Adaptive Meta-learner via Gradient Similarity for Few-shot Text Classification

Tianyi Lei[1], **Honghui Hu**[1], **Qiaoyang Luo**[2], **Dezhong Peng**[1], **Xu Wang**[1*]
[1]College of Computer Science, Sichuan University
[2]The University of Adelaide
{leity828,wangxu.scu}@gmail.com

Abstract

Few-shot text classification aims to classify the text under the few-shot scenario. Most of the previous methods adopt optimization-based meta learning to obtain task distribution. However, due to the neglect of matching between the few amount of samples and complicated models, as well as the distinction between useful and useless task features, these methods suffer from the overfitting issue. To address this issue, we propose a novel Adaptive Meta-learner via Gradient Similarity (AMGS) method to improve the model generalization ability to a new task. Specifically, the proposed AMGS alleviates the overfitting based on two aspects: (i) acquiring the potential semantic representation of samples and improving model generalization through the self-supervised auxiliary task in the inner loop, (ii) leveraging the adaptive meta-learner via gradient similarity to add constraints on the gradient obtained by base-learner in the outer loop. Moreover, we make a systematic analysis of the influence of regularization on the entire framework. Experimental results on several benchmarks demonstrate that the proposed AMGS consistently improves few-shot text classification performance compared with the state-of-the-art optimization-based meta-learning approaches. The code is available at: https://github.com/Tianyi-Lei.

1 Introduction

As a fundamental task of few-shot learning (Fei-Fei et al., 2006) in natural language processing theme, few-shot text classification (Yu et al., 2018; Geng et al., 2019) requires a model to predict categories that are not seen in training. Meta learning (Schmidhuber, 1987; Thrun and Pratt, 2012), which plays a crucial role in general few-shot learning, aims to improve generalization ability and fast adaptation ability of the learner through modelling the distribution of tasks. To adapt few-shot tasks, typical supervised meta-learning methods (Vinyals et al., 2016; Finn et al., 2017) model task distributions from a few support tasks over meta-training episodes. Subsequently, numerous methods based on meta-learning (Bao et al., 2020; Luo et al., 2021; Han et al., 2021) are proposed to solve few-shot text classification problem.

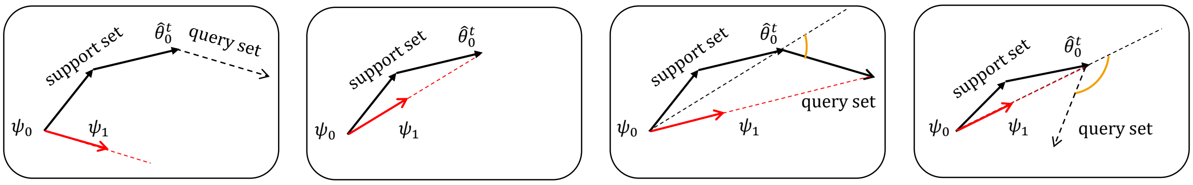
Within the meta-learning frameworks, Bao et al. (2020) trains an attention-based model to enhance the text representation of distributional signature, Luo et al. (2021) leverages label-semantic augmentation to help BERT compensate for the ambiguity of the class definition caused by the limited data, and Han et al. (2021) strengthens the generalization of a model using an adversarial domain adaptation network. However, these methods are similar to the traditional meta-learning methods, neglecting the overfitting problem caused by utilizing the few number of data in the complicated models under the meta-learning frameworks.

To address the above problem in few-shot text classification, several methods are proposed based on a principle *i.e.*, obtaining more task-distribution can ameliorate the risk of over-fitting to the training task distribution. Bansal et al. (2020) alleviates overfitting through joint training of self-supervised tasks and classification tasks in pre-trained models. We also follow this method and use a self-supervised Mask Token Prediction (MTP) task in meta training phase. Unfortunately, the increased task distribution generated by this joint training is not always positive for meta-training.

In order to further overcome the overfitting challenge in meta-training, we propose the adaptive meta-learner via gradient similarity based on another principle *i.e.*, distinguishing positive and negative features by feature selection of deep model can enhance generalization by alleviating overfitting. In optimization-based meta-learning framework, the gradient contains all the information transmitted from the inner-learner to the outer-

[*] Corresponding author

Proceedings of the 29th International Conference on Computational Linguistics, pages 4873–4882
October 12–17, 2022.



