

SSR: Utilizing Simplified Stance Reasoning Process for Robust Stance Detection

Jianhua Yuan¹, Yanyan Zhao¹, Yanyue Lu¹, Bing Qin^{1, 2}

¹Harbin Institute of Technology, Harbin 150001, China

²Pengcheng Laboratory, Shenzhen 518066, China

{jhyuan, yyzhao, yyly, qinb}@ir.hit.edu.cn

Abstract

Dataset bias in stance detection tasks allows models to achieve superior performance without using targets (Kaushal et al., 2021). Most existing debiasing methods are task-agnostic, which fail to utilize task knowledge to better discriminate between genuine and bias features. Motivated by how humans tackle stance detection tasks, we propose to incorporate the stance reasoning process as task knowledge to assist in learning genuine features and reducing reliance on bias features. The full stance reasoning process usually involves identifying the span of the mentioned target and corresponding opinion expressions, such fine-grained annotations are hard and expensive to obtain. To alleviate this, we simplify the stance reasoning process to relax the granularity of annotations from token-level to sentence-level, where labels for sub-tasks can be easily inferred from existing resources. We further implement those sub-tasks by maximizing mutual information between the texts and the opinioned targets¹. To evaluate whether stance detection models truly understand the task from various aspects, we collect and construct a series of new test sets. Our proposed model achieves better performance than previous task-agnostic debiasing methods on most of those new test sets while maintaining comparable performances to existing stance detection models.

1 Introduction

The task of stance detection aims to predict the stance of the text towards the given target. It is crucial for various downstream tasks including fact verification, rumor detection, etc. It has a wide application in analyzing political opinions and product reviews. Existing works usually treat this task as a text pair classification problem and many design target-tweet interaction structures (Augenstein et al., 2016) to learn target-aware stance representations. However, Kaushal et al. (2021) have shown

¹refers to targets that a given tweet expresses opinions on.


Tweet: Hilarity of the day: Hillary said she went 'above and beyond' in transparency. Really? What about the 30k deleted emails? #SemST 	
Given Target: Hillary Clinton	Gold Stance: Against
Target 1: Hillary Clinton	✓ Pred. stance: Against
Target 2: Feminist Movement	✗ Pred. stance: Against
Target 3: Atheism	✗ Pred. stance: Against

Figure 1: An example illustrating that BERT model does not change predictions based on the target.

that those models (Du et al., 2017; Devlin et al., 2019) can achieve superior performances only using the tweet. Those end-to-end stance detection models treat the stance reasoning process as a black box and are prone to rely on bias features in the dataset instead of learning the underlying task. For instance, in Figure 1, the BERT model still predicts *Against* even when the target is changed to an unrelated target like *Atheism*. Meanwhile, common stance detection models perform poorly on out-of-distribution datasets (Kaushal et al., 2021) and unseen targets, which calls for debiasing stance detection models to get rid of spurious correlations in the datasets.

While Kaushal et al. (2021) made the first attempt to reveal dataset bias in stance detection, mitigating bias in other natural language understanding (NLU) tasks has been extensively explored. The key challenge in debiasing is how to discriminate between genuine and bias features. One line of work (Clark et al., 2019; Utama et al., 2020a,b) implicitly hypothesized that features, learnt by small models or by large models at earlier steps, could potentially be bias features. In addition, others (Kaushik et al., 2021; Kaushal et al., 2021; Yang et al., 2021) tried data augmentation to break spurious correlation in the training data and treated features learnt on the augmented data as genuine features. Another line of work (Tu et al., 2020) adopted multi-task learning with auxiliary tasks,

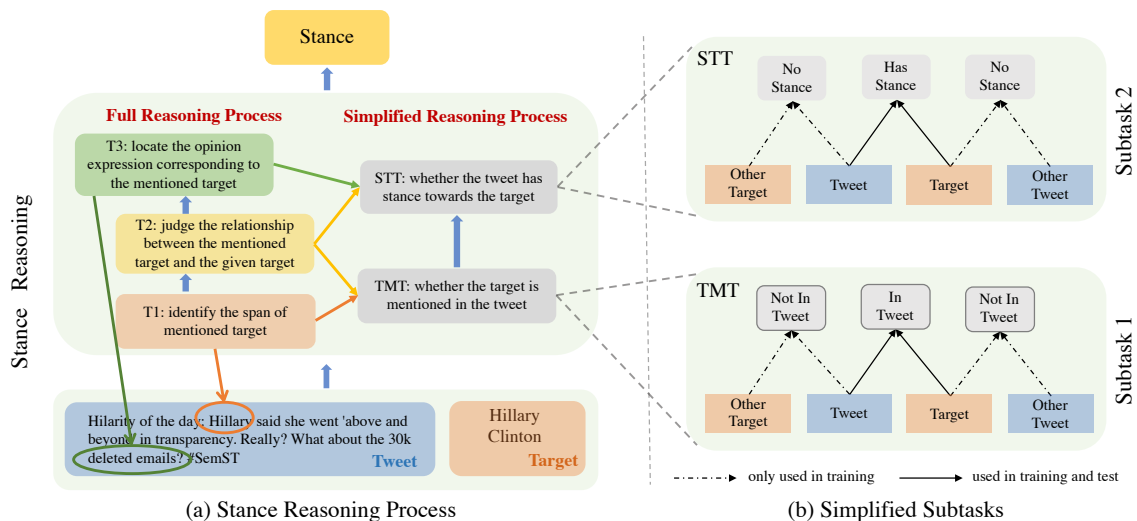


Figure 2: (a) An illustration of how humans perform stance reasoning on a given <target, tweet> pair and our simplified stance reasoning process. (b) We implement the simplified subtasks by maximizing the mutual information between the tweet and the opinioned target. During training, one positive example and two negative examples are constructed based on a given <target, tweet> pair for each subtask.

where features shared by multiple tasks are seen as genuine features. While these methods achieved superior debiasing performances, most of them neglected to explicitly leverage task knowledge to help discriminate between genuine and bias features.

