International Journal of Computational Intelligence Systems (2023) 16:176
https://doi.org/10.1007/s44196-023-00359-7

RESEARCH ARTICLE

# Adversarial Distillation Adaptation Model with Sentiment Contrastive Learning for Zero-Shot Stance Detection

**Yu Zhang**[1] · **Chunling Wang**[2] · **Jia Wang**[1]

Received: 18 April 2023 / Accepted: 28 October 2023


**Abstract**
Zero-shot stance detection is both crucial and challenging because it demands detecting the stances of previously unseen targets in the inference stage. Learning transferable target invariant features effectively from training data is crucial for zero-shot stance detection. This paper proposes an adversarial adaptation approach for zero-shot stance detection, which applies an adversarial discriminative domain adaptation network to transfer knowledge efficiently. Specifically, the proposed model applies knowledge distillation to prevent overfitting the destination data and forgetting the learned source knowledge. Moreover, stance contrastive learning is applied to enhance the quality of feature representation for superior generalization, and sentiment information is extracted to assist with stance detection. The experimental results indicate that our model performs competitively on two benchmark datasets.

**Keywords** Stance detection · Zero-shot learning · Domain adaptation · Contrastive learning

**Abbreviations**

| | |
|---|---|
| GAN | Generative adversarial network |
| DANN | Domain adversarial neural network |
| ADDA | Adversarial discriminative domain adaptation |
| ADSC | Adversarial distillation with sentiment contrastive learning |
| FM | Feminist movement |
| LA | Legalization of abortion |
| DT | Donald Trump |
| HC | Hillary Clinton |
| A | Atheism |
| CC | Climate change |

## 1 Introduction

Stance detection is a significant research in sentiment analysis and text mining, which focuses on the stance (e.g., Favor, Against, or Neutral) expressed in text toward a given target [1–3]. It can be effectively applied to social opinion analysis [4], rumor detection [5], and other research fields by mining text opinions.

Traditional stance detection [3] has a limited range of applications since it requires training and testing under the same target and depends on a lot of labeled data to achieve excellent performance. However, topics on social media platforms are updated frequently and in great quantities, as well as manually labeling new targets is expensive and laborious, making it impossible to create a labeled dataset with all prospective targets [6]. Therefore, the study of zero-shot stance detection for unseen targets is essential and promising [7].

For the zero-shot stance detection task, existing works generally incorporate external knowledge as support for inference [8, 9] or introduce an attention mechanism to capture the relationships between targets [7]. However, none of these approaches consider explicit modeling of the transferable knowledge between source and destination targets. Some works employ adversarial learning to make the model learn the target invariant representation [10]. Still, their adversarial learning strategy is extremely unstable and prone to degrading prediction performance when the target distribution is unbalanced. As shown in Table 1, zero-shot stance detection identifies the stance of an unknown target by training on numerous targets with labels; for example, the test set may contain the target "Feminist Movement", while the training set contains targets such as "Donald Trump"

✉ Yu Zhang
zhangyu2021@mail.dlut.edu.cn

[1] School of Marxism, Dalian University of Technology, Dalian 116024, Liaoning, China

[2] School of Information Science and Technology, Dalian Maritime University, Dalian 116024, Liaoning, China

Published online: 07 November 2023

**Table 1** Examples of zero-shot stance detection

| Training data | | Test data |
|---|---|---|
| Source Target1: Donald Trump | Source Target2: Hillary Clinton | Destination target: Feminist Movement |
| Text: Donald J. Trump, I am voting for you to be our next "El Presidente"! | Text: How could anyone vote for that woman? | Text: Now let's raise the pay for females and make it equal to what men get payed |
| Stance: Favor | Stance: Against | _ |

and "Hillary Clinton". In order to effectively generalize to unknown targets, it is essential to learn transferable stance feature knowledge from the training data. Hence it is especially crucial to find appropriate and effective knowledge transfer methods. In addition, we find a certain correlation between sentiment information and stance detection [9]. For instance, when a document contains some positive words, it generally implies a Favor stance. Stance detection will perform better if some sentiment knowledge can be acquired concurrently.

To address the above challenges, we propose an adversarial distillation adaptation model with sentiment contrastive learning. Specifically, since the training and test sets of zero-shot stance detection belong to different targets (domains), the domain adaptation method can be adopted to transfer knowledge. We employ an adversarial discriminative domain adaptation network [11]. By obfuscating a domain discriminator, the model is motivated to learn more target invariant features to ensure the transferability of information across different targets. Moreover, we consider that catastrophic forgetting occurs when the adversarial network is applied to the BERT model [21]. Knowledge distillation [22] can serve as a regularization method that maintains the information learned from the source data while being adaptable to the destination data. Supervised contrastive learning is also applied to generalize to unknown target stance detection by distinguishing stance category features in the potential distribution space. Given that stance detection is influenced by sentiment information, we employ the cross-attention module to inject the sentiment knowledge encoded by SentiBERT into BERT and adjust the fusion process according to the training loss of stance detection.