(a) MAML(First order) (b) REPTILE (c) AMGS (ours, scheme 1) (d) AMGS (ours, scheme 2)

Figure 1: Diagram of the comparison of different methods for gradient direction optimization. The black arrow, black dotted arrow, red arrow and red dashed line denote the actual update of base-learner, the update direction of based-learner, the actual update of meta-learner and the update direction of meta-learner, respectively. **(a)** MAML(First order): A set of initial parameters ψ_0 is updated in the direction of the red arrow, *i.e.*, the gradient of query set loss, which is calculated at $\hat{\theta}$ after t-step updates. Note that, since the gradient calculation of MAML contains the Hessian matrix, it is hard to represent in the figure, we use the First Order MAML (FOMAML) to replace MAML. **(b)** REPTILE: ψ_0 is updated along the red arrow pointing to the t-step optimization solution. **(c)** and **(d)** are different schemes of our proposed adaptive meta-learner, which distinguish the positive gradient cosine similarity (scheme 1) and negative gradient cosine similarity (scheme 2). If the gradient direction obtained on the query set is similar to the gradient direction of the sum of the t updates (black dashed line), ψ_0 is updated in the direction of the sum of all gradients. While if their gradient directions are opposite, we remove the gradient obtained from the query set.

learner, including “features” mentioned in above principle. Thus, the gradient obtained by base-learner can be regarded as the "features". To compare with other training strategies for meta-learner, we plot Figure 1. Other strategies often adopt all the gradient obtained by the base-learner without distinction. They also may consume enormous computing resources for calculating the Hessian matrix, sacrifice the stability and accuracy in order to adopt the first-order algorithm, or discard the query set in the training Batch in order to simplify the calculation. By contrast, our method only needs to distinguish the gradient similarity between the gradient of the loss on the query set and the current gradient of the base-learner during the meta-training process. Subsequently, we utilize the corresponding gradient of its loss to help meta-learner quickly adapt to the optimization space. Such method selects the more useful gradients for meta-learner in current training batch. In addition, it neither increases the computational complexity nor causes waste of text information in the same training episode.

According to the above principles, we propose a novel Adaptive Meta-learner via Gradient Similarity (AMGS) algorithm based on optimization-based meta learning scheme. We firstly construct the self-supervised task called Mask Token Prediction (MTP) for the base-learner in the inner loop. Such approach can generate the extension of the task distribution from unlabeled text and constraint the gradient updating of primary classification task to increase the robustness of the model. Moreover, in the outer loop, we utilize the adaptive meta-learner to improve the utilization of the task features from the inner loop. As Figure (1) shows, our strategy can more efficiently leverage query set samples in a training episode, which optimizes the scope of gradient optimization. Therefore, the adaptive meta-learner directly accomplish additional amelioration of overfitting.

The contributions of this paper are summarized as follows: (1) We construct an optimization-based meta-learning framework named AMGS and elaborately design a meta-training algorithm to effectively tackle the overfitting issue in few-shot text classification based on two different principles. (2) We propose an adaptive meta-learner that selects the positive gradients and removes the negative gradients to improve the generalization ability of the model on the few-shot task (3) Experimental results demonstrate that the proposed AMGS outperforms the state-of-the-art optimization-based meta-learning models.

2 Related Work

Few-shot text classification via meta learning Few-shot learning is an application of meta-learning. In most meta-learning frameworks, the strategies can be divided into two categories: metric-based meta-learning and optimization-based meta-learning. Prototypical Network (Snell et al., 2017), Induction Network (Geng et al., 2019) and Relation Network (Sung et al., 2018) are dedicated to construct a metric space between classes and samples. In the optimization-based meta-learning methods, most of them consist of an inner (or base) algorithm and an outer (or meta) algorithm. Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017) and Reptile (Nichol et al., 2018) are examples of such optimization-based algorithms. LEOPARD (Bansal et al., 2020) achieves a good performance on diverse classification tasks by using BERT (Devlin et al., 2019). Meanwhile, recent work (Bao et al., 2020) proposes a meta-learning-based method by using distributional signatures for few-shot text classification. More recently, LaSAML (Luo et al., 2021) uses label information for few-shot text classification. Another one (Han et al., 2021) applies a domain discriminator into a meta-learning framework. However, these algorithms suffer from overfitting caused by the imbalance between the few data and the deep model in the few-shot setting. By contrast, our proposed AMGS which expands the task distribution in the inner loop and distinguishes the positive and negative gradient in the outer loop can address this issue indirectly and directly.

Auxiliary learning In general, auxiliary learning can assist the main task to learn more accurately and quickly in deep learning (Wang et al., 2022, 2019b), especially in the multi-task learning field. SSL-Reg (Zhou et al., 2021) builds a regularizer of the loss of self-supervised learning tasks to improve performance on text classification. Besides constructing a task, external auxiliary data can also be introduced into the model to obtain more latent information (Zhang et al., 2018).

Similarly, auxiliary tasks are valuable to adapt the meta-learning scheme. MAXL (Liu et al., 2019) adopts a self-supervised learning scheme to generate auxiliary labels, improving the generalization ability of the primary task in gradient update. Furthermore, self-supervised auxiliary tasks can promote fast adaptation during the testing phase (Chi et al., 2021). Hybrid SMLMT (Bansal et al., 2020) creates a specific self-supervised auxiliary task for multi-task learning. Similar to these auxiliary tasks, our auxiliary task MTP is self-supervised to generate richer task distribution during meta-training.

