specific background knowledge, and achieves remarkable improvements to 0.751. Likewise, the Efficient Parameter learning module (EP) proves beneficial for the stance detection model on the low-resource VAST data with the help of collaborative adaptors. When comparing the performance across different stances, we observe that the neutral stance exhibits significantly higher scores compared to the positive and negative stances.

Our Staged Optimization (SO) module tries to address this bias by incorporating label smooth and weighted loss, resulting in overall performance improvements. Furthermore, extensive ablation studies are conducted to assess different combinations of the modules. We can observe that two module combinations further improve model performance by 1%–2%. Similarly, our method incorporating all three modules achieves the best overall F1-score of 0.807 on VAST, highlighting the effectiveness of each module in addressing the challenges of the low-resource VAST task.

We also conduct an ablation study on PStance, which benefits from a relatively rich training source than the low-resource VAST dataset. Table 5 presents a summary of the ablation study on PStance, focusing on the efficiency of three modules with the same backbone RoBERTa. Notably, the ablation study results differ from the results obtained for VAST. Interestingly, we observe a slight decrease in model performance with the collaborative knowledge infusion module than the vanilla backbone. This performance drop may be attributed to the specific dataset, as PStance only contains four targets compared to diverse targets in VAST. In other words, the extensive sequential knowledge of these targets may impede the model's ability to learn features from shorter and raw sequences.

In the single-modular settings, we find backbone incorporating knowledge infusion or staged optimization exhibits superior performance when they are compared with the efficient parameter learning module in the rich-source PStance task. This suggests that full parameter finetuning is more effective than the adaptor-based solution in data-rich tasks. Similarly, in the two-modular settings, we observe that the setting with EP module (Setting 4 and 6) performs worse than the setting without EP modular (Setting 5). We also observe that the backbone with two-modular settings yields more improvements in the overall F1-score than the single-modular settings. In the three-modular settings, we find the performances of the 'Trump' and 'Sanders' targets could be further improved compared to the two-modular settings. However, the 'Biden' category experiences a significant drop compared to its results in Setting 6. We attribute this to the negative impact of the EP module in the three-modular setting, which slightly decreases the overall performance compares to the two-modular Setting 6. Thus, we can conclude that the adaptor-based solution does not always perform well in rich-resource tasks.

Table 4 Ablation study on VAST with F1-score.

Module			Zero-Shot setting (ZS)				Few-Shot setting (FS)				Average
KI	EP	SO	Negative	Positive	Neutral	Average	Negative	Positive	Neutral	Average	
–	–	–	0.657	0.590	0.950	0.733	0.656	0.605	0.974	0.745	0.739
✓	–	–	0.664	0.649	0.935	0.749	0.664	0.632	0.963	0.753	0.751
–	✓	–	0.714	0.723	0.907	0.781	0.677	0.686	0.885	0.749	0.765
–	–	✓	0.745	0.720	0.920	0.795	0.692	0.678	0.870	0.747	0.770
✓	✓	–	0.707	0.704	0.955	0.789	0.689	0.701	0.959	0.783	0.786
–	✓	✓	0.691	0.735	0.949	0.792	0.685	0.728	0.967	0.793	0.793
✓	–	✓	0.752	0.753	0.951	**0.819**	0.715	0.726	0.947	**0.796**	**0.807**

Table 5 Ablation study on PStance with F1-score.

Setting	Module			Target			Average
	KI	EP	SO	Trump	Biden	Sander	
0	–	–	–	0.855	0.827	0.769	0.817
1	✓	–	–	0.846	0.827	0.769	0.814
2	–	✓	–	0.763	0.805	0.785	0.784
3	–	–	✓	0.844	0.836	0.784	0.821
4	✓	✓	–	0.848	0.827	0.737	0.804
5	–	✓	✓	0.861	0.833	0.805	0.833
6	✓	–	✓	0.850	**0.855**	**0.805**	**0.837**
7	✓	✓	✓	**0.862**	0.841	**0.805**	0.836

5.2 Cross-target stance detection

We evaluate the model's generalization performance by cross-target stance detection, training model on the rich-source PStance, that trains the stance model on one target, and evaluate it on another target (e.g., training on Trump and testing on Biden). We employ BERT-Tweet[33] and WS-BERT-Dual[4] as the strong baselines. BERT-Tweet is pretrained on the Twitter corpus, benefiting from domain-specific knowledge. WS-BERT-Dual is a dual-path architecture using BERT and BERT-Tweet as feature encoders to incorporate target-specific Wikipedia knowledge. We follow the experimental settings of WS-BERT-Dual, testing on three targets: Trump, Biden, and Sanders.