The contributions of our work can be summarized as follows:

1. We apply an adversarial discriminative domain adaptation network with knowledge distillation to solve the target knowledge transfer problem for zero-shot stance detection while improving the stability of the adversarial training.
2. The proposed model employs supervised contrastive learning to learn enhanced target invariant representations by learning correlations and differences between data with different stance labels. Sentiment information is extracted to assist in stance detection.
3. Experimental results on two datasets show that our method obtains competitive results compared to several strong baselines.

## 2 Related Work

### 2.1 Zero-Shot Stance Detection

Stance detection aims to identify the attitude of a text on a prescriptive target [1]. Most previous studies concentrated on intra-target stance detection, where the training and testing phases shared identical target sets [2, 3]. However, there is insufficient labeled data when new topics emerge. As a result, some studies explored cross-target stance detection [14–16], which involved training the model on one target and testing it on another related target. Xu et al. [16] presented a self-attentive model that extracted shared features learned from source targets to the destination target. Wei et al. [15] further exploited the hidden topics between targets as transferred knowledge. In contrast to cross-target settings, zero-shot stance detection does not require a prior assumption of target correlation. It is a more general study that can effectively deal with the reality that targets appear irregularly.

For zero-shot stance detection, Allaway et al. [7] created a dataset containing many targets and proposed a topic grouping attention model that implicitly captured the relationships between targets by generating generalized topic representations. Liu et al. [8] proposed a common sense knowledge augmentation graph model based on GCN and BERT, which utilized text information and relationship graph structure information to increase the generalization and reasoning capabilities of the model. Liang et al. [12] proposed a hierarchical contrastive learning model based on an agent task that distinguished the types of stance expressions to aid zero-shot stance detection.

### 2.2 Domain Adaptation

Domain adaptation can effectively deal with the problem of inadequate labeling data. It can compensate for the absence of label information in the destination domain by using sufficient label information in the source domain. The purpose of domain adaptation is to reduce domain differences and effectively transfer knowledge. Inspired by generative adversarial network (GAN) [17], adversarial loss methods have been commonly applied to domain adaptation. In the domain adversarial neural network (DANN) [18], a gradient inversion layer was presented to confuse the domain discriminator and enable the feature extractor to acquire domain invariant knowledge. Adversarial discriminative domain adaptation (ADDA) [11] used an adversarial framework that included discriminative models, unshared weights, and GAN loss.

Allaway et al. [10] regarded each target as a domain and modeled zero-shot stance detection as a domain adaptation problem, which successfully learned the target invariant representation. Inspired by the above works, we explore employing a more robust and efficient ADDA framework to handle zero-shot stance detection.

## 3 Methods

In this section, we introduce our proposed adversarial distillation adaptation network with sentiment contrastive learning for zero-shot stance detection (ADSC) in detail. As shown in Fig. 1, the model consists of two main parts.

1. Pretraining: we pretrain the source encoder with sentiment information and the classifier on the source labeled data while designing stance contrastive learning.
2. Adversarial distillation domain adaptation: we initialize the target encoder with the source encoder's parameters and train it via adversarial learning and knowledge distillation. The dotted box indicates that the parameters are fixed.

### 3.1 Task Description

Suppose we are given a set of labeled data $D_s = \{(x_s^i, t_s^i, y_s^i)\}_{i=1}^{N_s}$ from source targets and a set of unlabeled data $D_d = \{(x_d^i, t_d^i)\}_{i=1}^{N_d}$ from a destination target (unknown target), where $x$ is a document, $t$ and $y$ are its corresponding target and stance label, respectively, and $N$ is the number