3 Methods

In this section, we first introduce the preliminaries for few-shot classification (Vinyals et al., 2016). Next, we describe Adaptive Meta-learner via Gradient Similarity (AMGS) method in detail.

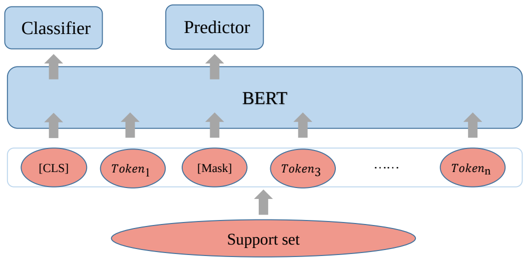


Figure 2: The main framework of the proposed AMGS.

3.1 Overview

Problem setup The setting of few-shot classification often includes *training episode* and *testing episode*. Suppose we have examples with labels from the classes $y_{train}$of training episode and need to predict the labels of examples from unseen but related classes y_{test} of testing episode. The training classes and testing classes are mutually exclusive, denotes as $y_{train} \cap y_{test} = \Phi$. To create a training episode, we need to build a set of *N-way K-shot* tasks. For each task, we sample N classes, $k+q$ examples of each class randomly. The $N \times k$ $\{x_s, y_s\}$ pairs including examples and corresponding labels constitute the support set, while the $N \times q$ labeled examples $\{x_q, y_q\}$ are known as the query set. It is the same way to create a support set in testing episode, but leverage the unlabeled examples $\{y_q\}$ to create the query set in testing. By repeating the above procedure, we can obtain enough training and testing episodes, so that we can use them in meta-training and meta-testing respectively. In short, such setting requires the model to have the ability to generalize from seen classes in training episodes to unseen classes in testing episodes.

Model architecture BERT (Devlin et al., 2019) performs well in the conventional text classification, thus we leverage it as text encoder in our proposed AMGS framework to explore the problem of few-shot text classification. As shown in Figure (2), the model architecture consists of the BERT encoder, a classifier and a predictor. The model performs the primary task (i.e, classification) and the auxiliary task (i.e., token prediction) simultaneously, which constitutes multi-task learning. In training period, the support set are used to obtain the BERT encoding and label prediction in the primary branch. While in the auxiliary task branch, BERT encoder updates parameters through the self-supervised task without labels.

For convenience to explain in following section, we define parameters of the total network $\theta = \{\theta_b, \theta_c^{pri}, \theta_p^{aux}\}$, where θ_b denotes the shared weights of BERT encoding, θ_c^{pri} represents classification weights for the primary task, and θ_p^{aux} is prediction weights for the auxiliary task. Concretely, the primary-branch weights and the auxiliary-branch weights are respectively denoted as $\theta^{pri} = \{\theta_b, \theta_c^{pri}\}$, and $\theta^{aux} = \{\theta_b, \theta_p^{aux}\}$.

Self-supervised Mask Token Prediction task As mentioned above, we leverage BERT as text encoder. Considering that our self-supervised auxiliary task should be adapted to BERT, we adopt the Mask Token Prediction **(MTP)** task used in BERT pre-training stage (also known as MLM). MTP randomly masks the tokens in the sentences according to the specified ratio. These masked tokens are fed into BERT to be predicted by putting the final hidden vector corresponding to the masked token into the output softmax over the vocabulary. The original strategy of MTP in BERT set 15% probability for each token to replace with the [MASK] token 80% of the time, a random token 10% of the time, and the unchanged token 10% of the time. In some cases, if the text used to construct MTP task is very short, none of the tokens in this text would be masked with high probability. This will cause the effect of MTP to fail in the downstream task.

Therefore, we improve the replacement probability of each token being masked to 30% instead of 15%. Meanwhile, the masking rating of replacing the target token with a random token and an unchanged token are both set to 0%, because random and unchanged replacement both occur for 3% of all tokens, which leads to instability. This change helps the model acquire the new task distribution more stably. The masked strategy is explored in experiment demonstrated in Appendix A.

3.2 Adaptive Meta-learner via Gradient Similarity (AMGS)

The optimization-based meta-learning methods (Finn et al., 2017; Nichol et al., 2018) learn an appropriate initial parameters by meta learner, achieving encouraging performance. However, these methods ignore the overfitting issue in the few-shot learning. Considering that the direction of gradients could be used to distinguish the positive and negative gradients, we propose AMGS framework with explicit regularization. The training procedure of AMGS is decomposed into two steps: (i) The base-learner collects gradient for adaptive meta-learner, which utilizes the multi-task network to learn primary and auxiliary tasks together on support set. Then it collects the gradient of the loss on the query set by leveraging the supervised primary task. (ii) The adaptive meta-learner via gradient similarity distinguishes the positive and negative gradient obtained by the first stage, then updates the parameters of the total meta-network by meta-learner. By completing two training steps, our method ensures that the meta-learner learns the more balanced initial parameters and makes the loss of new tasks decrease faster.

