Integrating Pattern- and Fact-based Fake News Detection via Model Preference Learning

Qiang Sheng, Xueyao Zhang
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
shengqiang18z@ict.ac.cn
zhangxueyao19s@ict.ac.cn

Juan Cao
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
caojuan@ict.ac.cn

Lei Zhong
Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
zhonglei18s@ict.ac.cn

ABSTRACT
To defend against fake news, researchers have developed various methods based on texts. These methods can be grouped as 1) pattern-based methods, which focus on shared patterns among fake news posts rather than the claim itself; and 2) fact-based methods, which retrieve from external sources to verify the claim’s veracity without considering patterns. The two groups of methods, which have different preferences of textual clues, actually play complementary roles in detecting fake news. However, few works consider their integration. In this paper, we study the problem of integrating pattern- and fact-based models into one framework via modeling their preference differences, i.e., making the pattern- and fact-based models focus on respective preferred parts in a post and mitigate interference from non-preferred parts as possible. To this end, we build a Preference-aware Fake News Detection Framework (PREFEND), which learns the respective preferences of pattern- and fact-based models for joint detection. We first design a heterogeneous dynamic graph convolutional network to generate the respective preference maps, and then use these maps to guide the joint learning of pattern- and fact-based models for final prediction. Experiments on two real-world datasets show that Pref-END effectively captures model preferences and improves the performance of models based on patterns, facts, or both.

CCS CONCEPTS
• Information systems → Data mining; • Computing methodologies → Natural language processing.

KEYWORDS
fake news detection, preference learning, graph neural networks, pattern mining, fact-checking

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Qiang Sheng, Xueyao Zhang, Juan Cao, and Lei Zhong. 2021. Integrating Pattern- and Fact-based Fake News Detection via Model Preference Learning. In Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM ’21), November 1–5, 2021, Virtual Event, QLD, Australia. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3459637.3482440

1 INTRODUCTION
Fake news that spreads on “online” social media continually causes “offline” real-world harms in crucial domains, such as politics [11], finance [25], and public security [36]. The most recent example is the COVID-19 infodemic [30] where thousands of fake news pieces spread through social media [52]. Under such severe circumstances, developing fake news detection systems has been critical for maintaining a trustful online news ecosystem.

To detect fake news on social media, researchers propose to extract hand-crafted features or deep-learning features [4] from contents, social contexts, propagation networks, etc. In this paper, we focus on the deep learning methods based on textual contents, which can be grouped as: 1) Pattern-based methods (e.g.,

Figure 1: A motivating example. Ideally, given the same news post, the pattern-based and the fact-based model have different preferences on textual clues to predict whether the post is fake. The post is translated into English.
Our contributions are summarized as follows:

- To the best of our knowledge, our work is the first that combines pattern- and fact-based fake news detection. We discuss their complementary roles in fake news detection and propose to consider their preferences for better integration.
- We propose a novel framework, Pref-FEND, which leverages a heterogeneous dynamic GCN to learn model preferences and effectively integrates them for fake news detection.
- Extensive experiments on two newly constructed datasets demonstrate the effectiveness of Pref-FEND on learning models’ preferences and improving the detection performance for both single-preference models and integrated models. The code and datasets are available at https://github.com/ICTMCG/Pref-FEND.

2 RELATED WORK

2.1 Fake News Detection

Fake news detection aims at automatically classifying a news piece as real or fake. Existing methods mostly capture features from contents (texts or and images) and social contexts that generate in the spread process, such as propagation networks, and external resources. Some works judge the veracity of a news piece more objectively, with references to pre-constructed encyclopedias. A more flexible way is to directly use articles retrieved by search engines as evidence to predict the news veracity. In the very first work on evaluating information credibility on social media, Castillo et al. [5] list a series of post-based features, including the length, whether the post contains exclamation or question marks, etc. Following this line, Volkova et al. [49] injects subjectivity, psycholinguistic, and moral foundations features into deep neural networks. Przybyla [35] focuses on writing styles. Some works attempt to differentiate the patterns across multiple topical categories. A recent trend of pattern-based methods is to refocus on the sentiment and emotional patterns, as the use of eye-catching terms in deceptive and fake posts may manipulate the readers’ emotions.

Pattern-based Fake News Detection. As fake news often contains opinionated and inflammatory language to attract readers, common patterns that are different from those in real news are shared across fake news pieces of different topics. In this paper, we focus on the text-based methods which can be grouped as:

Fact-based Fake News Detection. These methods judge the veracity of a news piece more objectively, with references to pre-constructed external resources such as knowledge graphs and online encyclopedias. A more flexible way is to directly use articles retrieved by search engines as evidence to predict the news veracity. Popat et al. [34] use post-specific attention to model the post-article interactions, while the following works consider text entailment, such as coherence and conflicts using the attention mechanism.