From the results of Table 6, we can observe that the WS-BERT-Dual achieves significant improvements (average 5% F1-score) in all six cross-target pairs compared to the baseline BERT-Tweet, benefiting from the additional BERT branch for encoding extra Wikipedia knowledge. In contrast, our method only uses the RoBERTa as the backbone for cross-target stance detection. We further improve four of six cross-target pairs by introducing the staged optimization. Additionally, we notice that the two pairs' target results are not symmetric to each other. Overall, our proposed solution achieves the state-of-the-art performance in cross-target stance detection on the PStance dataset.

Table 6 Cross-target stance detection with F1-score.

Cross-target	BERT-Tweet	WS-BERT-Dual	Ours
Trump→Biden	0.589	**0.683**	0.682
Trump→Sander	0.565	0.644	**0.678**
Biden→Trump	0.636	0.677	**0.721**
Biden→Sander	0.670	0.690	**0.748**
Sander→Trump	0.587	0.**636**	0.634
Sander→Biden	0.730	0.768	**0.789**
Average	0.630	0.683	**0.709**

5.3 Efficient parameter learning

In this section, we compare the performance of our efficient parameter learning paradigm with the full model finetuning paradigm in the low-resource setting. We select the classic BERT and RoBERTa for full model finetuning, using both basic (B) and large (L) model sizes. Our efficient parameter learning approach utilizes the large-size RoBERTa as the backbone. All the models are evaluated in the zero-shot and few-shot settings as defined by VAST dataset. Note, the few-shot and zero-shot in traditionally computer vision tasks are evaluated with same test data. However, in our VAST NLP dataset, the few-shot and zero-shot settings are evaluated with different test data.

Table 7 presents the performance of different models on the zero-shot stance detection task with respect to three stances: positive, negative, and neutral. In the full model finetuning setting, we observe that the basic-sized models outperform the large-sized models in terms of average F1-score. Specifically, the BERT models experience a 1% drop in average F1-score from basic-size to large-size, while the RoBERTa models even encounter a decrease in F1-score of more than 4%. The large model's zero-shot performance decay may attribute to combined training few-shot on the low-resource VAST dataset. The large-size models exhibit reduced generalization capability with limited training data samples, resulting in performance decay in zero-shot stance detection tasks. Consistent with the findings of the ablation study, the neutral stance achieves significantly higher F1-score (above 0.9) than the positive and negative stances in the zero-shot setting. In contrast, our efficient parameter learning method, which maintains the pretrained model's generalization capability by freezing its parameters, achieves the best performance in the zero-shot stance detection task.

Table 8 presents the results of different models in the few-shot VAST stance detection task. Similar to the zero-shot setting, we observe a decline in overall performance as the model size increases. Additionally, the neutral stance detection performance exhibits significant superiority over the positive and negative stances. For the full model finetuning paradigm, there have been no notable variations between basic and large-size models in the positive and neutral stances. However, in the negative stance, the large-size models experience a sharp performance drop.

Table 7 Zero-shot F1-score on VAST.

Method	Number of parameters	Positive	Negative	Neutral	Average
BERT-B	1.1×10^8	0.640	0.632	0.942	0.738
BERT-L	3.4×10^8	0.600	0.668	0.941	0.724
RoBERTa-B	1.1×10^8	0.674	0.723	0.937	0.778
RoBERTa-L	3.4×10^8	0.657	0.590	0.950	0.733
Ours	3×10^3	**0.752**	**0.753**	**0.951**	**0.819**

Table 8 Few-shot F1-score on VAST.