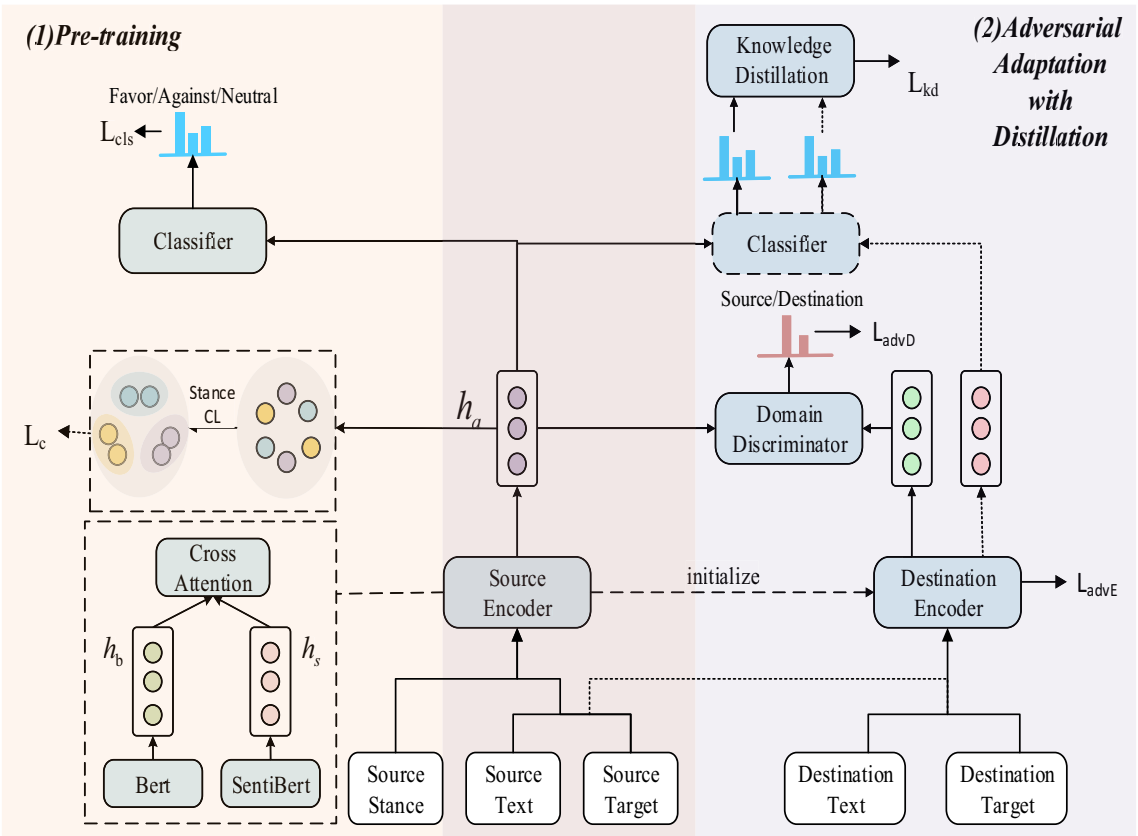


**Fig. 1** Overview of the ADSC model

of examples. The purpose of zero-shot stance detection is to train the model according to the labeled data of multiple source targets to predict the stance labels of the unknown target examples.

### 3.2 Encoder with Sentiment Information

Considering that the stance of a text is influenced by sentiment information, we learn the sentiment knowledge of the text to increase prediction accuracy. Following Zhou et al. [19], we exploit a perceptual sentiment language model (SentiBERT) to extract sentiment knowledge.

The SentiBERT framework includes sentiment masking and several pretraining goals. We first mask some tokens, including ordinary words, sentiment words, and emoticons. Sentiment words and emoticons are masked with a higher probability than ordinary words to emphasize the sentiment information of the sentence. Therefore, sentiment information can be learned through recovery. The pretraining goal requires the encoder to reconstruct masked sentiment tokens and predict the sentiment ratings of the whole sentence.

Specifically, the masked corrupted text $\hat{x}$ is input to the BERT encoder to obtain the representation $h_i$ of each word and the final state $h_{[CLS]}$ as the sentence representation. The softmax function is used on $h_i$ to predict the probability, sentiment polarity, and emoticon probability of each word separately. The overall sentiment score of the text $\hat{x}$ is predicted using a softmax layer on $h_{[CLS]}$. Each task is jointly trained and optimized. SentiBERT performs well in the cross-domain sentiment analysis task after being trained on the Amazon Review dataset and the Yelp 2020 challenge dataset.

Therefore, we adopt a pretrained SentiBERT model and input the given document $x$ and target $t$ into the model in the form of "$[CLS]x[SEP]t[SEP]$" to obtain a hidden vector representation $h_s$ with sentiment information.

$$h_s = SentiBERT([CLS]x[SEP]t[SEP]) \tag{1}$$

SentiBERT can be utilized as an outstanding sentiment feature extractor since it has successfully learned sentiment knowledge in large-scale datasets. We fix the parameters of SentiBERT during the training process to keep sentiment information stable.

Moreover, to take advantage of the contextual information, we also adopt a pretrained BERT [13] model to jointly embed document $x$ and target $t$ to obtain a hidden vector representation $h_b$ of each example.

$$h_b = BERT([CLS]x[SEP]t[SEP]) \tag{2}$$

Then $h_b$ and $h_s$ are concatenated, and the information of both is fused by the cross-attention module. Cross-attention can effectively capture the interdependencies between text and sentiment, facilitating the integration of knowledge and resulting in the generation of more accurate and meaningful features. The hidden state of the [CLS] token is used as the final output $h_a$:

$$h_a = CrossAttention\left(\left[h_b, h_s\right]\right)[CLS] \tag{3}$$

### 3.3 Stance Contrastive Learning

Contrastive learning allows the feature representation of the anchor to be similar to the positive examples and dissimilar to the negative examples [11, 20]. A superior semantic representation space can be learned from the examples by using the pair-based contrastive loss function. Supervised contrastive learning can bring examples belonging to the same class closely together and push examples of different classes away from each other, effectively improving the quality of feature representation.