In contrast, Dua et al. (2020) introduced manual annotations of intermediate reasoning steps and employed multi-task learning to combat dataset bias in question answering. Motivated by how humans perform stance reasoning processing in Figure 2, we follow this line of work and make the first attempt to incorporate stance reasoning process to mitigate dataset bias. Specifically, we consider the following reasoning steps: (T1) identifying the span of the mentioned target, (T2) judging the relationship between the mentioned target and the given target, and (T3) locating the opinion expression corresponding to the mentioned target. However, due to the informality of tweet texts, such fine-grained annotations for those reasoning steps are difficult and expensive to acquire.

To alleviate this, we seek to simplify the above reasoning process into two easier sub-tasks: 1) **TMT**, classifying whether the given target is mentioned in the tweet, as a simplification of T1 and T2. 2) **STT**, determining whether the tweet expresses any stance towards the given target, as an easy version of T2 and T3. The simplified subtasks only require sentence-level labels instead of token-level ones. More importantly, the labels for

these two sub-tasks can be easily inferred *without additional annotations*. We enhance these two sub-tasks by maximizing mutual information between the tweets and opinioned targets.

To help thoroughly evaluate whether models understand the stance detection task, we further collect and construct 6 new test sets. Those new test sets will assess whether stance detection models alter predictions based on the target, whether they overfit to shortcut features, whether they understand implicit mention of targets, and whether they can handle negations in the target part.

To summarize, our contributions are three folds:

- We make the first attempt to incorporate stance reasoning process to mitigate dataset bias in stance detection, where the labels for intermediate steps can be easily acquired without further annotations.
- We construct 6 test sets² to facilitate evaluation of whether stance detection models understand the task from various aspects.
- The proposed approach outperforms existing debiasing methods on 4/6 new test sets while maintaining comparable performances to common stance detection models on in-domain datasets.

²<https://github.com/Surpriseshelf/StanceSSR>

2 Approach

In this section, we present the **SSR** model that employs simplified stance reasoning process to combat dataset bias in stance detection. We first describe the basic text encoder that we use to encode tweet, target and target-tweet pairs. Then, we introduce two intermediate sub-tasks based on our observation of the stance reasoning process. After that, we elaborate on why and how we simplify the introduced two sub-tasks. These sub-tasks are implemented by maximizing the mutual information between the opinioned target and the tweet. Finally, we show how we combine these two sub-tasks for final stance predictions.

2.1 Text Encoder

We use $BERT_{base}$ as the text encoder. Given a text sequence $D = \{x_1, x_2, \dots, x_i, \dots, x_{|d}|\}$, where $|d|$ is the number of words in D , we transform its format to $X_d = \{[CLS], x_1, x_2, \dots, x_i, \dots, x_{|d}|, [SEP]\}$ to be compatible with the input of BERT. We use the hidden vector of [CLS] from the last transformer layer as the text representation for X_s . Thus, given a target-tweet pair, we could encode the tweet, target, and target-tweet pair as h_d , h_t and h_{pair} respectively.

2.2 Simplified Stance Reasoning Process

To deliberate the stance of a tweet towards the given target, one may identify the mention of the given target in the tweet, extract the span of corresponding opinions towards the given target in the tweets and capture the interactions between the target and opinions. However, due to the informal form of texts used on social network platforms, such intermediate annotations are difficult and expensive to acquire at scale for existing stance benchmarks. As a result, it is impractical to train stance detection model with these intermediate opinion and entity extraction, and opinion understanding sub-tasks.

To tackle these challenges, we seek to simplify those intermediate sub-tasks into easier ones that require only sentence-level instead of token-level annotations. Specifically, we consider the following two binary classification sub-tasks³:

- **TMT**: whether a target T is mentioned in a tweet D . It is designed to make the model

³We describe details of the acquisition of labels for these tasks in Appendix A.4

aware of the (both explicit and implicit) existence of target in tweet. This can be seen a simplified version of boundary detection of the target in the tweet. Intuitively, this could help restrain stance detection models from assigning stance to non-mentioned entities.

- **STT**: whether a tweet D expresses stance towards a target T . It requires the model to distinguish between the *None* stance and other stances.

These two sub-tasks cannot be solved by only using the tweets, thus preventing stance detection models from relying on spurious features in tweets. For instance, in Figure 2, the TMT sub-task will discourage stance models from predicting *Against* when the target is *Feminist Movement* as it does not appear in the tweet.

2.3 Mutual Information Maximization

Our key intuition behind introducing intermediate tasks is to strengthen the interaction between the opinioned target and the tweet. To achieve this, we implement two sub-tasks through maximizing mutual information (**MIMax**) between the tweets and opinioned targets. Motivated by (Hjelm et al., 2018; Tian et al., 2019; Yeh and Chen, 2019), we estimate the lower bound of mutual information between two random variables X and Y using Jensen-Shannon divergence (JS), which is implemented using the binary cross-entropy (BCE) loss:

$$\begin{aligned} MI(X, Y) &\geq E_P[\log(g(x, y))] \\ &+ \frac{1}{2}E_N[1 - \log(g(x, \bar{y}))] \\ &+ \frac{1}{2}E_N[1 - \log(g(\bar{x}, y))] \end{aligned} \quad (1)$$

where E_P and E_N refer to expectations over positive and negative samples respectively, and g is the discriminator function that outputs a real number modeled by a neural network. And (x, \bar{y}) and (\bar{x}, y) are negative samples sampled from the product of marginals. The discriminator function g is a bi-linear function defined as follows:

$$g(x, y) = x^T \mathbf{W} y$$

where W is a learnable scoring parameter.