Method	Number of parameters	Positive	Negative	Neutral	Average
BERT-B	1.1×10^8	0.642	0.651	0.951	0.748
BERT-L	3.4×10^8	0.642	0.613	0.918	0.738
RoBERTa-B	1.1×10^8	0.646	0.708	0.951	0.768
RoBERTa-L	3.4×10^8	0.656	0.605	0.974	0.745
Ours	3×10^3	**0.715**	**0.726**	**0.947**	**0.796**

The performance drop can be attributed to the unbalanced data distribution, which leads to model performance decay as training on the limited samples. Therefore, the full model finetuning is not an optional solution for zero-shot and few-shot stance detection tasks. The reason for zero-shot setting achieveing better performance than the few-shot setting mainly depends on the special test data setting in the VAST dataset that the few-shot and zero-shot settings have totally different test data samples. This setting is quite different from traditional few-shot and zero-shot settings with the same test data. In other words, the settings in Tables 7 and 8 can be seen as two datasets. This divergence from traditional evaluation methods is a key in understanding the experimental results. Moreover, this pattern of zero-shot settings outperforming few-shot settings is not unique to our study but is also observed in comparable works, such as TAN[35], BERT[36], CKE-NET[9], BSRGCN[8], and WS-BERT-Dual[4], which report the similar anomalies in Table 1. Our proposed method addresses the challenge of low-resource stance detection with an efficient solution, requiring only 3000 trainable parameters. This represents a significant reduction compared to the millions of parameters required for full model finetuning. Despite the parameter reduction, our method still can achieve superior performance with a considerable margin.

5.4 Collaborative adaptor analysis

Collaborative adaptor is an essential part of efficient parameter learning in the low-resource detection task, which consists of three modules: gated LoRA, prefix-tuning, and attentive fusion. To assess the importance of each module, we conduct evaluations by removing individual modules from our collaborative adaptor.

Table 9 presents the performance of different settings of the collaborative adaptor. For instance, the setting "without gated LoRA" indicates the removal of the "gated LoRA" module. All settings are evaluated on the low-resource VAST using the same hyper-parameters. We observe the "gated LoRA" and "prefix-tuning" modules exhibit similar drops of approximately 1% with slight variations across different stances. Surprisingly, in the "without attentive fusion" setting, all stance scores experience a sharp decline of approximately 7% on average. Through the performance comparisons, we discover that the attentive fusion module has more significant impact on down-steam stance detection tasks than the gated LoRA and Prefix-tuning modules in the efficient parameter learning paradigm. One possible explanation for this observation is that attentive fusion is more closely connected to the stance prediction classifier than the other two models, which serve as feature extractors with less impact on the final stance prediction.

5.5 Low-resource stance detection

We further evaluate our method's performance in the

Table 9 Collaborative adaptor analysis with F1-score.

Setting	Positive	Negative	Neutral	Average
without gated LoRA	0.719	0.703	0.958	0.793
without prefix-tuning	0.706	0.733	0.943	0.794
without attentive fusion	0.661	0.621	0.925	0.736
Ours	**0.734**	**0.739**	**0.949**	**0.807**

low-resource settings of stance detection on the subsets of PStance and COVID-19-Stance datasets. Specifically, we randomly sample 5%, 10%, 15%, and 20% data from their training sets as our low-resource data and keep its test sets for evaluation.

Following the setting of the ablation study, we set the BERT as the baseline and backbone of our method in all the low-resource stance detection evaluations. Tables 10 and 11 summarize the comparison results on different low-resource settings of PStance and COVID-19-Stance.

From results of Table 10 on different low-resource settings, we can observe that our method surpasses the baselines in all low-resource data settings, which shows our method presents effectiveness. Overview the whole performance across different settings, we can obviously find the progressive increasing trends with the training data adding in both the baseline and our method. Last but not least, our method benefits more with less training data, and the average improvements are reduced with more data introduced from 5% to 20%.

Table 11 summarizes the low-resource setting evaluation results on the COVID-19-Stance dataset. Similar to the low-resource setting results of PStance, our method surpasses the baseline with large margins in all low-resource settings. The average performance can achieve more than 6%. Different from the performance trends in PStance, the performance improvement trends do not keep consistently changing with increasing of training data. We think the main reason is the domain gap and diverse data distributions in different stance topics of COVID-19-Stance.