To improve the generalization ability of the stance representation, based on the stance label information of the examples, we perform contrastive learning on their hidden vectors. Specifically, given the hidden vectors $H = \{h_i\}_{i=1}^{N_b}$ of a batch of examples (where $N_b$ is the size of the batch), for a specific anchor $h_i \in H$, if $h_j \in H$ and $h_i$ have the same stance label, i.e., $y_j = y_i$ (where $y_j$ and $y_i$ are the stance labels of $h_j$ and $h_i$ respectively), then $h_j$ is considered to be a positive example of $h_i$, while other examples $h_k \in H$ are considered to be negative examples of $h_i$. The final contrastive loss is calculated over all positive pairs, including $(h_i, h_j)$ and $(h_j, h_i)$ in a batch:

$$L_c = \frac{1}{N_B} \sum_{h_i \in H} l\left(h_i\right) \tag{4}$$

$$l\left(h_i\right) = -\log \frac{\sum_{j=1}^{N_b} 1_{[j \neq i]} 1_{[y_i = y_j]} \exp\left(sim\left(\boldsymbol{h_i}, \boldsymbol{h_j}\right)/\tau\right)}{\sum_{k=1}^{N_b} 1_{[k \neq i]} \exp\left(sim\left(\boldsymbol{h_i}, \boldsymbol{h_k}\right)/\tau\right)} \tag{5}$$

$$sim(\boldsymbol{m}, \boldsymbol{n}) = \boldsymbol{m}^{\mathrm{T}} \boldsymbol{n} / ||\boldsymbol{m}|| ||\boldsymbol{n}|| \tag{5}$$

where $1_{[i=j]} \in \{0, 1\}$ is an indicator function that evaluates to $\mathbf{1}$ iff $i = j$. $sim(\boldsymbol{m}, \boldsymbol{n})$ represents the cosine similarity of vectors $\boldsymbol{m}$ and $\boldsymbol{n}$.$\tau$ denotes a temperature parameter.

### 3.4 Training

Since the source and destination data come from distinct targets (domains), directly applying the model trained on the source data to the destination data has poor performance because of domain bias. To achieve effective domain transfer, we must make predictions based on features that cannot tell the training (source) and testing (destination) domains apart. Therefore, we employ an adversarial learning-based

domain adaptive approach to learn domain invariant information.

In the domain adaptation task, we have obtained the labeled source data $D_s = \{(x_s^i, t_s^i, y_s^i)\}_{i=1}^{N_s}$ and the unlabeled destination data $D_d = \{(x_d^i, t_d^i)\}_{i=1}^{N_d}$, and they have identical label distribution spaces. We regard $E_s$ as the source encoder function and $E_d$ as the destination encoder function, and they map the input data $d$ (including text $x$ and target $t$) to the encoder output $h$. $C$ denotes the classifier function that converts the encoder output to the stance category. $D$ denotes the discriminator function that converts the encoder output to the domain category (source or destination). We wish to learn a destination encoder $E_d$ and a destination classifier $C_d$ that can accurately predict the stance class of the destination examples in the absence of labels. As a result, we reduce the distance between the source and destination data representations through adversarial training. In this case, it can be considered that the source and destination domains have identical distributions in the mapped space. Then, the source classifier $C_s$ can be applied directly for stance detection on the destination data without learning a separate destination classifier. So we set $C = C_s = C_d$.

#### 3.4.1 Pretraining

We train the source encoder with sentiment information $E_s$ and the classifier $C$ on the text, target and label pairs $(x_s, t_s, y_s) \in \{0, 1, 2\}$ from the source dataset $D_s$ in a supervised manner. Furthermore, we enable the encoder to learn a superior class representation by minimizing the stance contrastive loss $L_c$ (see Eq. 4) and improve the performance of the classifier by minimizing the standard cross-entropy loss $L_{cls}$. The final loss is the sum of the two losses:

$$_{E_s,C}{}^{min}L_{cls} = -E_{(x_s,t_s,y_s)\sim D_s} \sum_{k=1}^{K} 1_{[k=y_s]} log\big(C\big(E_s\big(x_s,t_s\big)\big)\big) \tag{7}$$

$$L_{all} = L_{cls} + L_c \tag{8}$$

where k is the specific category and $K$ is the number of categories. $1_{[k=y_s]} \in \{0, 1\}$ is an indicator function that evaluates to 1 iff $k = y_s$. The parameters of the source encoder and source classifier are fixed at the end of pretraining.

#### 3.4.2 Adversarial Adaptation with Distillation

We initialize the destination encoder $E_d$ with the parameters of the pretrained source encoder. We fix the source encoder during adversarial training and use it as a reference to make the target representation match the source distribution as closely as possible. The following loss $L_{adv_E}$ can be optimized to produce a fantastic target encoder.

$$_{E_d}{}^{min}L_{adv_E} = -E_{(x_d,t_d)\sim D_d} \log\big(D\big(E_d\big(x_d,t_d\big)\big)\big) \tag{9}$$

The domain discriminator $D$ is designed to differentiate whether the data feature representations originate from the source or destination domain. $D$ is optimized according to the standard supervised loss $L_{adv_D}$, where the labels point to the origin domain.

$$\begin{aligned} _{D}{}^{min}L_{adv_D} = &-E_{(x_s,t_s)\sim D_s} log\big(D\big(E_s\big(x_s,t_s\big)\big)\big) \\ &-E_{(x_d,t_d)\sim D_d} log\big(1-D\big(E_d\big(x_d,t_d\big)\big)\big) \end{aligned} \tag{10}$$

Although the destination encoder contains unbound weights from the source encoder, this offers it more flexibility to learn features of the destination domain while also preventing it from learning degenerate solutions. However, as new domains are added during the training process, the previously learned domain features are gradually forgotten, thus overfitting the target data. The inaccessibility of class labels and the difference from the original task lead to random classification performance [21].