Note that the claims provided by the datasets for evaluation of fact-based methods is generally normalized by the human fact-checkers to be declarative and concise, so they are not suitable to evaluate the pattern-based ones. In this paper, we construct two new datasets (in English and Chinese) by referring to existing datasets and external sources for evaluation of pattern-and-fact-based methods. Different from the above methods, our work do not develop better pattern- or fact-based methods, but integrate the existing ones for comprehensively detecting fake news based on texts.
2.2 Graph Neural Networks for Text Mining

Due to its expressive power for integrating structural and semantic information, graph neural networks (GNNs) have been widely used for applications in text mining such as information extraction [17] and sentiment analysis [50]. Most works use homogeneous GNNs which treat nodes as the same type. Hu et al. [19] leverages a heterogeneous graph to aggregate multi-type neighbors and updates the correlation matrix every layer. Two readouts use the correlation matrix of the $L$-th layer to generate preference maps. Only parts of nodes and edges are shown. (c) and (d) exemplify how a pattern-based and a fact-based model works with the preference map, respectively. With the Maps, (c) and (d) attend to helpful tokens for capturing patterns or facts.

3 PROBLEM STATEMENT

Let $P$ be a news post on social media containing $n$ tokens. Let $D$ be the set of relevant articles of $P$. $D$ is retrieved from a fact-checking source $\mathcal{D}$. Following most existing works, we treat fake news detection as a binary classification problem. The ground-truth label $y$ is 1 if $P$ is fake, otherwise 0. We formulate the following tasks:

**Pattern-based Fake News Detection:** Given $P$, learn a function $f_P : f_P(P) \rightarrow \hat{y}$, such that it maximizes the predictive accuracy w.r.t. $y$.

**Fact-based Fake News Detection:** Given $P$, retrieve relevant articles $D$ from $\mathcal{D}$, learn a function $f_r : f_r(P, D) \rightarrow \hat{y}$, such that it maximizes the predictive accuracy w.r.t. $y$.

**Joint Pattern-and-Fact-based Fake News Detection:** Given $P$, $D$, a pattern-based model $f_P$ and a fact-based model $f_r$, learn a function $f : f(P, D, f_P, f_r) \rightarrow \hat{y}$, such that it maximizes the predictive accuracy w.r.t. $y$.

4 PROPOSED FRAMEWORK

Figure 2(a) overviews the architecture of the proposed Pref-FEND, whose goal is to learn the models’ preferences and employ them for better joint fake news detection. Given a post $P$, Pref-FEND
first respectively generates preference maps (i.e., token-level preference scores) for the pattern- and fact-based model with a heterogeneous dynamic GCN. Then, the preference maps are fed into the corresponding model along with $P$ to help the model focus on its preferred information. Finally, the models’ output features are fused to predict if $P$ is real or fake. Besides the normal classification loss, we design two auxiliary losses as enhancements, whose goals are to minimize the similarity between the two maps and to minimize the classification loss when the input maps are exchanged and ground-truth labels are reversed, respectively. (see Section 4.2)

### 4.1 Preference Map Generation

Assuming that $P$ has $n$ tokens, a preference map is a score distribution of length $n$ where the $i$-th score represents to what extent the $i$-th token is preferred by the corresponding fake news detection model. For the pattern- and the fact-based model, we respectively generate Pattern Preference Map and Fact Preference Map

$$m_P = [m_{P,i}]_{i=1}^n, m_F = [m_{F,i}]_{i=1}^n,$$

where all scores are in $[0, 1]$ and the sum of each map is 1.

#### 4.1.1 Stylistic Tokens & Entities Recognition

As illustrated in Section 1, a pattern-based model focuses on common patterns (generally, writing styles) while a fact-based one focuses on verifiable objective claims. To guide the map generation, we exploit the prior knowledge with reference to the existing pattern- and fact-based works. Specifically, we recognize tokens that are likely to represent writing styles or key objective elements. To indicate patterns, we recognize a set of stylistic tokens $S = \{s_1, \ldots, s_n\}$ (e.g., emotional words, pronouns, punctuation marks) [62]; and to indicate facts, we extract the entities $E = \{e_1, \ldots, e_n\}$ because a verifiable claim generally contains at least one entity [46]. These indicating tokens are derived using pre-constructed dictionaries and public tools. In detail, to recognize stylistic tokens, we follow [62], which summarizes diverse emotion-related features and other useful linguistic features to represent textual patterns, and then generate a stylistic token table for each dataset. The types and references are shown in Table 1. A simple exact matching is performed to recognize the stylistic tokens in texts. To recognize the entities, we use two public tools: LAC [20] for Chinese and TexSmart [26, 61] for English. The tokens excluded by $S$ and $E$ are in a set $T = \{t_1, \ldots, t_m\}$ where $n = n_s + n_e - n_t$.

#### 4.1.2 Heterogeneous Dynamic GCN

Although the stylistic tokens and entities recognized by general dictionaries or tools provide a good prior to what tokens might be preferred, directly using the recognition result for map generation is insufficient: First, the coverage is limited, leading the map to overlook some other preferred and useful tokens for detection models; Second, a token’s preference score should be dynamically determined in its context (i.e., the post) rather than static rules. To enable the information of different types of nodes to dynamically and sufficiently interact with each other, we design a graph-based preference learner, Heterogeneous Dynamic Graph Convolutional Network (HetDGCN). As shown in Figure 2(b), we first construct a heterogeneous graph that contains multi-type nodes (tokens) with a learnable correlation matrix (i.e., adjacent matrix). Then, we leverage a heterogeneous graph convolution to enable message passing among different types of nodes. The final preference scores are obtained using the learned correlation matrix. The stages are as follows:

**Graph Initialization.** Recall that we have divided the tokens in $P$ into three parts: stylistic tokens $S$, entities $E$, and others $T$. To preserve their different roles, we construct a heterogeneous graph $G$, where each node corresponds to a token in $S, E$, or $T$ and the weight of each edge represents the correlation between the connected tokens. The node representation is initialized with the pre-trained language model (here, BERT [8], denoted as $H^{(0)} \in \mathbb{R}^{d \times d}$ where $d$ is the dimensionality of each node vector. Note that this matrix is stacked with the representation of $S, E$, and $T$, i.e., $H^{(0)} = [H_S^{(0)}; H_E^{(0)}; H_T^{(0)}]$. The edge weights (correlations) are initialized with calculating the cosine similarity of token pairs [19] which is scaled to $[0, 1]$:}

$$A^{(0)}(i, j) = \frac{h_i^{(0)} \cdot h_j^{(0)}}{\|h_i^{(0)}\| \|h_j^{(0)}\|} + 0.5,$$

where $h_i^{(0)}$ and $h_j^{(0)}$ are the initial node features, and $A^{(0)}(i, j) \in [0, 1]$ is the initial weight of the edge connecting the $i$-th and the $j$-th node. Following [23], we define the normalized correlation matrix of the $l$-th layer $\hat{A}^{(l)} = \left((D_l)^{-\frac{1}{2}}A^{(l)}(D_l)^{-\frac{1}{2}}\right)$, where $D_l$ is the degree matrix of the $l$-th layer where $D_l(i, i) = \sum_j A^{(l)}(i, j)$. The degree weights (correlations) are updated each layer according to the present node similarity and expect the final correlations (edge weights) could reflect the bias of the nodes in the context. In detail, the feature matrix of $(l + 1)$-th layer is calculated with

$$H^{(l+1)} = \text{ReLU}\left(\sum_{t \in T} \hat{A}^{(l)} \cdot H^{(l)} \cdot W_T^{(l)}\right),$$

where $\hat{A}^{(l)}$ is the submatrix of the correlation matrix of the $l$-th layer $\hat{A}^{(l)}$ whose rows contain all the nodes and columns record their correlation with nodes of the type $t \in \{S, E, T\}$. $W_T^{(l)}$ is the
learnable weight matrix of the type $r$ in this layer. Then, the correlation matrix is updated using

$$\Delta A^{(l+1)} = \sigma \left( H^{(l+1)} W_A H^{(l+1)T} \right),$$  \hspace{1cm} (4)$$

$$A^{(l+1)} = A^{(l)} + (1 - \alpha) \Delta A^{(l+1)},$$  \hspace{1cm} (5)$$

where $W_A^{(l+1)}$ is the learnable weight matrix for updating correlations, $\sigma$ denotes the sigmoid function and $\alpha$ is a trade-off factor in $[0, 1]$.

**Preference-aware Readout.** After the $L$-layer HetDGCN, we obtain the correlation matrix $A^{(L)}$, on which we expect to estimate the preference levels to pattern- and fact-based models of each token. For the $i$-th node, the pattern preference score $m_{Pi}$ is calculated by its correlation with any nodes except those representing entity tokens:

$$m_{Pi} = \sum_{j=1}^{n} A^{(L)}(i, j) \cdot \sum_{k=1}^{n} A^{(L)}(i, k).$$  \hspace{1cm} (6)$$

Similarly, the fact preference score excludes the correlation with the stylistic nodes:

$$m_{Fi} = \sum_{j=1}^{n} A^{(L)}(i, j) \cdot \sum_{k=1}^{n} A^{(L)}(i, k).$$  \hspace{1cm} (7)$$

Finally, the preference maps are obtained by normalized the correlation sums of each token:

$$m_p = \frac{m_{Pi}}{\sum_{j=1}^{n} m_{Pj}}, m_f = \frac{m_{Fi}}{\sum_{j=1}^{n} m_{Fj}},$$  \hspace{1cm} (8)$$

### 4.2 Preference-aware Joint Fake News Detection

As the fact-based and pattern-based models are diverse, we here use the typical pattern- and fact-based detection process to illustrate how to integrate the generated preference maps into them. Most specific models can be easily reformulated similarly to accommodate our framework.

**4.2.1 Pattern-based Model.** As shown in Figure 2(c), a typical pattern-based uses a textual feature extractor to obtain a vector for final prediction. Here, we use the Pattern Preference Map as attention weights to make the model attend to its preferred tokens in the post $P$. For example, if the extractor is a BERT [8] or an LSTM [18] whose output is $[p_1; \ldots; p_n]$, the aggregated vector is calculated as

$$p = \frac{\sum_{i=1}^{n} m_{Pi} p_i}. \hspace{1cm} (9)$$

Note that our preference map is at the token level, for the extractor that does not output $n$ vectors such as TextCNN [22], the map might be used before the extractor, right after we obtain token embeddings from pre-trained models.