2.3.1 Sub-task1: TMT

In TMT, as tweets are expected to carry the information of their opinioned targets, we choose to

maximize the averaged MI between the representation of the tweet and the representation of the target appearing in the tweet. Specifically, a positive example is obtained if the target appears in the tweet. Negative examples, on the other hand, are constructed by replacing the current target with a new target that does not appear in the tweet, and by replacing the current tweet with a new tweet that does not contain the current target respectively.

Following Equation 1, the objective for the sub-task TMT is formulated as follows:

$$\begin{aligned} \mathcal{L}_{TMT}(x_t, x_d, \bar{x}_t, \bar{x}_d) = & E_P[\log(g(x_t, x_d))] \\ & + \frac{1}{2} E_N[1 - \log(g(\bar{x}_t, x_d))] \\ & + \frac{1}{2} E_N[1 - \log(g(x_t, \bar{x}_d))] \end{aligned} \quad (2)$$

where x_d is a tweet and x_t is a target that is referred to in x_d , \bar{x}_t is another target that is not mentioned in x_d , and \bar{x}_d is another tweet that does not mention x_t .

2.3.2 Sub-task2: STT

Different from TMT, STT aims to uncover whether the tweet expresses any stance towards the given target. If the target is not mentioned in the tweet or the tweet does not express any polarized opinion towards the target, the label for STT will be *No*. Specifically, a positive example is obtained if the target expresses *Favor* or *Against* stance towards the target. Negative examples are constructed by replacing the current target with a new target that the tweet has no opinion on and by replacing the current tweet with another tweet that does not express any stance towards the given target respectively.

Following Equation 2, the objective for the sub-task TMT is formulated as follows:

$$\begin{aligned} \mathcal{L}_{STT}(x_t, x_d, \tilde{x}_t, \tilde{x}_d) = & E_P[\log(g(x_t, x_d))] \\ & + \frac{1}{2} E_N[1 - \log(g(x_t, \tilde{x}_d))] \\ & + \frac{1}{2} E_N[1 - \log(g(\tilde{x}_t, x_d))] \end{aligned} \quad (3)$$

where x_t is a target and x_d is a tweet that expresses opinion on x_t , \tilde{x}_t is another target and x_d holds no stance on \tilde{x}_t , and \tilde{x}_d is a tweet that does not expresses opinion on x_t .

2.4 Stance Classification

We feed the concatenation of the given <target, tweet> pair into BERT encoder to learn a target-aware stance representation \mathbf{h}_{pair} . We also feed

Target	#Total	#Train	#Test
Atheism	733	513	220
Climate Change	564	395	169
Feminist Movement	949	664	285
Hillary Clinton	984	689	295
Abortion	933	653	280
All	4163	2914	1249

Table 1: Statistics of SemEval2016 Task 6 Subtask A.

New Test Sets	Number
Tweet_only Failed (TOF)	319
PMI	403
Opinion Towards (OT)	425
Donald Trump (DT)	707
Target Replaced (Replaced)	3978
Target Negated (Negated)	1249

Table 2: Statistics of collected and constructed test sets.

the given pair into two sub-task and obtain feature representation \mathbf{h}_{tmt} and \mathbf{h}_{stt} respectively. We further concatenate \mathbf{h}_{pair} , \mathbf{h}_{tmt} and \mathbf{h}_{stt} as \mathbf{h}_{final} , and feed \mathbf{h}_{final} into a simple feed-forward network for stance classification:

$$\mathbf{h}_{sc} = \mathbf{h}_{pair} \oplus \mathbf{h}_{tmt} \oplus \mathbf{h}_{stt} \quad (4)$$

$$y_s = \text{softmax}(\mathbf{W}_{sc2} \sigma(\mathbf{W}_{sc1} \mathbf{h}_{sc})) \quad (5)$$

where \mathbf{W}_{sc2} and \mathbf{W}_{sc1} are learnable weight matrices.

And the classifier is trained with the following cross-entropy loss of stance classification:

$$\mathcal{L}_{SC} = -\frac{1}{N_s} \sum_{i=1}^{N_s} \sum_{j=1}^{N_o} \hat{y}_s^i(j) \log y_s^i(j) \quad (6)$$

where N_s is the number of training instances and N_o is the number of different stance labels.

The final objective for our multiple sub-task learning method becomes:

$$\mathcal{L}_{msl} = \mathcal{L}_{SC} + \lambda_1 \mathcal{L}_{TMT} + \lambda_2 \mathcal{L}_{STT} \quad (7)$$

where λ_1 and λ_2 are hyper-parameters to control impacts of two sub-tasks respectively.

3 Experimental Setups

3.1 Datasets

To thoroughly assess whether a model understands the stance detection task, we collect and construct a series of test sets to assess the understanding of the stance detection task from various perspectives.