To improve the stability of adversarial training and prevent pattern collapse, we employ a regularization approach to mitigate catastrophic forgetting. Knowledge distillation can provide the model with flexible adversarial adaptation and the capability to keep class information at high values of temperature $t$ [21]. The loss of knowledge distillation is as follows:

$$\begin{aligned} L_{kd} = &-t^2 \times E_{(x_s,t_s)\sim D_s} \sum_{k=1}^{K} softmax\big(f_k^s/t\big) \\ &\times log\big(softmax\big(f_k^d/t\big)\big) \end{aligned} \tag{11}$$

where $f^s = C(E_s(x_s, t_s)), f^d = C(E_d(x_s, t_s))$. We sequentially feed the data into the encoder and classifier to obtain the probability distribution of the stance and normalize it with the softmax function. Thus, the loss function for training the destination encoder is:

$$L = \alpha L_{adv_E} + \beta L_{kd} \tag{12}$$

where $\alpha$ and $\beta$ are tuning hyperparameters. All methods minimize the source and destination representation distances by alternating between the destination encoder and the discriminator. We conduct adversarial adaptation by learning the destination encoder so the discriminator cannot accurately predict the domain labels of the source and destination examples based on their feature representations.

### 3.5 Testing

We utilize the destination encoder obtained after adversarial domain adaptation and the classifier with fixed parameters to predict the stance of the destination examples.

$$\widehat{y_d} = argmax\big(C\big(E_d\big(x_d, t_d\big)\big)\big) \tag{13}$$

# 4 Experiment

## 4.1 Datasets

SEM16 [3] is a Twitter dataset that contains six targets for stance detection, including the Feminist Movement (FM), Legalization of Abortion (LA), Donald Trump (DT), Hillary Clinton (HC), Atheism (A), and Climate Change is a Real Concern (CC). Each text in the dataset contains a stance (favor, against, neutral) for a specific target.

WT-WT [23] is a stance detection dataset in the financial domain. The dataset contains four targets, including CVS_AET(CA), CI_ESRX (CE), ANTM_CI (AC), and AET_HUM (AH). Every example involves a stance label of refute (against), support (favor), comment (neutral), and irrelevant opinion. We eliminate text labeled as irrelevant to ensure consistency with other datasets.

Following[12], we utilize the data from one target as the test set and the remaining targets as the training set. Table 2 represents the statistics of the two datasets.

## 4.2 Experimental Implementation

### 4.2.1 Training Settings

We employ the pretrained SentiBERT model provided by Zhou et al. as well as the pretrained uncased BERT as the encoder, and their maximum sequence length is 85. The batch size is 32. In the pretraining phase, the source encoder and classifier are trained for 3 epochs using the Adam optimizer [24] with a learning rate of 5e−5, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. In the adversarial domain adaptation phase, we also use the unlabeled data from the destination domain to train the destination encoder and discriminator for 3 epochs with a learning rate of 1e−5. The temperature value $t$ for knowledge distillation is set to 20. We also apply a gradient clip to a target encoder with a gradient norm of 1.0 and a discriminator with a clip value of 0.01 to increase the stability of the adversarial training [21]. The temperature parameter for the contrastive loss is 0.07.

**Table 2** The statistics of the SEM16 and WT-WT datasets

| Dataset | Target | Favor | Against | Neutral | Unlabeled |
|---|---|---|---|---|---|
| SEM16 | DT | 148 | 299 | 260 | 2194 |
| | HC | 163 | 565 | 256 | 1898 |
| | FM | 268 | 511 | 170 | 1951 |
| | LA | 167 | 544 | 222 | 1899 |
| | A | 124 | 464 | 145 | 1900 |
| | CC | 335 | 26 | 203 | 1900 |
| WT-WT | CA | 2469 | 518 | 5520 | – |
| | CE | 773 | 253 | 947 | – |
| | AC | 970 | 1969 | 3098 | – |
| | AH | 1038 | 1106 | 2804 | – |

### 4.2.2 Evaluation Metric

For the SEM16 dataset, following [10], we report the $F_{avg}$: the average of F1 for favor and against. For the WT-WT dataset, following [23], we report the Macro F1 scores of each target.

## 4.3 Baselines

To demonstrate the validity of the proposed model, we compare the ADSC with several strong baselines.