**4.2.2 Fact-based Model.** In a typical fact-based model, the post $P$ are first used to retrieve from a fact-checking source to collect the related articles (or evidence) $D$. Assuming $n_f$ articles are returned, we represent the articles in $D$ as $[d_1; \ldots; d_{nf}]$. Then the post and evidence vectors are fed into an inference module, which is often designed to capture the complicated interactions such as coherence and conflicts between $P$ and $D$ (e.g., [27, 56]). The output vectors of inference module $f$, which implicitly represent the relationship of the post-evidence pairs, is used for final prediction.

To avoid the interference of non-check-worthy parts (e.g., the publisher’s remark), the Fact Preference Map guides the inference module by using the attention mechanism to aggregate the token vectors in $P$ before post-evidence inference. The final vector is calculated as

$$q = \sum_{i=1}^{n} m_{Fi} q_i, \hspace{1cm} (10)$$

$$f = \text{InferenceModule}(q_i [d_1; \ldots; d_{nf}]), \hspace{1cm} (11)$$

where $q_i$ is the representation of the $i$-th token in $P$ for fact-based methods.

**4.2.3 Joint Detection.** For final prediction, we concatenate the output vectors of pattern- and fact-based models and feed it into a multi-layer Perceptron (MLP) and obtain the prediction $\hat{y}$:

$$\hat{y} = \text{MLP}(\{p; f\}). \hspace{1cm} (12)$$

**4.2.4 Losses.** During training, we use three losses to supervise 1) the prediction of binary (fake and real) classification; and 2) the differentiation of the two preference maps. For the first goal, we minimize the cross-entropy loss between the prediction $\hat{y}$ and the label $y$

$$L_{cls}(y, \hat{y}) = \text{CELoss}(y, \hat{y}) \hspace{1cm} (13)$$

where $\text{CELoss}(y, p) = -y \log p - (1 - y) \log (1 - p)$. For the second goal, we consider the reciprocal roles of the two models and let them supervise mutually. In detail, we minimize the cosine similarity between the Pattern and the Fact Preference Map

$$L_{cos} = \frac{m_p \cdot m_f}{\|m_p\| \|m_f\|}, \hspace{1cm} (14)$$

and the cross-entropy loss under the condition that the input maps for the two models are exchanged and the ground-truth label is reversed

$$L_{cls}(\hat{y_{rev}}, \hat{y'}) = \text{CELoss}(\hat{y_{rev}}, \hat{y'}) \hspace{1cm} (15)$$

where $\hat{y_{rev}} = [1 - y]$ and the predictive value $\hat{y'} = \text{MLP}(\{p'; f'\})$. $p'$ and $f'$ are respectively the output of the pattern-based and the fact-based model with each other’s preference map as input. When receiving non-preferred information, the models are expected to be misled and generate non-distinctive features. The total loss of a sample to minimize is

$$L = \beta_1 L_{cls}(y, \hat{y}) + \beta_2 L_{cos} + \beta_3 L_{cls}(\hat{y_{rev}}, \hat{y'}) \hspace{1cm} (16)$$

where $\beta_1, \beta_2$ and $\beta_3$ are trade-off factors in $[0, 1]$. We average the loss of samples in each mini-batch before backpropagation.

### 5 EXPERIMENTS

We conduct experiments on two datasets to answer the following evaluation questions:

**EQ1:** Can Pref-FEND improve the performance of fake news detection models with single preference?  

**EQ2:** Can Pref-FEND improve the performance for fake news detection that is integrated by pattern- and fact-based models?  

**EQ3:** How effective are the designed components of Pref-FEND?  

**EQ4:** How different are the Fact and the Pattern Preference Map?
Table 2: Statistics of the Weibo and the Twitter dataset.

| Number of     | Weibo       | Twitter      |
|---------------|-------------|--------------|
|               | Train       | Val  | Test | Train | Val | Test |
| Fake News     | 1,896 | 632   | 633  | 3,419 | 1,140 | 1,140 |
| Real News     | 1,920 | 640   | 641  | 5,406 | 1,802 | 1,802 |
| Total         | 3,816 | 1,272 | 1,274 | 8,825 | 2,942 | 2,942 |
| Relevant Articles | 17,849 |       |      | 12,419 |

5.1 Datasets

As no existing dataset of fake news detection provides social media posts and relevant articles (as the fact-checking source) simultaneously, we construct two datasets of different languages (Chinese and English) based on the existing data and external sources. The statistics are shown in Table 2. The details are as follows:

Weibo Dataset

*Post.* We utilize the Weibo-20 dataset [62] which contains 6,362 news posts and the ratio of fake and real news posts is roughly 1:1. We keep its original temporal split with a ratio of 6:2:2 for train, validation, and test set.

*Relevant Articles.* We collect fact-checking articles and other relevant articles to construct our fact-checking source. In detail, we use the fact-checking articles crawled in [39] from multiple websites such as Jiaozhen⁵, Zhuoyaoyi⁶, and Baidu Piyao⁷. Then, we crawl other relevant articles from Baidu News, with the keywords in the Weibo posts as queries. The keywords are extracted using jieba⁸. For each query, we obtain at most 30 items and attempt to download full articles using Newspaper 3k⁹. Finally, the de-duplication of all accessible articles lead to an article base containing 17,849 articles.