3.2.1 Collecting gradient for adaptive meta-learner

This subsection describes that how the base-learner collects gradient for adaptive meta-learner. We leverage the self-supervised MTP task to acquire a more abundant task distribution and improve the base-learner robustness. In addition, as mentioned above, we build multi-task learning by using the MTP to limit the training of classification tasks. In other words, the constraint on the loss of the primary task has been enforced via the auxiliary task. This limitation prevents the base-learner to obtain extra characteristics of each training task to alleviate overfitting. Formally, we compute the total loss of the multi-task network as follows:

$$\mathcal{L}_{total} = (1-\rho)\mathcal{L}_{pri} + \rho\mathcal{L}_{aux}, \tag{1}$$

where $\mathcal{L}_{total}$, $\mathcal{L}_{pri}$, $\mathcal{L}_{aux}$ and ρ represent the total loss, primary classification loss $\mathcal{L}_{pri}(x, y; \theta^{pri})$, auxiliary prediction task loss $\mathcal{L}_{aux}(x; \theta^{aux})$, and the contribution of the auxiliary task, respectively. x and y denote training texts and their labels. We use cross entropy loss to implement both text classification and the masked token prediction. In our experiments, we set $\rho = 10^{-3}$. The sensitivity study is shown in Appendix B.

When training on the tasks T_i in the *support set*, the total loss Eq.(1) after one or a few gradient updates can be defined as follows:

$$\hat{\theta} = \theta - \alpha\nabla_{\theta}\mathcal{L}_{T_i}^{total}(x_s, y_s; \theta), \tag{2}$$

where x_s and y_s are texts and corresponding labels in the support set. α is the adaptation learning rate. By Eq.(1) and Eq.(2), we can obtain more semantic

Algorithm 1 Training procedure of AMGS

```
Input: learning rate $\alpha$, $\beta$, texts and corresponding labels $x, y$
Initialize $\Psi = \theta = \{\theta_b, \theta_c^{pri}, \theta_p^{aux}\}$ with BERT
while not converged do
    Sample batch of tasks $T_i \sim p(T)$
    Sample support set ($x_s$, $y_s$), query set ($x_q$, $y_q$)
    for all $T_i$ do
        Compute adapted parameters with gradient descents: $\hat{\theta} = \theta - \alpha \nabla_\theta \mathcal{L}_{T_i}^{total}(x_s, y_s; \theta)$
        Compute the gradients of primary task on $\hat{\theta}$: $(\theta_b, \theta_c^{pri}) = (\theta_b, \theta_c^{pri}) - \alpha \nabla_\theta \mathcal{L}_{T_i}^{pri}(x_q, y_q; \hat{\theta}_b, \hat{\theta}_c^{pri})$
    end for
    if $cos(\nabla_\theta \mathcal{L}_{T_i}^{total}(x_s, y_s; \theta), \nabla_\theta \mathcal{L}_{T_i}^{pri}(x_q, y_q; \hat{\theta}_b, \hat{\theta}_c^{pri})) \geq 0$ then
        Update: $\hat{\Psi} \leftarrow \Psi - \beta \nabla_\Psi \sum_{T_i \sim p(T)} (\mathcal{L}_{T_i}^{total}(x_s, y_s; \theta) + \mathcal{L}_{T_i}^{pri}(x_q, y_q; \hat{\theta}^{pri}))$
    else
        Update: $\hat{\Psi} \leftarrow \Psi - \beta \nabla_\Psi \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}^{total}(x_s, y_s; \theta)$
    end if
end while
```

representation to apply explicit regularization to the primary loss. In general, the query set is used for testing and inference, while it contains rich task distribution which can be applied to meta-learn. We argue that the query set can be used to fine-tune and enhance the gradient learned by the base-learner through the multi-task network. In the step, we accomplish the collection of gradient of the parameters $\{\theta_b, \theta_c^{pri}\}$ on the query set. Finally, the objective can be defined as follows:

$$\underset{\theta_b, \theta_c^{pri}}{\arg\min} \, \mathcal{L}_{T_i}^{pri}(x_q, y_q; \hat{\theta}^{pri}), \tag{3}$$

where x_q and y_q are texts and corresponding labels in the query set.