- *BiCond* [2] A model that utilizes two BiLSTM layers to encode topic and text separately.
- *CrossNet* [16] A BiCond-based model for adding topic-specific self-attentive layers.
- *TOAD* [10] A BiCond-based model with adversarial learning.
- *BERT* [13] A powerful pretrained language model for NLP tasks.
- *BERT-GCN* [8] A BERT-based model using GCN for node information aggregation.
- *TGA Net* [7] A topic-group attention model.
- *TPDG* [14] A GCN-based model for designing target-adaptive pragmatic dependency graphs.

In addition, we designed several variants of the ADSC model to conduct ablation studies to verify the validity of different components.

1. "w/o $L_c$" denotes without stance contrastive learning loss.
2. "w/o *SentiBERT*" denotes that SentiBERT is not utilized to extract sentiment information.
3. "w/o $L_{kd}$" denotes without knowledge distillation loss.

## 4.4 Main Results

The results of the comparison experiments are shown in Table 3. It can be observed that our proposed ADSC model achieves competitive and stable performance on most of the target sets, which validates the effectiveness of our approach to this task. Specifically, BiCond and CrossNet perform the worst overall, and BERT and BERT-GCN perform similarly poorly since they do not consider the targets' invisibility to learn transferable information. Despite adopting an adversarial strategy as well, the TOAD model is generally inferior to our method. It is demonstrated that we utilize a sophisticated adversarial domain adaptation network and

**Table 3** Experimental results on two datasets

| Model | | BiCond | CrossNet | TOAD | BERT | BERT-GCN | TPDG | TGA Net | ADSC |
|---|---|---|---|---|---|---|---|---|---|
| SEM16 | DT | 30.5 | 35.6 | 49.5 | 40.1 | 42.3 | 47.3 | 40.7 | **58.3** |
| | HC | 32.7 | 38.3 | **51.2** | 49.6 | 50.0 | 50.9 | 49.3 | 50.1 |
| | FM | 40.6 | 41.7 | **54.1** | 41.9 | 44.3 | 53.6 | 46.6 | 50.0 |
| | LA | 34.4 | 38.5 | 46.2 | 44.8 | 44.2 | 46.5 | 45.2 | **46.7** |
| | A | 31.0 | 39.7 | 46.1 | **55.2** | 53.6 | 48.7 | 52.7 | 50.0 |
| | CC | 15.0 | 22.8 | 30.9 | **37.3** | 35.5 | 32.3 | 36.6 | 33.5 |
| WT-WT | CA | 56.5 | 59.1 | 55.3 | 56.0 | 67.8 | 66.8 | 65.7 | **68.1** |
| | CE | 52.5 | 54.5 | 57.7 | 60.5 | 64.1 | 65.6 | 63.5 | **67.5** |
| | AC | 64.9 | 65.1 | 58.6 | 67.1 | 70.7 | **74.2** | 69.9 | 72.3 |
| | AH | 63.0 | 62.3 | 61.7 | 67.3 | 69.2 | **73.1** | 68.7 | 71.6 |

Bold indicates the best score for each test target

add knowledge distillation to enhance the stability of adversarial training while ensuring the effective transfer of the target knowledge. In contrast to the attention-based model, our method effectively generalizes the stance representation learned from known targets to unseen targets by exploring contrastive learning.

## 4.5 Ablation Study

We further conduct ablation studies to analyze the impact of different components of ADSC. As shown in Table 4, the experimental results show that removing stance contrastive learning ("w/o $L_c$") significantly decreases the model's performance. This suggests that supervised contrastive learning during the pretraining phase assists the encoder in learning better class representations, improving generalizability. The removal of sentiment information ("w/o *SentiBERT*") reduces model performance, implying that the model may learn the potential relationship between sentiment and stance and make judgments on the stance with the help of sentiment information. For example, the model learns a strong association between positive sentiment words and support stances and weak associations between negative sentiment words and support stances. The effect of removing knowledge distillation ("w/o $L_{kd}$") becomes worse, which indicates that some source information is forgotten during adversarial training. So regularization of knowledge distillation is useful in improving performance.

## 4.6 Analysis of Contrastive Learning

To further analyze the effectiveness of stance contrastive learning in the model, we use T-SNE [25] to visualize the intermediate layer embedding. The visualization results without and with contrastive learning are shown in Fig. 2. It can be observed that the representation distributions without using contrastive learning have great overlap, especially for the favor and against stances. This suggests that contrastive learning may effectively separate the representations of different stances and learn a better potential space, further demonstrating its effectiveness and significance.

## 4.7 Analysis of Adversarial Domain Adaptation

To further understand the influence of adversarial domain adaptation on the zero-shot stance detection task, we employ t-SNE to visualize the feature distribution encoded by the destination encoder. Domain invariance is determined by the degree of overlap between features. As shown in Fig. 3, we employ the destination encoder to encode both the source and destination data. Domain adaptation makes the domain overlap more prominent. This demonstrates that adversarial domain adaptation may align the source and target domain feature distributions as nearly as feasible, resulting in significant target invariant features.