3.1.1 In-domain Dataset

We train our model on the dataset from SemEval 2016 Task 6 Sub-task A. The dataset is comprised of 4163 English tweets and each is assigned with a target and a manually annotated stance label towards that target. There are a total of five targets in Sub-task A. The detailed statistics of this dataset are shown in Table 1. We use the official train/test split. We randomly select 15% of samples from the training data as the validation set.

3.1.2 New Test Sets

To test whether the trained stance detection models overly rely on bias features in the training set, we collect three subsets where bias features from the original test set may not hold. Additionally, we use the data from SemeEval2016 Task 6 sub-task B as an out-of-domain test set to evaluate the generalization ability of stance detection models. Moreover, we also construct two adversarial sets to test the sensitivity of stance detection models when changing targets.

In-distribution Hard Set. Those hard sets are collected from the original test set. 1) Tweet Only Failed (**TOF**). This subset is collected from the original test set where three BERT_{nt} (see Sec.3.3) models with different seeds all fail. The filtered subset is to assess whether models could succeed in cases where only using tweets may not make correct predictions. 2) **PMI**. This subset is filtered by removing instances containing features with the top 200 point-wise mutual information scores of each stance from the original test set. It could help test whether models perform well on long-tailed features. 3) Opinion Towards (**OT**). We keep instances from the original test set with indirect mention and no mention of targets according to additional annotations provided by (Mohammad et al., 2016). This could be used to diagnose whether given models are aware of implicit mentions of targets in the tweet part.

Adversarial Test Sets. Those hard sets are adversarially constructed based on the original test set. 1) Target Replaced (**Replaced**). To obtain this set, we replace the original target with other targets from the SemEval2016 training dataset. To ensure that the replaced targets are not mentioned in the tweet, we utilize ConceptNet (Speer et al., 2017) to enhance pattern matching. After replacement, we label the new instances with *None*. 2) Target Negated (**Negated**). It is constructed by negating

the targets and keeping the tweets unchanged. The stance labels flip accordingly. For entity-like targets, we add ‘NOT’ in front of the original targets. For ‘Atheism’, we add the negated target ‘Theism’. For claim-like targets, we add ‘not’ into the sentence to negate the claim.

Out-of-distribution Hard Set. Donald Trump (**DT**). It comes from SemEval2016 task B with an unseen target ‘Donald Trump’ in task A, which is used to test the generalization ability of stance detection models.

3.2 Evaluation Metrics

Similar to previous work, we adopt the macro-average of F1-score of Favor and Against across targets as the evaluation metric (see Appendix A.2). We report the averaged results of 5 random seeds for all experiments. For details of implementation, please see Appendix A.3.

3.3 Baselines

Stance Detection Models Methods on Stance detection: 1) BERT_{wt} (*wt*: with target) and BERT_{nt} (*nt*: no target), which are based on BERT_{base}. BERT_{nt} only uses the tweet as input while BERT_{wt} takes the <target, tweet> pair as inputs. 2) TAN (Du et al., 2017), which is an LSTM based model that incorporates target-specific attention. We adopt the BERT version of TAN (Kaushal et al., 2021). 3) Stancy (Popat et al., 2019), which is a BERT based model with an additional cosine similarity score between the tweet representation and the target-tweet pair representation.

Debiasing Models Apart from sophisticated stance detection models, we also compare with recent debiasing methods for natural language inference and fact verification tasks. These methods are: 1) Product-of-expert (PoE) (Clark et al., 2019), which combines the learned probabilities of a bias-only model and a full model using PoE. 2) LMH (Clark et al., 2019), which explicitly determines how much to trust the bias in PoE and employs an entropy-based regularization to encourage the bias component to be non-uniform. 3) E2E PoE (Karimi Mahabadi et al., 2020), which proposes an end-to-end training version of PoE. 4) Conf-Reg (Utama et al., 2020a), which utilizes signals from bias models to scale the confidence of models’ predictions.

Models	Original	TOF	PMI	OT
BERT _{nt}	67.84	73.9	50.8	38.15
BERT _{wt}	69.21	85.56	51.66	42.62
TAN	68.44	86.51	59.31	43.88
Stancy	70.3	95.34	59.78	44.67
PoE	68.96	94.06	55.15	41.53
LMH	64.88	82.69	47.46	33.81
E2E-PoE	61.68	84.88	53.93	39.08
Conf-Reg	70.24	90.45	61.26	41.16
SSR (ours)	71.36	96.47	56.08	46.58

Table 3: Results on the SemEval test dataset and three subsets. The average of F_{Favor} and $F_{Against}$ is adopted as the evaluation metric. For comparison with other stance detection models on each target, please refer to Table 6 in Appendix.

4 Results and Analysis

4.1 Main Results

4.1.1 On Original Test Set and its Subsets

As shown in Table 3, while existing stance models achieve remarkable progress on the original test set, models that consider targets (BERT_{wt}, Stancy) only slightly outperform models that do not consider targets (BERT_{nt}, TAN). This indicates that existing dataset bias allows stance detection model to achieve good results solely relying on tweets. While debiasing models tend to down-weight bias features and samples, useful features for the stance detection task are inevitably influenced, leading to the performance drop of PoE, LMH, and E2E-PoE on original test set. On the contrary, our model improves BERT_{wt} by 2.15% on the original test sets, showing that utilization of stance reasoning sub-tasks could facilitate learning better features for stance detection.

Though TOF is constructed by selecting samples from the test set where three BERT_{nt} models with random seeds fail, a new BERT_{nt} model with another seed still reaches 73.9%. This implies that *failure* examples may not transfer across the same models with different initialization. Nevertheless, models explicitly considering targets outperform those not considering the target by a large margin.