5.6 Comparison with ChatGPT 3.5

ChatGPT attracts lots of attention in the natural language processing community due to its impressive performance on conversational tasks, leading to its utilization in various downstream NLP tasks. In this section, we aim to evaluate the performance of ChatGPT 3.5 on VAST, which is a varied stance topics dataset with over a thousand targets. To adapt ChatGPT for the stance detection task, we construct the prompt as follows: "Please choose one stance from 'negative, positive, neutral' for ⟨TARGET⟩($\mathcal{T}$) on following content: ⟨TWEET⟩(C)?". We sequentially select and evaluate 100 samples, comprising 33 negative, 33 positive, and 34 neutral instances. Regarding ChatGPT is an evolving system (the ChatGPT results are evaluated on 04 June 2023), all the evaluation results are reported as follows.

Table 12 summarizes the evaluation results, with rows representing the target labels and columns indicating the model predictions. The results reveal that ChatGPT's performance on VAST stance detection dataset is not as impressive as anticipated, which is similar to the findings of the work directly using the chain-of-thought[41] in ChatGPT for stance detection on VAST dataset with only 0.623 of F1-score. From the result analysis, we observe that ChatGPT often predicts the positive or negative stances to the neutral stance, resulting in the neutral stance in a low recall score.

Table 10 Low-resource stance detection on PStance with F1-score.

Setting	Baseline			Ours			Average
	Trump	Biden	Sanders	Trump	Biden	Sanders	
5%	0.693	0.705	0.635	0.712	0.739	0.682	3.4% ↑
10%	0.705	0.724	0.710	0.738	0.750	0.725	2.5% ↑
15%	0.729	0.761	0.723	0.742	0.781	0.739	1.7% ↑
20%	0.740	0.784	0.745	0.769	0.794	0.751	1.7% ↑

Note: "↑" denotes the improvements of our method over the baseline.

Table 11 Low-resource stance detection on COVID-19-Stance with F1-score.

Setting	Baseline				Ours				Average
	Fauci	Home	Mask	School	Fauci	Home	Mask	School	
5%	0.346	0.440	0.370	0.227	0.408	0.440	0.370	0.227	6.6% ↑
10%	0.578	0.599	0.423	0.263	0.729	0.599	0.423	0.263	12.7% ↑
15%	0.696	0.621	0.490	0.403	0.772	0.621	0.490	0.403	6.4% ↑
20%	0.708	0.742	0.540	0.454	0.806	0.742	0.540	0.454	12.8% ↑

Note: "↑" denotes the improvements of our method over the baseline.

Table 12 Vast samples evaluation on ChatGPT 3.5.

Target	Prediction			Recall (%)
	Negative	Positive	Neutral	
Negative	18	3	12	54.5
Positive	4	18	11	54.5
Neutral	9	14	11	32.4
Precision (%)	58.1	51.4	33.3	–

This tendency might stem from ChatGPT's inclination to produce mild and friendly responses[42], leading to a bias toward predicting neutral stances. Furthermore, we observe that ChatGPT trends to output a neutral stance for sensitive topics, such as voting, humanity, and elections.

6 Conclusion

In this paper, we propose a method for low-resource stance detection that collaborative infuses verified target knowledge with efficient parameter learning. Firstly, we enhance the infusion of target-related knowledge by extending it beyond structured Wikipedia to encompass a broader range of unstructured information from the entire Internet. To ensure the selection of relevant semantic background knowledge, a knowledge verifier is employed. Secondly, we introduce efficient parameter learning through collaborative adaptors, which involve a minimal number of trainable parameters by freezing the weights of large PLM-based models. This manner not only facilitates efficient model training in low-resource stance detection tasks, but also retains the rich prior knowledge encoded in pretrained models. Thirdly, a staged optimization algorithm is proposed to mitigate the impact of unbalanced data. Additionally, knowledge augmentation and prompting techniques are integrated into our efficient parameter learning framework for low-resource stance detection. Experimental results demonstrate the effectiveness of our method on three public datasets with state-of-the-art performance. In future work, we plan to further explore efficient parameter learning in the context of multi-modal stance detection tasks.