**Table 4** Experimental results of the ablation study

| Model | SEM16 | | | | | | WT-WT | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DT | HC | FM | LA | A | CC | CA | CE | AC | AH |
| ADSC | 58.3 | 50.1 | 50.0 | 46.7 | 50.0 | 33.5 | 68.1 | 67.5 | 72.3 | 71.6 |
| w/o $L_c$ | 57.1 | 49.0 | 48.8 | 45.5 | 48.7 | 32.3 | 66.8 | 66.8 | 71.1 | 70.4 |
| w/o *SentiBERT* | 57.9 | 49.6 | 49.3 | 46.0 | 49.4 | 33.0 | 67.6 | 67.0 | 71.6 | 70.8 |
| w/o $L_{kd}$ | 52.3 | 49.1 | 48.5 | 45.7 | 47.6 | 32.3 | 66.7 | 67.0 | 71.5 | 70.4 |

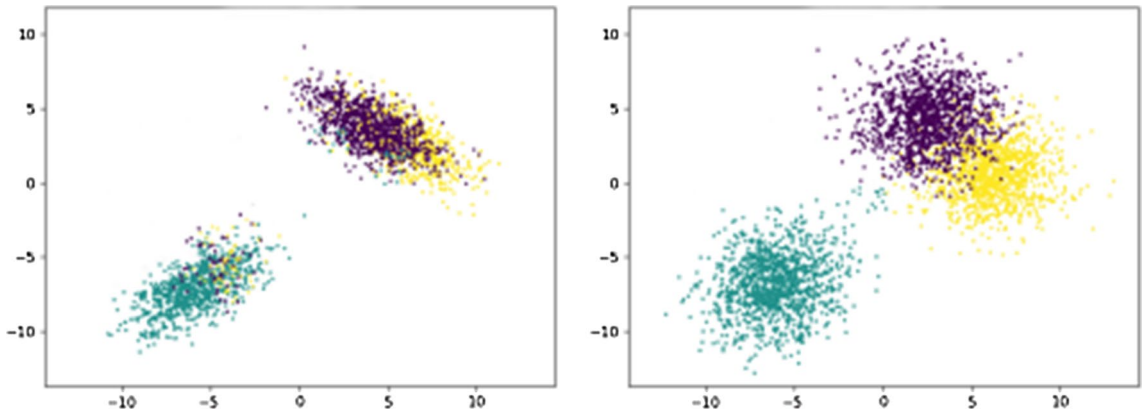

**Fig. 2** Visualization of intermediate embeddings. The left figure is the visualization with contrastive learning, and the right figure is the visualization without contrastive learning. Purple dots indicate favor examples, yellow dots indicate against examples, and green dots indicate neutral examples

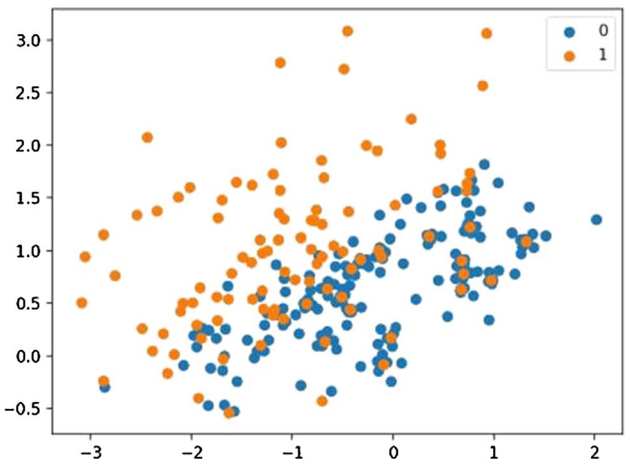


**Fig. 3** Visualization of the distribution of features. The source domain features are represented by 1, and the destination domain features are represented by 0

## 4.8 Case Study

We conduct a case study to illustrate the validity and perform error analysis. We select three cases from the test data of SEM16 and compare our results to the predictions of BERT and TOAD. Table 5 reports these results.

In the first case, TOAD with adversarial learning and our model accurately forecast the outcome while BERT predicts it incorrectly. This is primarily because BERT does not learn transferable knowledge for unknown targets, whereas exploring adversarial domain adaptation approaches can effectively learn target invariant information and increase generalization ability. In the second case, only our method makes the correct prediction. This demonstrates that depending only on contextual information is insufficient and adding sentiment information strengthens the model's comprehension of texts with the sarcastic sentiment. In the third case, all three methods make incorrect predictions. We speculate that the model does not understand the hidden relationship between "NBC" and "Donald Trump", and it is difficult to make correct predictions for sentences that contain underlying ideas or require more profound understanding. Thus, domain knowledge is beneficial to the model. In the future, we will explore the introduction of common sense knowledge of the destination domain, which may significantly improve the model's generalizability.