Twitter Dataset

*Post.* We first combine two datasets for detecting previously fact-checked claims released by Shaar et al. [38] and Vo and Lee [47], respectively, as they not only provide tweets but also relevant articles from Snopes⁵. As our task is formulated as a binary classification task, we merge true, mostly-true, correct-attribute into real, and false, mostly-false, misattributed, and legend into fake. The other categories are dropped. As these two datasets are largely imbalanced (1,047 real and 8,992 fake), we utilize PHEME [24] dataset as a supplement, whose annotation files provide some referred news links. For PHEME, we merge real and non-rumor into real and obtain 5,090 real and 638 fake news posts. After pre-processing using TexSmart and dropping failure cases, we obtain 14,709 posts.

*Relevant Articles.* Because the Twitter dataset has fewer topics than the Weibo dataset, we start from the articles in these datasets to construct the relevant article base. First, we incorporate the fact-checking articles from the datasets released in [38] and [47], and referred news articles in the PHEME dataset (if accessible). Then, we use their titles (tokenized using NLTK [3]) as queries and search on Google News using GNews⁹. After post-processing, we obtain an article base containing 12,419 articles. Note that we do not use the existing DeClarE [34] and MultiFC [2] datasets which provide both claims (posts) and relevant articles (or webpages) because its claims are normalized and thus with weak patterns of social media posts. We split the train, validation, test set temporally with a ratio of 6:2:2.

5.2 Base Models

We use six representative text-based models as base models:

*Pattern-based Models*

- **Bi-LSTM** [15] is widely used in many existing works of our task for text encoding [16, 21, 37]. We implement a one-layer Bi-LSTM with a maximum sequence length of 100 and a hidden size of 128. We average all the hidden states as representations of posts which are further fed into an MLP for prediction.

- **EANN-Text** [51] is a model that tries to distract the fake news detection model from memorizing event-specific features. It uses TextCNN for text representation and adds an auxiliary task of event classification for adversarial learning using gradient reversal layer [12]. We re-implement the model according to the public code¹⁰. The complete EANN is a multi-modal model but we here use its text-only version. For TextCNN, the number of filters is 20 and the window sizes are {1, 2, 3, 4}. The labels for the auxiliary event classification task are derived by clustering the training set with K-means where K = 300.

- **BERT-Emo** [62] is a model that uses BERT to encode the text and captures the emotion that news publishers express. As we focus on the contents rather than social contexts, we adopt a simplified version where emotions in comments are not considered. We use the author-released code¹¹. The maximum sequence length is 150 and the size of embedding vectors is 768.

*Fact-based Models*

- **DeClarE** [34] is a model which uses claim-specific attention to focus on salient words in relevant articles. We remove the source embedding which is unavailable in the datasets. We re-implement the model according to the third-party code¹². The text encoder is a one-layer Bi-LSTM with the hidden size of 128.

- **EVIN** [56] is an evidence inference network, which captures the semantic conflicts between the post and relevant articles using the attention mechanism. We re-implement the model according to the paper as no public code is available. The hidden size of one-layer Bi-LSTM is 60. The maximum sequence length is 200.

- **MAC** [48] is a hierarchical multi-head attentive network that combines word- and article-level attention. We re-implement according to the authors’ released code¹³. We use one-layer Bi-LSTM networks with a hidden size of 300 to build MAC. Two multi-head attention modules have 5 and 2 heads, respectively.

Note that when base models are used as a module in Pref-FEND, we extract the last-layer feature before the MLP layer.
Table 3: Performance comparison with pattern- or fact-based models. w/ Pref-FEND$_S$ means the model is incorporated as a module of Pref-FEND$_S$ framework.

| Method                  | Weibo Acc. | macF1 | $P_{fake}$ | $R_{fake}$ | $F_{fake}$ | $P_{real}$ | $R_{real}$ | $F_{real}$ |
|-------------------------|------------|-------|------------|------------|------------|------------|------------|------------|
| Bi-LSTM                 | 0.667      | 0.600 | 0.626      | 0.820      | 0.710      | 0.744      | 0.516      | 0.610      |
| w/ Pref-FEND$_S$        | 0.709      | 0.709 | 0.696      | 0.735      | 0.715      | 0.723      | 0.683      | 0.702      |
| EANN-Text               | 0.692      | 0.690 | 0.860      | 0.785      | 0.717      | 0.739      | 0.601      | 0.663      |
| w/ Pref-FEND$_S$        | 0.740      | 0.740 | 0.760      | 0.697      | 0.722      | 0.723      | 0.783      | 0.752      |
| BERT-Emo                | 0.712      | 0.708 | 0.667      | 0.839      | 0.743      | 0.787      | 0.587      | 0.672      |
| w/ Pref-FEND$_S$        | 0.746      | 0.744 | 0.703      | 0.847      | 0.768      | 0.811      | 0.647      | 0.720      |

| Method                  | Twitter Acc. | macF1 | $P_{fake}$ | $R_{fake}$ | $F_{fake}$ | $P_{real}$ | $R_{real}$ | $F_{real}$ |
|-------------------------|--------------|-------|------------|------------|------------|------------|------------|------------|
| Bi-LSTM                 | 0.767        | 0.732 | 0.753      | 0.923      | 0.829      | 0.811      | 0.522      | 0.635      |
| w/ Pref-FEND$_S$        | 0.793        | 0.788 | 0.870      | 0.779      | 0.822      | 0.700      | 0.816      | 0.754      |
| EANN-Text               | 0.770        | 0.725 | 0.742      | 0.960      | 0.837      | 0.881      | 0.472      | 0.614      |
| w/ Pref-FEND$_S$        | 0.798        | 0.788 | 0.837      | 0.832      | 0.834      | 0.737      | 0.744      | 0.741      |
| BERT-Emo                | 0.794        | 0.762 | 0.769      | 0.950      | 0.850      | 0.873      | 0.550      | 0.675      |
| w/ Pref-FEND$_S$        | 0.804        | 0.776 | 0.781      | 0.945      | 0.855      | 0.870      | 0.582      | 0.697      |