Acknowledgment

This work was supported by the RCA founding of A*STAR and DSO National Laboratory (Nos. 2208-526-RCA-CFAR and SC23/22-3204FA). We thank for the constructive comments from Chieu Hai Leong, Chong Wen Haw, and Yap Yong Keong at DSO National Laboratory, Singapore.

References

[1] S. Ghosh, P. Singhania, S. Singh, K. Rudra, and S. Ghosh, Stance detection in web and social media: A comparative study, in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, F. Crestani, M. Braschler, J. Savoy, A. Rauber, H. Müller, D. E. Losada, G. H. Bürki, L. Cappellato, and N. Ferro, eds. Cham, Switzerland: Springer, 2019, pp. 75–87.

[2] A. Sen, M. Sinha, S. Mannarswamy, and S. Roy, Stance classification of multi-perspective consumer health information, in *Proc. ACM India Joint Int. Conf. Data Science and Management of Data*, Goa, India, 2018, pp. 273–281.

[3] K. Kawintiranon and L. Singh, Knowledge enhanced masked language model for stance detection, in *Proc. 2021 Conf. North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, Seattle, United States, 2021, pp. 4725–4735.

[4] Z. He, N. Mokhberian, and K. Lerman, Infusing knowledge from Wikipedia to enhance stance detection, in *Proc. 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, Dublin, Ireland, 2022, pp. 71–77.

[5] R. Liu, Z. Lin, Y. Tan, and W. Wang, Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph, in *Proc. Findings of the Association for Computational Linguistics*: *ACL-IJCNLP 2021*, Virtual Event, 2021, pp. 3152–3157.

[6] O. Agarwal, H. Ge, S. Shakeri, and R. Al-Rfou, Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training, in *Proc. 2021 Conf. 9th American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*, Virtual Event, 2021, pp. 3554–3565.

[7] Y. Xu, C. Zhu, S. Wang, S. Sun, H. Cheng, X. Liu, J. Gao, P. He, M. Zeng, and X. Huang, Human parity on CommonsenseQA: Augmenting self-attention with external attention, in *Proc. 31st Int. Joint Conf. Artificial Intelligence*, Vienna, Austria, 2022, pp. 2762–2768.

[8] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, KEPLER: A unified model for knowledge embedding and pre-trained language representation, *Trans. Assoc. Comput. Linguist*., vol. 9, pp. 176–194, 2021.

[9] Y. Lin, Y. Meng, X. Sun, Q. Han, K. Kuang, J. Li, and F. Wu, BertGCN: Transductive text classification by combining GNN and BERT, in *Proc. Findings of the Association for Computational Linguistics*: *ACL-IJCNLP 2021*, Virtual Event, 2021, pp. 1456–1462.

[10] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in *Proc. 36th Int. Conf. Neural Information Processing Systems*, New Orleans, LA, USA, 2022, p. 1800.

[11] E. Allaway and K. McKeown, Zero-shot stance detection: A dataset and model using generalized topic

representations, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing* (*EMNLP*), Virtual Event, 2020, pp. 8913–8931.

[12] C. Zhu, Y. Xu, X. Ren, B. Y. Lin, M. Jiang, and W. Yu, Knowledge-augmented methods for natural language processing, in *Proc. 60th Annu. Meeting of the Association for Computational Linguistics*: *Tutorial Abstracts*, Dublin, Ireland, 2022, pp. 12–20.

[13] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, K-BERT: Enabling language representation with knowledge graph, in *Proc. 34th AAAI Conf. Artificial Intelligence*, New York, NY, USA, 2020, pp. 2901–2908.

[14] D. Yu, C. Zhu, Y. Yang, and M. Zeng, JAKET: Joint pre-training of knowledge graph and language understanding, in *Proc. 36th AAAI Conf. Artificial Intelligence*, Virtual Event, 2022, pp. 11630–11638.

[15] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, ERNIE: Enhanced language representation with informative entities, in *Proc. 57th Annu. Meeting of the Association for Computational Linguistics*, Florence, Italy, 2019, pp. 1441–1451.