By comparing SSR and BERT_{wt}, we find that over 70% improvement of SSR over BERT_{wt} on original test sets comes from the PMI subset. This shows that by considering the simplified stance reasoning process, SSR is less likely to rely on bias features.

On OT set, we can see that BERT_{nt} performs poorly as it does not consider target and thus is not capable of capturing implicit mention of targets. In

Model	DT	Replaced	Negated
BERT _{nt}	11.42	32	17.59
BERT _{wt}	28.12	47.08	17.38
TAN	27.09	35.55	<u>20.05</u>
Stancy	32.34	49.5	19.67
PoE	19.42	46.7	19.54
LMH	<u>36.76</u>	34.97	25.36
E2E-PoE	33.72	33.01	19.98
Conf-Reg	36.27	34.17	19.51
SSR (ours)	37.66	59.7	17.6

Table 4: Performances of different models on DT, Replaced, Negated test sets. Note that, results on DT are not directly comparable to those reported in (Allaway et al., 2021; Liang et al., 2022) as they used 4,163 pairs for training while we only use 2,914 pairs.

contrast, SSR explicitly models whether the target is mentioned in the tweet and achieves the best performance.

4.1.2 On New Test Sets

To test whether stance detection models understand the task instead of solely fitting the dataset, we present results of several representative stance detection models in Table 4.

On DT set, we note that BERT_{nt} performs the worst on the out-of-domain dataset. BERT_{wt} and TAN perform slightly better. Our SSR model performs better than other models, which suggests that leveraging intermediate tasks could help learn more transferable features for cross-target generation. For debiasing methods, PoE and E2E-PoE work well on in-distribution hard sets while performing worse on out-of-distribution test sets, and vice versa.

As the Replaced set is to test awareness to change of the target, BERT_{wt}, Stancy and SSR that explicitly capture interactions between the target and the tweet, outperform other models by a large margin.

On the Negated set, while LMH model achieved the highest performance of 25.36%, many other models only tangle around 20%, showing that those models can not tackle with negations of semantics in the targets. Our model performs worse on this set as negated targets and original targets will have the same labels in both sub-tasks.

4.2 Ablation Study

As shown in Figure 5, both the TMT and STT sub-tasks contribute to the performance on the original test set. This indicates that appropriate subtasks could help learn better features for the main task.

Models	Orig.	TOF	PMI	OT
SSR	71.36	96.47	56.08	46.58
w/o tmt	70.10	91.91	62.21	44.64
w/o stt	70.65	92.47	63.4	47.77
SSR w/o MIMax	70.36	88.88	66.33	45.79
w/o tmt	68.97	93.7	63.58	41.01
w/o stt	68.99	91.08	63.39	45.25
		DT	Replaced	Negated
SSR		37.66	59.7	17.6
w/o tmt		37.61	57.37	19.58
w/o stt		32.64	48.95	18.24
SSR w/o MIMax		26.01	49.35	18.76
w/o tmt		34.83	44.77	20.22
w/o stt		29.5	49.01	18.71

Table 5: Ablation study on the proposed SSR model.

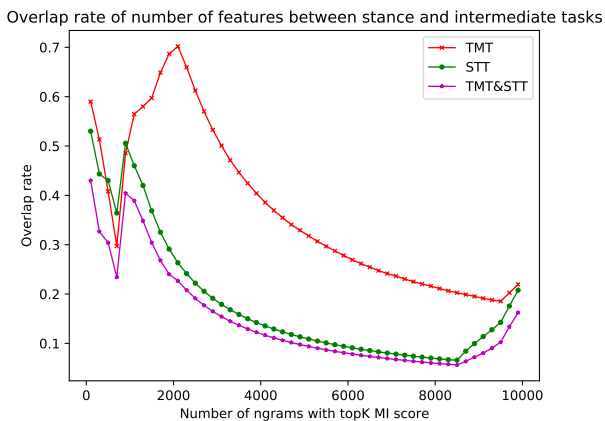


Figure 3: Overlap of top features of different tasks according to mutual information.

Adding MIMax further improves the performances by 1%.

While on *Negated* set, both tasks fail to handle the negation and lead to worse performance. As the TMT task is capable of understanding implicit mention of targets in the tweet, it is more useful than the STT task on *OT* set. Since TMT and STT can detect whether the target is changed, they both contribute to performance gain on the *Replaced* set. Generally, MIMax helps the SSR model learn more genuine stance features and improves performances on the *TOF* and *DT* set, where mutual connections between tweets and opinion targets are crucial, and either TMT or STT alone is not enough to capture such connections.

4.3 Analysis

Intermediate reasoning tasks help reduce reliance on shortcut features. After obtaining synthetic labels for these two subtasks, we conduct qualitative analysis to show that subtasks could potentially regularize the features used by the stance detection task. As mutual information (MI) scores

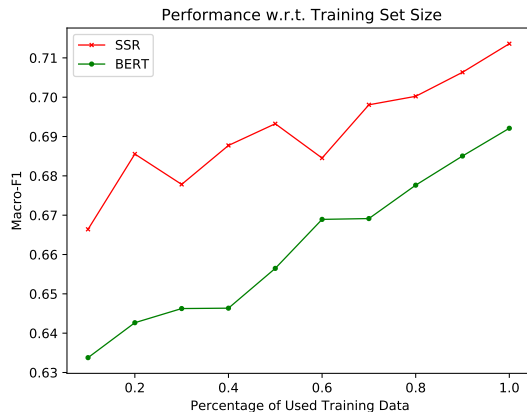


Figure 4: Performance of different models w.r.t the training size.

between features and categories serve as a good indicator of the importance of features, we first collect bag-of-words features for N-grams in the training set and sort those N-grams according to their MI scores with the labels in corresponding tasks. In Figure 3, we can see that both subtasks have less than 60% overlap of the top 1000 features, which means many important features used by the stance detection task are not useful for these subtasks. This could be seen as implicitly re-weighting features used by the stance detection task based on intermediate stance reasoning subtasks, which hopefully will promote learning genuine features instead of bias ones.