Table 4: Performance comparison with integrated (pattern-and-fact-based) models.

| Method                  | Weibo Acc. | macF1 | $P_{fake}$ | $R_{fake}$ | $F_{fake}$ | $P_{real}$ | $R_{real}$ | $F_{real}$ |
|-------------------------|------------|-------|------------|------------|------------|------------|------------|------------|
| Last-layer Fusion       | 0.697      | 0.696 | 0.721      | 0.637      | 0.676      | 0.678      | 0.757      | 0.715      |
| Logits Average          | 0.692      | 0.685 | 0.646      | 0.840      | 0.730      | 0.776      | 0.544      | 0.640      |
| Pref-FEND               | 0.714      | 0.712 | 0.684      | 0.788      | 0.732      | 0.754      | 0.640      | 0.692      |

| Method                  | Twitter Acc. | macF1 | $P_{fake}$ | $R_{fake}$ | $F_{fake}$ | $P_{real}$ | $R_{real}$ | $F_{real}$ |
|-------------------------|--------------|-------|------------|------------|------------|------------|------------|------------|
| Last-layer Fusion       | 0.798        | 0.768 | 0.775      | 0.945      | 0.851      | 0.866      | 0.566      | 0.685      |
| Logits Average          | 0.784        | 0.750 | 0.762      | 0.943      | 0.843      | 0.855      | 0.534      | 0.657      |
| Pref-FEND               | 0.812        | 0.792 | 0.803      | 0.917      | 0.857      | 0.832      | 0.645      | 0.727      |

5.3 Experimental Setup

**Evaluation Metrics.** We report accuracy (Acc.) and macro F1 score (macF1). For each class, we also report precision, recall, and F1 score, denoted as $P_{cls}$, $R_{cls}$, and $F_{cls}$ where cls = \{fake, real\}.

**Implementation Details.** In Pref-FEND, the number of layers in HetDGCN-L is 2. We perform grid search in a small interval and finally let $\alpha = 0.5$, $\beta_1 = 2$, $\beta_2 = 1$, and $\beta_3 = 1$. For all base models and our Pref-FEND, the initial token embeddings are obtained from pretrained models in HuggingFace’s Transformers\[55\] (specifically, bert-base-chinese and bert-base-uncased). For all fact-based models, the top 5 retrieved articles are considered. Other hyperparameters have been described in Section 5.2. The methods are implemented with PyTorch [33] and Pytorch Geometric [10].

5.4 Performance Comparison (EQ1 & EQ2)

5.4.1 Comparing with Pattern- or Fact-based Methods. To fairly compare with existing single-preference (i.e., pattern- or fact-based) models, we reduce our framework to a single-model version named Pref-FEND$_S$. In detail, when comparing with a pattern-based model, we remove the fact-based model but preserve the Fact Preference Map for training; and vice versa. From the results in Table 3, we have the following observations:

First, Pref-FEND$_S$ successfully improves the performance of all the pattern-based and fact-based models on the two datasets. This verifies our observation that the original base models might be distracted from non-preferred information, which thus limits their generalizability to unseen samples. With the help of Pref-FEND$_S$, the base models are more focused during training.

Second, BERT-Emo outperforms Bi-LSTM and EANN-Text. This is as expected because BERT can generate expressive representations and the additional emotion-related features are proved helpful for this task. With the guidance of Pref-FEND$_S$, it gains a boost of 3.6 percent points in macro F1 scores on Weibo and a boost of 1.4 percent points on Twitter. This reveals the importance of preference modeling for alleviating the overfitting of specific features.

Third, MAC outperforms DeClarE and EVIN, though they are all based on the attention mechanism. This might be because it effectively uses multi-head attention to capture multi-aspect information. However, some heads might be distracted from the event description in the post, which can be alleviated by our framework.

5.4.2 Comparing with Integrated (Pattern-and-fact-based) Methods. We implement the following methods which fuse the information from pattern- and fact-based models:

- **Last-layer Fusion** which uses the post as input and concatenates the last-layer features of two models for final prediction;
- **Logits Average** which averages the models’ logits (which are in $[0,1]$) for final prediction.
We implement these fusion methods and Pref-FEND with two groups of base models, Bi-LSTM+DeClarE and BERT-Emo+MAC. The results are shown in Table 4. Our observations are as follows:

First, Pref-FEND outperforms the two pattern-and-fact-based methods, which validates its effectiveness for integrating pattern- and fact-based models.