[16] T. Févry, L. B. Soares, N. FitzGerald, E. Choi, and T. Kwiatkowski, Entities as experts: Sparse memory access with entity supervision, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing* (*EMNLP*), Florence, Italy, 2020, pp. 4937–4951.

[17] I. Beltagy, K. Lo, and A. Cohan, SciBERT: A pretrained language model for scientific text, in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, China, 2019, pp. 3615–3620.

[18] K. R. Kanakarajan, S. Ramamoorthy, V. Archana, S. Chatterjee, and M. Sankarasubbu, Saama research at MEDIQA 2019: Pre-trained BioBERT with attention visualisation for medical natural language inference, in *Proc. 18th BioNLP Workshop and Shared Task*, Florence, Italy, 2019, pp. 510–516.

[19] D. Q. Nguyen, T. Vu, and A. T. Nguyen, BERTweet: A pre-trained language model for English tweets, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing*: *System Demonstrations*, Virtual Event, 2020, pp. 9–14.

[20] V. Shwartz, P. West, R. Le Bras, C. Bhagavatula, and Y. Choi, Unsupervised commonsense question answering with self-talk, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing* (*EMNLP*), 2020, pp. 4615–4629.

[21] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. T. Yih, Dense passage retrieval for open-domain question answering, in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing* (*EMNLP*), Virtual Event, 2020, pp. 6769–6781.

[22] Y. Yao, S. Huang, L. Dong, F. Wei, H. Chen, and N. Zhang, Kformer: Knowledge injection in transformer feed-forward layers, in *Proc. 11th CCF Int. Conf.*, Guilin, China, 2022, pp. 131–143.

[23] D. Küçük and F. Can, Stance detection: A survey, *ACM Comput. Surv.*, vol. 53, no. 1, p. 12, 2020.

[24] A. ALDayel and W. Magdy, Stance detection on social media: State of the art and trends, *Inf. Process. Manage.*, vol. 58, no. 4, p. 102597, 2021.

[25] M. Hardalov, A. Arora, P. Nakov, and I. Augenstein, A survey on stance detection for mis- and disinformation identification, in *Proc. Findings of the Association for Computational Linguistics*: *NAACL 2022*, Seattle, DC, USA, 2022, pp. 1259–1277.

[26] M. Mohtarami, J. Glass, and P. Nakov, Contrastive language adaptation for cross-lingual stance detection, in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, China, 2019, pp. 4442–4452.

[27] E. Zotova, R. Agerri, M. Nuñez, and G. Rigau, Multilingual stance detection in tweets: The Catalonia independence corpus, in *Proc. 12th Language Resources and Evaluation Conf.*, Marseille, France, 2020, pp. 1368–1375.

[28] Y. Luo, Z. Liu, Y. Shi, S. Z. Li, and Y. Zhang, Exploiting sentiment and common sense for zero-shot stance detection, in *Proc. 29th Int. Conf. Computational Linguistics*, Gyeongju, Republic of Korea, 2022, pp. 7112–7123.

[29] R. Liu, Z. Lin, H. Ji, J. Li, P. Fu, and W. Wang, Target really matters: Target-aware contrastive learning and consistency regularization for few-shot stance detection, in *Proc. 29th Int. Conf. Computational Linguistics*, Gyeongju, Republic of Korea, 2022, pp. 6944–6954.

[30] B. Liang, Q. Zhu, X. Li, M. Yang, L. Gui, Y. He, and R. Xu, JointCL: A joint contrastive learning framework for zero-shot stance detection, in *Proc. 60th Annu. Meeting of the Association for Computational Linguistics* (*Volume 1*: *Long Papers*), Dublin, Ireland, 2022, pp. 81–91.

[31] N. Reimers and I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. Natural Language Processing* (*EMNLP-IJCNLP*), Hong Kong, China, 2019, pp. 3982–3992.

[32] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, LoRA: Low-rank adaptation of large language models, in *Proc. 10th Int. Conf. Learning Representations*, Virtual Event, https://openreview.net/forum?id=nZeVKeeFYf9, 2024.