SSR requires less data for training. Previously, (Dua et al., 2020) found that collecting intermediate annotations for up to 10% training data can improve the performance of a reading comprehension model by 4-5%. Here, as annotations of intermediate tasks could be automatically acquired without human annotations, we would explore whether incorporating intermediate tasks could help reduce the demands for training data. In Figure 4, we show that when only 10% of training data is available, our model outperforms BERT_{wt} model by 3.3% on macro-F1. In contrast, BERT_{wt} model has to use 60% training data to reach comparable performance. Thus, utilizing intermediate stance reasoning tasks could help reduce the demand of training data and improve the performance in low-resource stance detection scenarios.

4.4 Case Study

We compare our model with two BERT baselines to illustrate the target awareness provided by in-


Tweet: Hilarity of the day: Hillary said she went 'above and beyond' in transparency. Really? What about the 30k deleted emails? #SemST 			
Target	BERT _{wt}	BERT _{nt}	SSR (TMT: Yes, STT: Yes)
Hillary Clinton	✓ Against	✓ Against	✓ Against (TMT: No, STT: No)
Atheism	✗ Against	✗ Against	✓ None

Figure 5: Case study on target-awareness of different models.

intermediate subtasks. In Figure 5, we can see that when the target is *Atheism*, which is not mentioned in the tweet at all, both BERT baselines falsely predict the stance as *Against*, which may owe to the existence of the shortcut word ‘email’. In contrast, as the TMT task tells the model that the given target is not mentioned in the tweet, and the STT task shows that the tweet has no stance towards the given target, our model could correctly predict the stance label as *None*.

5 Related Work

5.1 Stance Detection

Recently, detection stances in texts from social media platforms have attracted a lot of attention. Compared to traditional sentiment analysis tasks, stance detection is more challenging as the given target may not appear in the text. Inferring the relations between the given target and the opinioned entity usually requires rich world knowledge. In this paper, we focus on the single target stance detection on tweets where each tweet is given one target. Various methods (Augenstein et al., 2016; Du et al., 2017; Popat et al., 2019) have been proposed to model the inter-dependency between the target and tweet. However, Kaushal et al. (2021) recently noted that current stance detection models relied heavily on bias features in existing stance detection datasets, which makes it necessary to develop stance detection specific debiasing methods to combat these biases.

In this work, we also study the problem of dataset bias in stance detection and propose a novel method incorporating simplified stance reasoning process. Furthermore, we collect and construct 6 new test sets to facilitate evaluation of whether stance detection models truly understand the task.

5.2 Debiasing Dataset Bias in NLP

Recently, the community has shown that neural models can achieve good performances by leveraging dataset bias in various natural language understanding tasks, e.g. NLI (Gururangan et al., 2018; Poliak et al., 2018), question answering (Mudrakarta et al., 2018), VQA (Agrawal et al., 2018), machine translation, summarization, fact verification (Schuster et al., 2019) and sentiment analysis (Wang and Culotta, 2020, 2021; Kaushal et al., 2021; Yan et al., 2021; Yang et al., 2021). Such phenomenon mainly attributes that neural models tend to utilize superficial features in the dataset instead of understanding the semantics of underlying tasks, e.g. NLI models usually exploits word overlap and syntactic patterns, and even only use features from the hypothesis for final predictions.

To mitigate dataset bias, one line of work (Clark et al., 2019; Karimi Mahabadi et al., 2020; Utama et al., 2020a; Ghaddar et al., 2021) implicitly treated features learned by a smaller model or common model at earlier stages/layers as potential bias features and down-weighted these features in the main model. Another line of work adopted data augmentation strategies to weaken the spurious correlations between bias features and final labels. Beside, Tu et al. (2020) employed a multi-task learning based-approach by introducing auxiliary tasks like paraphrase identification to avoid overfitting to bias features. However, most previous methods are task-agnostic, which failed to utilize the task knowledge of the underlying task. To this end, Dua et al. (2020); Shao et al. (2021) introduced manual annotations of intermediate reasoning steps to combat dataset biases.

In this work, we make the first attempt to incorporate stance reasoning steps to combat dataset biases in stance detection. Our work differs in the following ways (1) we introduce two simplified sub-tasks whose labels can be automatically inferred instead of manual annotations (2) we novelly implement two sub-tasks via maximizing the mutual information between the tweet and the opinioned target.

6 Conclusion

In this paper, we propose to utilize the stance reasoning process as task knowledge to guide the discrimination between genuine and bias stance features. To alleviate demands for token-level intermediate annotations, we simplify the stance reasoning

process where labels for proposed subtasks can be automatically inferred without additional annotations. To evaluate whether stance detection models understand the task from various aspects, we collect and construct 6 new test sets. Empirical results show that our model outperforms task-agnostic debiasing methods on 4/6 new test sets while maintaining comparable performances to existing stance detection models on in-domain datasets.