Second, comparing with the results in Table 3, Pref-FEND brings further improvements based on the remarkable performance of Pref-FEND$_2$ w.r.t the same base models. For example, on the Weibo dataset, Pref-FEND with Bi-LSTM and DeClarE gains another increase of macro F1 by 0.3 percent points than Pref-FEND$_2$ with Bi-LSTM and 1.1 percent points than Pref-FEND$_2$ with DeClarE. This proves that our framework is applicable to both the single-preference models and the integrated models based on them.

Third, the last-layer fusion does not necessarily perform better than the simple logits average. This indicates that last-layer fusion may be insufficient to align the feature spaces of the pattern- and the fact-based model, which leads to negative fusion effects.

### Table 5: Ablation study of Pref-FEND.

| Method                      | Bi-LSTM (Pattern-based) + DeClarE (Fact-based) | Twitter                    |
|-----------------------------|------------------------------------------------|----------------------------|
| Pref-FEND                   | 0.714 0.712 0.684 0.788 0.732 0.754 0.640 0.692 | 0.812 0.792 0.803 0.917 0.857 0.832 0.645 0.727 |
| w/ rand init maps          | 0.694 0.693 0.676 0.736 0.705 0.715 0.652 0.682 | 0.788 0.765 0.787 0.896 0.838 0.790 0.616 0.692 |
| w/o $L_{cos}$              | 0.701 0.703 0.672 0.787 0.725 0.747 0.621 0.678 | 0.794 0.785 0.845 0.813 0.829 0.721 0.764 0.742 |
| w/o $L_{cls}(y_{rev}, y')$ | 0.703 0.702 0.710 0.679 0.694 0.696 0.725 0.710 | 0.792 0.764 0.775 0.932 0.846 0.842 0.571 0.681 |
| w/ only $L_{cls}(y, y')$   | 0.700 0.702 0.672 0.782 0.723 0.743 0.622 0.677 | 0.789 0.747 0.752 0.979 0.851 0.936 0.490 0.643 |

### Table 6: Frequently preferred tokens (separated by “|”) in the pattern-preferred and the fact-preferred token set.

| Set             | Category          | Token (Translated into English)                                                                 |
|-----------------|-------------------|-----------------------------------------------------------------------------------------------|
| Pattern-preferred | Punctuation       | , . : ! | : | . | . | | . . | @ | * | # | . | > | | > | | / |
|                 | Negation          | not | no | don’t |
|                 | Pronouns          | we | they | you |
|                 | Others            | find | such | think | may | certainly | so | release | some | as | careful | become | focus | loving | heart | but | kind | of |
| Fact-preferred  | Evidence-related  | claim | video | link | webpage | full text | say | picture | investigation | according to | uncover |
|                 | Entity-related    | China | Beijing | police | place | car | Shanghai | official |
|                 | Pronouns          | he | its | it | you |
|                 | Others            | ‘s | done | just | also | and | already | will | wait | go | to | do | female | too | want | what | certain | die | pass | death | when |
|                 |                   | in | second | more | make | suffer | night | society |

We implement these fusion methods and Pref-FEND with two groups of base models, Bi-LSTM+DeClarE and BERT-Emo+MAC. The results are shown in Table 4. Our observations are as follows:

First, Pref-FEND outperforms the two pattern-and-fact-based methods, which validates its effectiveness for integrating pattern- and fact-based models.

Second, comparing with the results in Table 3, Pref-FEND brings further improvements based on the remarkable performance of Pref-FEND$_2$ w.r.t the same base models. For example, on the Weibo dataset, Pref-FEND with Bi-LSTM and DeClarE gains another increase of macro F1 by 0.3 percent points than Pref-FEND$_2$ with Bi-LSTM and 1.1 percent points than Pref-FEND$_2$ with DeClarE. This proves that our framework is applicable to both the single-preference models and the integrated models based on them.

Third, the last-layer fusion does not necessarily perform better than the simple logits average. This indicates that last-layer fusion may be insufficient to align the feature spaces of the pattern- and the fact-based model, which leads to negative fusion effects.

### 5.5 Ablation Study (EQ3)

We study the effectiveness of our designed components or strategies based on the Pref-FEND models in Table 4.

#### 5.5.1 Effectiveness of Model Preference Learning.

Instead of recognizing the entities and stylistic tokens according to the prior knowledge, we randomly initialize preference maps (named as Pref-FEND w/ rand init maps). That forces the generation of preference maps to rely only on the supervision of ground-truth labels. The results show that although Pref-FEND w/ rand init maps is superior or comparable to the base models on both of the two datasets in terms of accuracy and macro F1, it falls behind the complete Pref-FEND. This proves the effectiveness of our model preference learning, which exploits prior knowledge in a dynamic graph representation learning process.

#### 5.5.2 Effective of Losses for Differentiating the Preference Maps.

We remove one of the two losses which aim at differentiating the two preference maps, or both. The variants are with the suffixes w/o $L_{cos}$, w/o $L_{cls}(y_{rev}, y')$, and w/ only $L_{cls}(y, y')$, respectively. We see that removing these losses brings performance drops w.r.t. accuracy. The largest drop occurs when removing both the two losses. This indicates that the auxiliary losses are effective and necessary to integrate the two models with different preferences.