3.2.2 Upgrade meta-learner with AMGS

This stage is mainly about updating meta-learner. Following previous work (Du et al., 2018), we leverage the gradient cosine similarity to measure whether the gradients obtained on query set are positive or negative. Based on Eq.(2) and Eq.(3), we get the gradient cosine similarity by calculating $cos(\nabla_\theta \mathcal{L}_{T_i}^{total}(x_s, y_s; \theta), \nabla_\theta \mathcal{L}_{T_i}^{pri}(x_q, y_q; \hat{\theta}_b, \hat{\theta}_c^{pri}))$. If the value of $cos(\cdot)$ is non-negative, such gradient is regarded as the **positive gradient**, which means the query set at this batch is beneficial to enhance generalization of the model. Therefore we obtain the gradient of its loss to perform gradient enhancement on the meta-learner. For this situation, the meta-objective can be written as:

$$\begin{aligned} \underset{\theta_b, \theta_c^{pri}, \theta_p^{aux}}{\arg\min} \sum_{T_i \sim p(T)} & (\mathcal{L}_{T_i}^{total}(x_s, y_s; \theta) \\ & + \mathcal{L}_{T_i}^{pri}(x_q, y_q; \hat{\theta}^{pri})). \end{aligned} \tag{4}$$

On the contrary, if $cos(\cdot)$ is negative, such gradient is considered as the **negative gradient**. We remove this query set loss to ensure that the model is not negatively affected, so the meta-objective is:

$$\underset{\theta_b, \theta_c^{pri}, \theta_p^{aux}}{\arg\min} \sum_{T_i \sim p(T)} (\mathcal{L}_{T_i}^{total}(x_s, y_s; \theta)). \tag{5}$$

According to above training procedure, our proposed meta-objective can distinguish the positive to use and the negative to by adaptive meta-learner, which can automatically filter appropriate regularization to limit the gradient optimization. This step reduces effective model capacity, hence it effectively alleviates overfitting and improves the generalization ability of the model. The full training procedure is demonstrated in the Algorithm 1.

3.2.3 Meta testing

The model parameters have been learned in meta training phase, and fine-tuned in the meta-learning testing phase for downstream tasks. MTP can continue to participate in the fine-tuning phase in order to help the primary classification adapt to the unseen classes for the new tasks quickly. From the perspective of test-time fast adaptation (Chi et al., 2021), our auxiliary task boosts the fast gradient descent of the loss function of the primary task in the testing procedure.

Methods		**HuffPost**		**Banking77**				**Clinc150** (cross domain)			
		5-way		10-way		15-way		10-way		15-way	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Metric	BERT+PROTO	40.59	53.48	63.05	78.60	59.18	74.12	57.43	72.90	52.31	66.06
	BERT+RELATION	40.80	51.87	63.88	73.48	56.29	64.57	54.65	60.09	46.54	58.83
	BERT+INDUCT	39.96	50.79	48.72	64.32	49.45	55.27	46.52	57.65	41.72	49.98
Optimization	BERT+MAML	41.03	57.13	59.21	85.55	55.69	81.48	60.14	80.24	55.00	65.20
	BERT+REPTILE	40.80	58.96	58.36	82.81	56.69	81.14	59.89	81.23	53.32	63.04
	BERT+R2D2	40.78	61.98	70.45	87.80	63.46	**85.65**	62.72	87.13	57.61	80.76
	DS+R2D2	41.34	62.48	59.33	83.71	53.37	78.96	55.56	78.76	53.41	79.69
	MLADA+R2D2	41.55	59.82	61.69	80.81	55.63	74.77	65.28	85.45	51.76	77.77
	BERT+AMGS (OURS)	**43.47**	**63.40**	**71.41**	**88.81**	**63.62**	84.93	**69.19**	**88.26**	**62.12**	**84.13**

Table 1: Results of 5-way 1-shot and 5-way 5-shot on HuffPost headlines dataset, 10-way 1-shot, 10-way 5-shot, 15-way 1-shot and 15-way 5-shot on Banking77 and Clinc150 datasets (cross domain) by using our proposed method and all baselines.

4 Experiments

4.1 Datasets

We use three datasets to evaluate the performance in experiment.

HuffPost headlines includes 36900 news headlines among 41 classes, which contains less information than other datasets. In order to complete a fair comparison test, we divide each training, validation, and testing set into 20, 5, and 16 classes by following the setting of Bao et al. (2020). **Banking77** (Casanueva et al., 2020) consists of 13083 fine-grained intents and 77 classes. As for the setting of data distribution and *N-way K-shot* classification tasks, we assign 30, 15, and 32 classes fixedly for training, validation, and testing set. **Clinc150** (Wang et al., 2019a) is a cross-domain intent classification dataset with 150 classes in 10 domains. It provides 22500 examples that cover 150 intents from 10 domains without overlap among classes. We allocate for each training, validation, and testing with 4, 1, 5 domains, respectively.

4.2 Baselines

In order to evaluate our AMGS, we compare with three metric-based methods and five optimization-based algorithms for few-shot text classification.