Acknowledgements

We thank the anonymous reviewers for their insightful comments and suggestions. We thank Haichao Zhu for his valuable suggestions on re-framing the paper. This work was supported by the National Key RD Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 62176078. Yanyan Zhao is the corresponding author.

References

- Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Emily Allaway, Malavika Srikanth, and Kathleen McKeown. 2021. 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*, pages 4756–4767, Online. Association for Computational Linguistics.
- Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.
- Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Jiachen Du, Lin Gui, Ruifeng Xu, Yunqing Xia, and Xuan Wang. 2020. Commonsense knowledge enhanced memory network for stance classification. *IEEE Intelligent Systems*, 35(4):102–109.
- Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3988–3994. ijcai.org.
- Dheeru Dua, Sameer Singh, and Matt Gardner. 2020. Benefits of intermediate annotations in reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5627–5634, Online. Association for Computational Linguistics.
- Abbas Ghaddar, Phillippe Langlais, Mehdi Rezagholizadeh, and Ahmad Rashid. 2021. End-to-end self-debiasing framework for robust NLU training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1923–1929, Online. Association for Computational Linguistics.
- Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of NAACL2018, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.
- Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2021. Cross-domain label-adaptive stance detection. *CoRR*, abs/2104.07467.
- R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.
- Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.
- Ayush Kaushal, Avirup Saha, and Niloy Ganguly. 2021. tWT-WT: A dataset to assert the role of target entities for detecting stance of tweets. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3879–3889, Online. Association for Computational Linguistics.

- Divyansh Kaushik, Amrith Setlur, Eduard H Hovy, and Zachary Chase Lipton. 2021. [Explaining the efficacy of counterfactually augmented data](#). In *International Conference on Learning Representations*.
- Yingjie Li and Cornelia Caragea. 2019. [Multi-task stance detection with sentiment and stance lexicons](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6299–6305, Hong Kong, China. Association for Computational Linguistics.
- Yingjie Li and Cornelia Caragea. 2021. [Target-aware data augmentation for stance detection](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1850–1860, Online. Association for Computational Linguistics.
- Bin Liang, Zixiao Chen, Lin Gui, Yulan He, Min Yang, and Ruifeng Xu. 2022. [Zero-shot stance detection via contrastive learning](#). In *Proceedings of the ACM Web Conference 2022, WWW '22*, page 2738–2747, New York, NY, USA. Association for Computing Machinery.
- Matthew Matero, Nikita Soni, Niranjana Balasubramanian, and H. Andrew Schwartz. 2021. [Melt: Message-level transformer with masked document representations as pre-training for stance detection](#). *CoRR*, abs/2109.08113.
- Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. [SemEval-2016 task 6: Detecting stance in tweets](#). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.
- Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. [Did the model understand the question?](#) In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1896–1906, Melbourne, Australia. Association for Computational Linguistics.
- Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. [Hypothesis only baselines in natural language inference](#). In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.
- Kashyap Papat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2019. [STANCY: Stance classification based on consistency cues](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6413–6418, Hong Kong, China. Association for Computational Linguistics.
- Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2020. [Stance detection benchmark: How robust is your stance detection?](#) *CoRR*, abs/2001.01565.
- Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. [Towards debiasing fact verification models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425, Hong Kong, China. Association for Computational Linguistics.
- Zhihong Shao, Lifeng Shang, Qun Liu, and Minlie Huang. 2021. [A mutual information maximization approach for the spurious solution problem in weakly supervised question answering](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4111–4124, Online. Association for Computational Linguistics.
- Robert Speer, Joshua Chin, and Catherine Havasi. 2017. [Conceptnet 5.5: An open multilingual graph of general knowledge](#). In *Thirty-First AAAI Conference on Artificial Intelligence*.
- Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. [Contrastive multiview coding](#). *arXiv preprint arXiv:1906.05849*.
- Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. [An empirical study on robustness to spurious correlations using pre-trained language models](#). *Transactions of the Association for Computational Linguistics*, 8:621–633.
- Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020a. [Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online. Association for Computational Linguistics.
- Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020b. [Towards debiasing NLU models from unknown biases](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online. Association for Computational Linguistics.
- Zhao Wang and Aron Culotta. 2020. [Identifying spurious correlations for robust text classification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3431–3440, Online. Association for Computational Linguistics.

- Zhao Wang and Aron Culotta. 2021. [Robustness to spurious correlations in text classification via automatically generated counterfactuals](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14024–14031.
- Hanqi Yan, Lin Gui, Gabriele Pergola, and Yulan He. 2021. [Position bias mitigation: A knowledge-aware graph model for emotion cause extraction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3364–3375, Online. Association for Computational Linguistics.
- Linyi Yang, Jiazheng Li, Padraig Cunningham, Yue Zhang, Barry Smyth, and Ruihai Dong. 2021. [Exploring the efficacy of automatically generated counterfactuals for sentiment analysis](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 306–316, Online. Association for Computational Linguistics.
- Yi-Ting Yeh and Yun-Nung Chen. 2019. [QAInfomax: Learning robust question answering system by mutual information maximization](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3370–3375, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 In-domain Dataset

We train our model on the dataset from SemEval 2016 Task 6 Sub-task A. The dataset is comprised of 4163 English tweets crawled from Twitter⁴ and each is assigned with a target and a manually annotated stance label towards that target. There are a total of five targets in Sub-task A, which are *Atheism* (AT), *Climate Change is a real Concern* (CC), *Feminist Movement* (FM), *Hillary Clinton* (HC), and *Legalization of Abortion* (LA).