### 5.6 Preference Map Analysis (EQ4)

#### 5.6.1 Analysis on Most Frequent Token Set.

To explore how different the Fact and the Pattern Preference Map are, we analyze the frequently preferred tokens in the Maps. For each post in the validation and test sets of Weibo, we first divide the tokens into a pattern group and a fact group, which indicates this token is scored higher in the Pattern or the Fact Preference Map. Then we extract the top 10 tokens in each group of all the posts and construct two token sets for frequency analysis. The frequent tokens in each set are shown with fine-grained categories in Table 6. We see that:
Table 7: Three examples of fake news posts. Red represents pattern-preferred tokens and blue represents fact-preferred tokens. Darker color indicates a higher preference score.

| # | Post (Translated into English) | Ground Truth: Fake | Judgment: Bi-LSTM (Real), DeClarE (Real), Pref-FEND (Fake) |
|---|--------------------------------|-------------------|------------------------------------------------------------|
| 1 | A group of city administration officials in Sishui, Shandong, chased an old man until all his eggs were broken on the ground. The old man hit helpless. The officials ran away after hitting. The white-haired man should be about 80 years old, and he can’t make much money by selling eggs. So why be aggressive? Is there no time for the officials to be alone? If the officials only oppress citizens, what’s the good of having these officials? You will be punished sooner or later for bullying the underprivileged. | Fake | Bi-LSTM (Fake), DeClarE (Real), Pref-FEND (Fake) |
| 2 | A student of ZJU jumping to the West Lake for a crazy graduation photo drowned. On June 29, Xin (not his real name) from ZJU and his classmates went to the waters near the scenic spot of Xiling bridge on Beishan Road. Xin asked his classmates to take pictures of him swimming underwater. He jumped into the West Lake from the side of the West Lake. Recently, he has received a full PhD scholarship from a U.S. university. | Fake | Bi-LSTM (Real), DeClarE (Fake), Pref-FEND (Fake) |
| 3 | Is anyone interested in raising a dog in Shanghai? No Charge. If no one adopts, they will be euthanized. Let these little cute lives accompany with you. If you are really not able to raise them, please forward this message. | True | Bi-LSTM (Real), DeClarE (Real), Pref-FEND (Fake) |

First, in a pattern-preferred token set, punctuations and negation words are important as they express the publishers’ tones and emotions. The other frequent tokens are closely related to self-expression, like “think”, “may”, and “kind of”.

Second, in the fact-preferred set, evidence-related tokens that indicate materials and actions (e.g., “video”, “webpage”, “picture”, “claim”, and “uncover”) and entity-related tokens (places, positions, etc.) are more focused. Some of the other words do not directly describe an event, but are often around the elements of a news event (e.g., “already” and “when”).

Third, the focus on pronouns is different between the Pattern and the Fact Preference Map. Plural personal pronouns (“we”, “they”, and “you all”) are frequently focused by pattern-based models, while single ones (“he”, “it”, and “you”) are preferred by fact-based models. The reason might be that a post with significant fake news patterns often discusses some groups or inspires the audience to take action, while a post with an event description is generally related to specific persons or things.

Our analysis reveals that the learned preference maps are highly correlated to the ideal model preferences and thus effective for the guidance of models’ focuses.

5.6.2 Case study. In Table 7, we show three fake news posts that are successfully judged by Pref-FEND with Bi-LSTM and DeClarE. Case 1 conveys strong signals of emotional patterns, which are preferred by pattern-based models, such as “helplessly”, and “aggressive”. Case 2 contains a large number of places and event descriptions, which is friendly to utilize the evidential texts in relevant articles. Due to the different dominant signals, the pattern-based Bi-LSTM judges correctly in Case 1, but fails in Case 2. And the judgments of the fact-based DeClarE are the opposite. However, in Case 3, both of them wrongly judge this post as real. Based on the observation, a pattern-based model can attend to the emotion trigger tokens like “cute” and “really”, while a fact-based model can use the place (“Shanghai”) and the dog breed (“Golden Retriever”) to find evidence. Generally, it is unlikely that the two models both fail. We speculate that the failure is led by the negative interference from the non-preferred information. With the help of model preference learning, our Pref-FEND, however, succeed in judging all three posts as fake. These cases demonstrate the necessity of model preference learning and the effectiveness of Pref-FEND.

6 CONCLUSION AND FUTURE WORK

We propose the framework Pref-FEND to integrate the pattern-based and fact-based fake news detection models in a preference-aware fashion. The learned preference maps guide the models to focus more on their preferred parts with less interference by the non-preferred parts. Experiments on the two newly constructed datasets show that Pref-FEND outperforms the existing detection models. Further analysis shows that preference learning helps models of different preferences more focused and thus makes both the single-preference and the integrated models better-performing.

How to enhance the interaction between the preference map generation and specific models and how to extend the framework to multi-class and multi-preference scenarios are expected to be explored in the future. The acquisition and exploitation of prior knowledge in this task are also worth studying further to improve overall performance.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by the National Key Research and Development Program of China (2017YFC0820604), and the National Natural Science Foundation of China (U1703261).

REFERENCES

[1] Oluwaseun Ajao, Deepayan Bhowmik, and Shahrazad Zargari. 2019. Sentiment aware fake news detection on online social networks. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2507–2511. https://doi.org/10.1109/ICASSP.2019.8683170