Proto (Snell et al., 2017) provides a metric-based method to learn the class vector by computing distances to prototype representations of each class. **Induct** (Geng et al., 2019) learns a generalized class-wise representation by leveraging the dynamic routing algorithm. **Relation** (Sung et al., 2018) compares the class vector and the query feature through a relation-based meta-learner. **MAML** (Finn et al., 2017) is one of the most typical optimization-based meta-learning algorithms, which trains a favorable initial point for the base learner by utilizing the meta learning that learns among tasks. **Reptile** (Nichol et al., 2018) is a first-order variant method of MAML. It achieves that the speed of calculation is greatly improved and the complexity is reduced, while the accuracy is almost the same as MAML. The base learner used by Ridge Regression Differentiable Discriminator **(R2D2)** (Bertinetto et al., 2019) is ridge regression based on linear regression model. The amount of calculation is related to the sample size of the task, which is conducive to the learning of the meta learner. **DS** (Bao et al., 2020) shows the best performance by leveraging the model that builds an attention generator and a ridge regressor to enhance the representational power of distributional signature. **MLADA** (Han et al., 2021) uses the meta-learning adversarial domain adaptation network to improve the adaptation and new classes embedding generation by creating a domain discriminator.

4.3 Implementation details

$BERT_{base}$ is used as the text encoder of all baselines. Because DS and MLADA have special requirements for textual representation and feature extraction, forcibly using BERT as encoder will be counterproductive. Thus, we re-implement the pre-train fastText embeddings (Joulin et al., 2016) for those model, and follow other settings in the original papers (Bao et al., 2020; Han et al., 2021). For the sake of fairness, the classifiers of these two algorithms use R2D2, so we constructed a comparison item with BERT as encoder.

All parameters are optimized with Adam optimizer (Kingma and Ba, 2015). The initial learning rates α, β are separately set to $5e{-}5$ and $2e{-}5$, and we utilize 5 gradient updates for the base adaptation step. As for the *N-way K-shot* classification setting, all experiments use 25 examples for the query set. We randomly sample 100 training episodes, 100 validation episodes, and 1000 testing episodes per epoch and apply early stopping on validation for 20 epochs. We evaluate the performance of the model based on 5 different random seeds. All experiments are conducted on a GEFORCE RTX 3090 GPU.

4.4 Experimental results

The total results of experiments are reported in table 1. By observing these experimental results, we obtain the following conclusions:

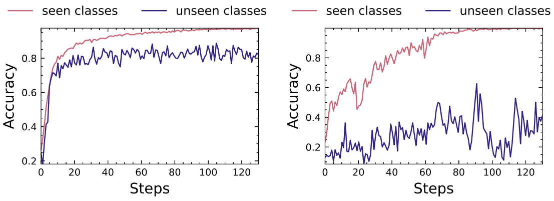
(1) Whether it is for texts with minimal semantics (Huffpost), fine-grained categorized (Banking77) or cross domains (Clinc150), our proposed method AMGS has an average improvement of 0.2-6.5% over the state-of-the-art model on both 1-shot and 5-shot classification. In particular, compared with our AMGS and MAML (Finn et al., 2017), Reptile (Nichol et al., 2018), we can draw the following observations from the Table 1: (i) Our proposed method achieves better performance on all tasks. In especial, in the 15-Way 5-shot task on Clinc150 dataset, our proposed method outperforms the best counterpart by 18.9%. (ii) MAML and Reptile perform better on fine-grained classification Banking77 dataset with more similar categories than on cross-domain Clinc150 dataset with less similar categories, and have a smaller gap with our AMGS. To verify that our AMGS perform better than MAML on alleviating overfitting, we plot their accuracy learning curves in Figure 3. In the figure, the training procudure of our AMGS is more stable than that of MAML from the beginning to the end. Besides, the gap between the accuracy of seen classes and unseen classes of our AMGS is less than that of MAML. These results are demonstrated that our AMGS can make model more stable in meta-training and more readily generalizeds to unseen classes by addressing the overfitting issue.

Methods	**Banking77**		**Clinc150**	
	15-way		15-way	
	1-shot	5-shot	1-shot	5-shot
AMGS w que	56.39	82.32	55.72	65.61
AMGS w sup	57.03	81.84	54.17	64.12
AMGS w que+sup	58.51	82.10	55.97	83.73
AMGS w our strategy	**63.62**	**84.93**	**62.12**	**84.13**

Table 2: Ablation study results for different strategies of meta-learner on Banking77 and Clinc150 (cross domain) datasets.

(2) With leveraging BERT as our text encoder, our method is better than all compared methods on Huffpost dataset. In Bao et al. (2020), it points out that BERT can better deal with highly contextual classification but not the keyword-based news classification, e.g., Huffpost dataset. Thus, "DS+R2D2" performs better on Huffpost than "BERT+R2D2", but worse on Banking77 and Clinc150. Nonetheless, our "BERT+AMGS" surpasses all BERT-based and non-BERT-based approaches on Huffpost dataset, which shows the superiority of our AMGS method. Furthermore, the performance of our model is increased by 2.1% on 1-shot classification and 0.9% on 5-shot classification when compared with BERT-based models.