A.2 Evaluation Metrics

Similar to previous work, we adopt the macro-average of F1-score of *Favor* and *Against* across targets as the evaluation metric. We report the averaged results of 5 random seeds for all experiments.

Similar to previous work, we adopt the macro-average of F1-score across targets as the evaluation metric, which is calculated as:

$$F_{Favor} = \frac{2P_{Favor}R_{Favor}}{P_{Favor} + R_{Favor}} \quad (8)$$

$$F_{Against} = \frac{2P_{Against}R_{Against}}{P_{Against} + R_{Against}} \quad (9)$$

$$F_{macro} = \frac{2(F_{Against} + F_{Favor})}{2} \quad (10)$$

where P and R are precision and recall. Then the average of F_{Favor} and $F_{Against}$ is calculated as the final metrics F_{macro} . Note that the final metrics do not disregard the *None* class. By taking the average F-score for only *Favor* and *Against* classes, we treat *None* as a class that is not of interest. We report the averaged results of 5 random seeds for all experiments.

A.3 Implementation Details

We adopt the uncased version of BERT_{base} for all our experiments. We fine-tune BERT_{base} model with Adam optimizer. The dropout rate is set to 0.5 for all parameters. The learning rate is chosen from $\{1, 2, 3, 4, 5\} \times 10^{-5}$ and batch size for training is set to 8. We choose λ_1 and λ_2 from $[0.1, 1.0]$ with a step size of 0.1. Final choices of all hyperparameters are selected according to performance on the validation set. λ_1 and λ_2 are set to 0.1 and 0.2 respectively. The learning rate is set to 5×10^{-5} .

⁴<https://www.twitter.com>

A.4 Details for label acquisition of sub-tasks

To acquire labels for the TMT task, we seek to expand the original targets with external resources. Specifically, we take two structured knowledge bases, ConceptNet and WikiData. ConceptNet mainly contains commonsense knowledge, WikiData mainly contains social knowledge. They complement each other and supply rich target related knowledge for target understanding. Each of the targets in the dataset is treated as a key for searching for the most related commonsense knowledge from them. In this way, we augment each target with external knowledge base. Then given a tweet, we use the augmented targets to performance exact string matching, if any of the augmented targets locates in the tweet, the TMT label would be *Yes*. Otherwise, the TMT label is set to *No*.

To obtain labels for STT task, given a <target, tweet> pair and its corresponding stance label, if the stance is *Favor* or *Against*, then the STT label would be *Yes*. Otherwise, the STT label is set to *No*.

As shown above, the label acquisition for TMT and STT subtasks are easy and straightforward, which does not involve additional manual annotation. Thus, it is practical to incorporate the above subtasks as parts of the simplified stance reasoning process.

A.5 Detailed results on original dataset

In Table 6, we compare our model with recent stance detection models. We can see that our SSR model performs comparably to existing stance detection models that utilized external stance detection datasets, lexicons and tweet corpora.

A.6 Additional Analysis

Bias is ubiquitous in multiple tasks on the same dataset. Similarly, we apply BERT_{nt} and BERT_{wt} model on the TMT and STT tasks respectively. As shown in in Table 7, BERT_{nt} could achieve relative performance even when no target information is utilized for those target-aware tasks, suggesting the ubiquitous of bias on different tasks in this dataset. Moreover, We can see that for the TMT task, the discrepancy between BERT_{nt} and BERT_{wt} is 8.7%, significantly larger than that of the STT task and stance detection task. This shows that TMT requires more interactions between the target and the tweet, which may account for the better performances on *OT* and *Replaced* hard sets

Models	AT	CC	FM	HC	LA	Overall
AT-JSS-Lex (Li and Caragea, 2019)	69.22	<u>59.18</u>	<u>61.49</u>	68.33	68.41	72.33
CKEMN (Du et al., 2020)	<u>62.69</u>	53.52	61.25	64.19	64.19	69.74
MT-DNN _{SDL} (Schiller et al., 2020)	-	-	-	-	-	70.18
MT-DNN _{MDL} (Schiller et al., 2020)	-	-	-	-	-	71.81
MoLE (Hardalov et al., 2021)	-	-	-	-	-	<u>72.08</u>
ASDA (Li and Caragea, 2021)	74.93	-	56.43	<u>67.01</u>	61.60	-
MeLT (Matero et al., 2021)	66	71	63	67	66	-
TAN	69.72	44.32	53.26	55.79	62.75	68.44
Stancy	66.08	54.67	59.91	62.0	58.69	70.3
BERT _{nt}	63.96	48.88	53.97	60.59	60.24	67.84
BERT _{wt}	65.36	44.91	48.25	65.09	54.33	69.21
SSR	63.17	56.88	59.68	62.69	57.33	71.36
SSR(-MIMax)	<u>70.09</u>	54.0	56.68	65.64	62.04	70.36

Table 6: Results on the SemEval dataset. The macro average of F_{Favor} and $F_{Against}$ is adopted as the evaluation metric. The results in bold are the best in corresponding columns. The underlined results are the second best in corresponding columns.

Task	BERT _{nt}	BERT _{wt}	Δ
TMT	83.39	92.09	8.70
STT	80.97	83.85	2.88
Stance	70.1	72.4	2.3

Table 7: Accuracy of BERT_{nt} and BERT_{wt} on different tasks.

when adding the TMT task instead of the STT task. Though two auxiliary sub-tasks have their own dataset bias to some extent, combing them with the main task still boosts the performance of the stance detection on both original and newly constructed hard sets. This supports our motivation of leveraging intermediate tasks to learn robust features for the main task.