[33] Y. Li, T. Sosea, A. Sawant, A. J. Nair, D. Inkpen, and C. Caragea, P-stance: A large dataset for stance detection in political domain, in *Proc. Findings of the Association for Computational Linguistics*: *ACL-IJCNLP 2021*, Virtual Event, 2021, pp. 2355–2365.

[34] K. Glandt, S. Khanal, Y. J. Li, D. Caragea, and C. Caragea, Stance detection in COVID-19 tweets, in *Proc. 59th Annu. Meeting of the Association for Computational Linguistics and the 11th Int. Joint Conf. Natural Language Processing* (*Volume 1*: *Long Papers*), Bangkok, Thailand, 2021, pp. 1596–1611.

[35] J. Du, R. Xu, Y. He, and L. Gui, Stance classification with target-specific neural attention, in *Proc. 26th Int. Joint*

Conf. Artificial Intelligence, Melbourne, Australia, 2017, pp. 3988–3994.

[36] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies, Volume 1* (*Long and Short Papers*), Minneapolis, MU, USA, 2019, pp. 4171–4186.

[37] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, Stance detection with bidirectional conditional encoding, in *Proc. 2016 Conf. Empirical Methods in Natural Language Processing*, Austin, TX, USA, 2016, pp. 876–885.

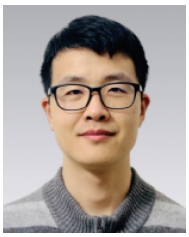
[38] W. Xue and T. Li, Aspect based sentiment analysis with gated convolutional networks, in *Proc. 56th Annu. Meeting of the Association for Computational Linguistics* (*Volume 1*: *Long Papers*), Melbourne, Australia, 2018, pp. 2514–2523.

[39] B. Huang and K. Carley, Parameterized convolutional neural networks for aspect level sentiment classification, in *Proc. 2018 Conf. Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 1091–1096.

[40] B. Zhang, M. Yang, X. Li, Y. Ye, X. Xu, and K. Dai, Enhancing cross-target stance detection with transferable semantic-emotion knowledge, in *Proc. 58th Annu. Meeting of the Association for Computational Linguistics*, Virtual Event, 2020, pp. 3188–3197.

[41] B. Zhang, X. Fu, D. Ding, H. Huang, Y. Li, and L. Jing, Investigating chain-of-thought with ChatGPT for stance detection on social media, arXiv preprint arXiv: 2304.03087, 2023.

[42] J. Wei, D. Huang, Y. Lu, D. Zhou, and Q. V. Le, Simple synthetic data reduces sycophancy in large language models, arXiv preprint arXiv: 2308.03958, 2023.



Ming Yan received the PhD degree from Sichuan University, China in 2019. He was a visiting PhD at Georgia State University (GSU), USA in 2018. He is currently a senior scientist at Centre for Frontier AI Research, and Institute of High-Performance Computing, Agency for Science Technology and Research, Singapore. He is the young associate editor in *Big Data Mining and Analytics*. He has published several papers on *ACL*, *IEEE TII*, *KBS*, *Neurocomputing*, etc. His research interests include medical image analysis, neural networks, and natural language processing.

Tianyi Zhou Joey received the PhD degree from Nanyang Technological University (NTU), Singapore in 2015. He is a principal scientist and an investigator at Centre for Frontier AI Research, and Institute of High-Performance Computing, Agency for Science Technology and Research, Singapore. He is an adjunct faculty at National University of Singapore. He was a senior research engineer at Sony US Research Center, San Jose, USA in 2017. His research focus is on how to enhance machine learning efficiency and robustness.

W. Tsang Ivor received the PhD degree in computer science from the Hong Kong University of Science and Technology, China in 2007. He is the director of Centre for Frontier AI Research, and a senior principal scientist at Institute of High-Performance Computing, Agency for Science Technology and Research, Singapore. He is also an adjunct professor at School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received the ARC Future Fellowship for his outstanding research on big data analytics and large-scale machine learning in 2013. In addition, he was conferred the IEEE fellow for his outstanding contributions to large-scale machine learning and transfer learning. His research focuses on transfer learning, deep generative models, learning with weakly supervision, and big data analytics for data with extremely high dimensions in features, samples, and labels.