Overall, the above observations point that AMGS can learn the commonalities and characteristics between few-shot task distribution well by mitigating overfitting, thereby obtaining a better initialized parameter for fast adaptation.



(a) AMGS (OURS)

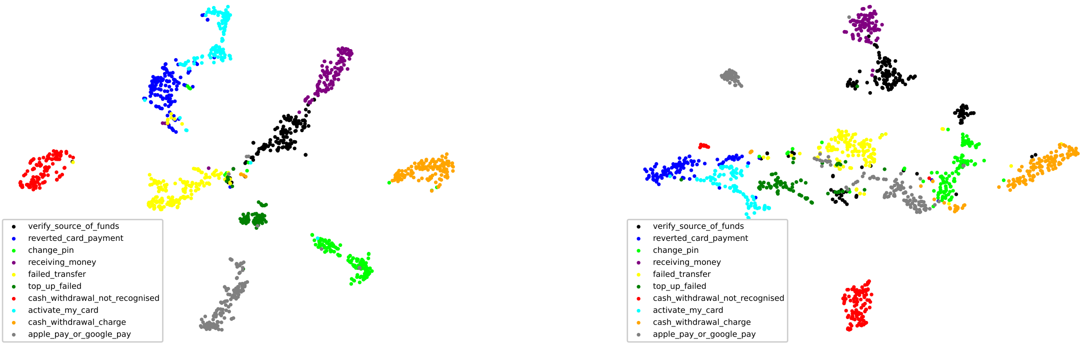
(b) MAML

Figure 3: Learning curves of AMGS (a) and MAML (b) on 15-way 5-shot task of the Banking77 dataset. We plot average accuracy from seen classes (red) and unseen classes (blue).

4.5 Ablation studies

In this section, we conduct several ablation experiments to verify the effectiveness of the adaptive meta-learner, MTP in the meta training phase, and MTP in the meta-testing fast adaptation phase.

The effectiveness of the adaptive meta-learner In this section, we further investigate the impact of



(a) AMGS (OURS)

(b) REPTILE

Figure 4: t-SNE visualization of the input representation for the query set of a testing episode (*N*=10, *K*=5, *L*=120) sampled from Banking77 dataset.

Methods	Banking77		Clinc150	
	15-way		15-way	
	1-shot	5-shot	1-shot	5-shot
AMGS w/o MTP	62.82	84.73	61.82	83.17
AMGS w/o MTP (testing)	63.26	84.32	61.97	84.01
AMGS w MTP	**63.62**	**84.93**	**62.12**	**84.13**

Table 3: Ablation study results of MTP in meta-training and meta-testing fast adaptation phase (testing) on Banking77 and Clinc150 (cross domain) datasets.

the different strategies for meta-learner. To compare with our strategy, we design three other comparison strategies. As shown in Table 2, "AMGS w que", "AMGS w sup", "AMGS w que+sup" respectively represent the meta-learner in AMGS only use the gradients of the query set, support set and both query and support set. None of these three strategies pay attention to distinguishing the positive or the negative of the gradients. Comparing our strategy with "AMGS w que+sup" strategy, we have improved significantly more on 1-shot task than on 5-shot task. From all the results, our adaptive meta-learner which filters the impact of the negative gradient achieves the better performance among these compared strategies.

The effectiveness of MTP in meta-training phase and in meta-testing fast adaptation phase As shown in Table 3, we first eliminate MTP in training stage. After losing a richer distribution of tasks, the performances of AMGS decrease by about 0.8%, which verifies the effectiveness of MTP in meta-training phase. Further, we explore MTP in meta-testing fast adaptation phase. The empirical results demonstrate that after joining the auxiliary task in meta-testing, the model performances have increased by about 0.5%. The testing auxiliary task makes the primary task more robust on the support set, and has some suppression effects on the occurrence of overfitting. All these results demonstrate that MTP task have a certain effect on Banking77 and Clinc150 datasets, but it can not significantly improve the experimental results.

4.6 Visualization

We visualize the results of the experiments to demonstrate that our model can generate a high-quality text representation for unseen classes.

T-SNE (Van der Maaten and Hinton, 2008) visualization illustrates the experimental results in Figure (4), we take out the generated sentence embedding layer before sending it to the classifier for visualization. Comparing Figure 4(a) and Figure 4(b), it is obvious that our method AMGS produces better separation than REPTILE, Especially for the categories represented by gray and lime, the sentence representations obtained by REPTILE are very similar, so that it is difficult to distinguish their categories. The above observations demonstrate the effectiveness of AMGS to generate a high-quality text representation for few-shot text classification.