**Table 5** Three cases of the predictions by BERT, TOAD and OUR MODEL

| Text | Target | Label | BERT | TOAD | ADSC |
|---|---|---|---|---|---|
| Remember that story about a charismatic businessman who will lie to get into power from an obscure position? | Donald Trump | Against | Neutral | Against | Against |
| I know you are the best candidate. You are the only one who can make America great again | Donald Trump | Against | Favor | Favor | Against |
| I guess NBC does not like to hear the truth | Donald Trump | Favor | Neutral | Neutral | Neutral |

## 5 Conclusion

This paper proposes an adversarial distillation adaptation framework (ADSC) with sentiment contrastive learning to perform zero-shot stance detection. We employ an adversarial discriminative domain adaptation network to transfer stance knowledge from training data to unknown targets, use stance contrastive learning to increase the model's generalizability, introduce sentiment information to aid stance detection, and add knowledge distillation to prevent catastrophic forgetting during training. The results on two benchmark datasets show that our model achieves competitive performance on some unseen targets. In future work, we will introduce some domain knowledge to improve the performance of the stance detection model.

**Acknowledgements** The authors would like to thank the reviewers for their valuable suggestions and feedback.

**Author Contributions** YZ and CW performed experimental method design, wrote the draft and analyzed experimental results. JW provided suggestions and feedback. All authors have read and approved the final manuscript.

**Funding** This work is supported by a grant from Social and Science Foundation of Liaoning Province (No. L20BTQ008).

**Data Availability** We use two public datasets, WT-WT and SEM16. The WT-WT data are available at https://github.com/jefferyYu/UMT/tree/master/data and the SEM16 data are available at http://www.saifmohammad.com/WebPages/StanceDataset.htm.

## Declarations

**Conflict of interest** The authors declare no competing interests.



## References

1. Küçük, D., Can, F.: Stance detection: a survey. ACM Comput. Surv.Comput. Surv. **53**(1), 1–37 (2020)
2. Augenstein, I., Rocktäschel, T., Vlachos, A., Bontcheva, K.: Stance detection with bidirectional conditional encoding. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 876–885 (2016)
3. Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., Cherry, C.: SemEval-2016 task 6: Detecting stance in tweets. In: Proceedings of the 10th International Workshop on Semantic Evaluation, pp. 31–41 (2016)
4. Lai, M., Cignarella, A.T., Hernandez-Farias, D.I., Bosco, C., Patti, V., Rosso, P.: Multilingual stance detection in social media political debates. Comput. Speech Lang.. Speech Lang. **63**, 1–27 (2020)
5. Kumar, S., Carley, K.M.: Tree lstms with convolution units to predict stance and rumor veracity in social media conversations. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5047–5058 (2019)
6. Wang, Z., Wang, Q., Lv, C., Cao, X., Fu, G.: Unseen target stance detection with adversarial domain generalization. In: Proceedings of the International Joint Conference on Neural Networks, pp. 1–8 (2020)
7. Allaway, E., Srikanth, M.: Zero-shot stance detection: a dataset and model using generalized topic representations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 8913–8931 (2020)
8. Liu, R., Lin, Z., Tan, Y., Wang, W.: Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP, pp. 3152–3157 (2021)
9. Luo, Y., Liu, Z., Shi, Y., Li, S.Z., Zhang, Y.: Exploiting sentiment and common sense for zero-shot stance detection. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 7112–7123 (2022)
10. Allaway, E., Srikanth, M., McKeown, K.: Adversarial learning for zero-shot stance detection on social media. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4756–4767 (2021)
11. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2962–2971 (2017)
12. Liang, B., Chen, Z., Gui, L., He, Y., Yang, M., Xu, R.: Zero-shot stance detection via contrastive learning. In: Proceedings of the ACM Web Conference 2022, pp. 2738–2747 (2022)
13. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186 (2019)
14. Liang, B., Fu, Y., Gui, L., Yang, M., Du, J., He, Y., Xu, R.: Target-adaptive graph for cross-target stance detection. In: Proceedings of the Web Conference 2021, pp. 3453–3464 (2021)
15. Wei, P., Mao, W.: Modeling transferable topics for cross-target stance detection. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1173–1176 (2019)
16. Xu, C., Paris, C., Nepal, S., Sparks, R.: Cross-target stance classification with self-attention networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 778–783 (2018)
17. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al.: Generative adversarial networks. arXiv preprint arXiv:1406.2661 (2014)
18. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Lempitsky, V.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**, 1–35 (2016)
19. Zhou, J., Tian, J., Wang, R., Wu, Y., Xiao, W., He, L.: Sentix: a sentiment-aware pre-trained model for cross-domain sentiment

analysis. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 568–579 (2020)
20. Liang, B., Zhu, Q., Li, X., Yang, M., Gui, L., He, Y., Xu, R.: JointCL: a joint contrastive learning framework for zero-shot stance detection. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 81–91 (2022)
21. Ryu, M., Lee, G., Lee, K.: Knowledge distillation for BERT unsupervised domain adaptation. Knowl. Inf. Syst.. Inf. Syst. **64**(11), 3113–3128 (2022)
22. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
23. Conforti, C., Berndt, J., Pilehvar, M.T., Giannitsarou, C., Toxvaerd, F., Collier, N.: Will-they-won't-they: a very large dataset for stance detection on Twitter. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1715–1724 (2020)
24. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
25. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)