5 Conclusion

In this paper, we present an Adaptive Meta-learner via Gradient Similarity (AMGS) framework for few-shot text classification. To be specific, we first leverage the self-supervised Mask Token Prediction (MTP) task to enrich the task distribution with the unlabeled text. Such approach can reduce the impact of overfitting caused by the mismatching between the few samples and the deep model.

Secondly, we construct an adaptive meta-learner via gradient similarity for the outer loop to distinguish the positive and negative gradient. Thus, the meta-learner alleviates overfitting by preventing the influence of negative features. Experimental results validate that our model achieves significant improvement on the few-shot text classification tasks by effectively alleviating the overfitting issue.

6 Acknowledge

This work is supported by the National Natural Science Foundation of China (Grant No. U19A2078), China Postdoctoral Science Foundation (Nos. 2021TQ0223, 2022M712236), Sichuan Science and Technology Planning Project (Grant Nos. 2022YFQ0014, 2022YFH0021, 2021YFS0390), and Fundamental Research Funds for the Central Universities (No. 2022SCU12081).

References

Trapit Bansal, Rishikesh Jha, Tsendsuren Munkhdalai, and Andrew McCallum. 2020. Self-supervised meta-learning for few-shot natural language classification tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 522–534.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot text classification with distributional signatures. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. 2019. Meta-learning with differentiable closed-form solvers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45.

Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. 2021. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9137–9146.

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.

Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. 2018. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*.

Li Fei-Fei, Rob Fergus, and Pietro Perona. 2006. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. 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 3904–3913.

Chengcheng Han, Zeqiu Fan, Dongxiang Zhang, Minghui Qiu, Ming Gao, and Aoying Zhou. 2021. Meta-learning adversarial domain adaptation network for few-shot text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1664–1673.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Shikun Liu, Andrew J Davison, and Edward Johns. 2019. Self-supervised generalisation with meta auxiliary learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 1679–1689.

Qiaoyang Luo, Lingqiao Liu, Yuhao Lin, and Wei Zhang. 2021. Don't miss the labels: Label-semantic augmented meta-learner for few-shot text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2773–2782.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999.

Jürgen Schmidhuber. 1987. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook.* Ph.D. thesis, Technische Universität München.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4080–4090.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

Sebastian Thrun and Lorien Pratt. 2012. *Learning to learn*. Springer Science & Business Media.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019a. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*.

Xu Wang, Peng Hu, Pei Liu, and Dezhong Peng. 2022. Deep semisupervised class- and correlation-collapsed cross-view learning. *IEEE Transactions on Cybernetics*, 52(3):1588–1601.

Xu Wang, Dezhong Peng, Peng Hu, and Yongsheng Sang. 2019b. Adversarial correlated autoencoder for unsupervised multi-view representation learning. *Knowledge-Based Systems*, 168:109–120.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215.

Yabin Zhang, Hui Tang, and Kui Jia. 2018. Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data. *CoRR*, abs/1807.10916.

Meng Zhou, Zechen Li, and Pengtao Xie. 2021. Self-supervised Regularization for Text Classification. *Transactions of the Association for Computational Linguistics*, 9:641–656.

A Ablation for different MTP masking strategies

In section 3.1, we mention that MTP adopts a different masking strategy from the one used in BERT pre-train stage. We explore the effect of different masking strategies in following ablation.

Masking prob.	**Masking strategy**			**Banking77**
	Mask	Same	Random	10-way 1-shot
15%	100%	0%	0%	73.38
	80%	10%	10%	72.06
30%	**100%**	**0%**	**0%**	**74.06**
	80%	10%	10%	72.20
45%	100%	0%	0%	72.56
	80%	10%	10%	72.42

Table 4: Ablation study results of different masking strategies on the validation episodes of Banking77.

As Table 3 shows, we explore the effect of different masking probabilities and strategies on Banking77. In the table, "Mask" means that we replace the token with [MASK] in MTP, "Same" means that we keep the target token unchanged and "Random" means that the token is replaced with the random token except itself. From the table, we can see that the masking strategy in BERT pre-training is not the best choice in the few-shot text classification. Therefore, in this paper, we attempt to alter the masking strategy which 100% changes the target token to [MASK].

B Sensitivity study on the trade-off parameter ρ

In order to set an appropriate value for the trade-off parameter of MTP mentioned in section 3.2.1, we study a sensitivity study for this hyper-parameter in 10-way 5-shot on Banking77 dataset.

trade-off ρ	0.9	0.5	10^{-1}	10^{-3}	10^{-5}	0
Accuracy	89.98	90.10	94.80	**95.60**	94.00	93.40

Table 5: Sensitivity study results of 10-way 5-shot on the validation episodes of Banking77.

The results of validation episodes have shown in Table 4. We explore a large scale trade-off ρ. demonstrating MTP has the greatest contribution when the trade-off ρ equals 0.001. Especially, ρ equals 0 means we remove the impact of MTP, which verifies the effectiveness of our MTP.