PGADA: Perturbation-Guided Adversarial Alignment for Few-shot Learning Under the Support-Query Shift

Siyang Jiang[†‡], Wei Ding[†], Hsi-Wen Chen[†], Ming-Syan Chen[†✉]

†Department of Electrical Engineering, National Taiwan University
‡School of Mathematics and Statistics, Huizhou University
{syjiang,wding,hwchen}@arbor.ee.ntu.edu.tw, mschen@ntu.edu.tw

Abstract. Few-shot learning methods aim to embed the data to a low-dimensional embedding space and then classify the unseen query data to the seen support set. While these works assume that the support set and the query set lie in the same embedding space, a distribution shift usually occurs between the support set and the query set, i.e., *the Support-Query Shift*, in the real world. Though optimal transportation has shown convincing results in aligning different distributions, we find that the small perturbations in the images would significantly misguide the optimal transportation and thus degrade the model performance. To relieve the misalignment, we first propose a novel adversarial data augmentation method, namely *Perturbation-Guided Adversarial Alignment (PGADA)*, which generates the *hard* examples in a self-supervised manner. In addition, we introduce *Regularized Optimal Transportation* to derive a smooth optimal transportation plan. Extensive experiments on three benchmark datasets manifest that our framework significantly outperforms the eleven state-of-the-art methods on three datasets. Our code is available at https://github.com/772922440/PGADA.

Keywords: Few-shot learning · Adversarial data augmentation · Optimal transportation

1 Introduction

Recently, deep learning models have celebrated success in several computer vision tasks [23,19,18], which require large amounts of labeled data. However, collecting sufficient labels to train the model parameters involves considerable human effort, which is unacceptable in practice [13]. Moreover, the labels of training data and testing data are usually disjoint, i.e., the labels in the testing phase are unseen in the training phase. In contrast, few-shot learning aims to learn the model parameter by a handful of training data (support sets) and adapt to the testing data (query sets) [32,12,2], effectively. In the testing phase, the model classifies the query set into the support sets according to the distance of their embeddings.

While these approaches extract rich contextual features from the images, the embeddings from the support set and the query set usually exhibit certain distribution shifts, i.e., *the Support-Query Shift* [2]. For example, the images are captured by various devices, e.g., smartphones and single-lens reflex cameras in different environments, e.g., foggy and high-luminance. Since the learned embeddings are located in different spaces, degrading the model performance, *optimal transportation* [9] has been proposed

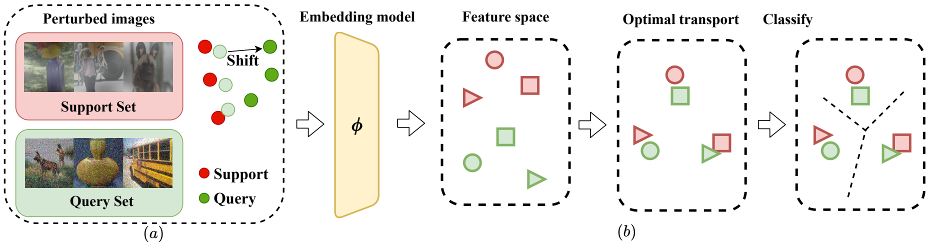

Fig. 1: (a) Illustration of the support-query shift in a 3-way 1-shot classification task, where the support set and the query set are embedded into different distributions. (b) First, we embed the support set and the query set via an embedding model ϕ into a feature space. Next, optimal transportation is employed to align the support set (red) and the query set (green). But, the small perturbations may misguide the transported results, leading to wrong predictions, i.e., classify the green circle to the red triangle.

to stage the embeddings from different domains into the same latent space. However, in this paper, we theoretically prove that optimal transport can be easily misguided by small perturbations in the images (as illustrated in Fig. 1).

Meanwhile, several training techniques [38,14] have been proposed to derive a more robust embedding model against the perturbations. On the one hand, data augmentation methods transform a single image [27] or combine multiple images [35,36] to create more training samples at pixel level. However, these methods cannot create new information which is not included in the given data [38]. On the other hand, adversarial training methods such as projected gradient descent (PGD) [25], AugGAN [17] are used to find the perturbed images to confuse the model, i.e., predicting an incorrect label, as the additional training samples. However, these methods usually require numerous iterations to generate the adversarial examples by optimizing a predefined adversarial loss, which is computationally intensive. Besides, a clear trade-off has been shown between the accuracy of a classifier and its robustness against adversarial examples [14].

To address the above issues, we propose *Perturbation-Guided Adversarial Alignment (PGADA)* to relieve the negative effect caused by the small perturbations in the support-query shift. PGADA aims to generate the perturbed data as the *hard* examples, i.e., less similar to the original data point in the embedding space but still classified into the same class. Next, the model is trained on these generated data by minimizing the empirical risk to enhance the embedding model's robustness of noise tolerance. We further introduce *smooth optimal transport*, which regularizes the negative entropy of the transportation plan to take more query data points as the anchor nodes, leading to a higher error tolerance of the transportation plan.

The contributions of this work are summarized as follows.

- We formally investigate how the perturbation in the images would affect the results of optimal transportation under *the Support-Query Shift*.
- We propose *Perturbation-Guided Adversarial Alignment (PGADA)* to relieve the misalignment problem from small perturbations via deriving a more robust feature extractor and a smooth transportation plan under distribution shifts.

- Extensive experiments manifest that PGADA outperforms eleven state-of-the-art methods by 10.91% on three public datasets.

2 Preliminary

2.1 Few-shot Learning

Given a labeled support set $\mathcal{S} = \cup_{c \in \mathcal{C}} \mathcal{S}^c$, with $\mathcal{C}$ classes, where each class c has $|S^c|$ labeled examples, the goal of few-shot learning is to classify the query set $\mathcal{Q} = \cup_{c \in \mathcal{C}} \mathcal{Q}^c$ into these $\mathcal{C}$ classes. Let ϕ denotes the embedding model $\phi(x) \in \mathrm{R}^d$, which encodes the data point x to the d-dimensional feature. ϕ is learned from a labeled training set $\mathcal{D} = \{x_i, y_i\}_{i \in [1, |\mathcal{D}|]}$, where x_i is the data point and y_i is the corresponding label. The embedding model can be learned by empirical risk minimization (ERM),

$$\min_{\phi, \theta} E_{\{x,y\} \sim \mathcal{D}}[L(\theta(\phi(x)), y)],$$

where θ is a trainable parameter to map the embedding $\phi(x_i)$ to the class y_i.

Through the embedding model ϕ, we can encode the data points in support set (i.e., $x_{s,i} \in \mathcal{S}$) and query set (i.e., $x_{q,j} \in \mathcal{Q}$) to the feature $\phi(x_{s,i})$ and $\phi(x_{q,j})$, respectively. These features are used as input to a comparison function M, which measures the distance , e.g., l_2-norm, between two samples. Specifically, we classify the query example $\phi(x_{q,j})$ by averaging the embedding $\phi(x^c_{s,i})$ of the support set in class $\mathcal{S}^c$, which can be written as follows.

$$\phi^c(x_s) = \frac{1}{|S^c|} \sum_{x_{s,i} \in S^c} \phi(x_{s,i}), \quad y_q = \arg\min_c M(\phi^c(x_s), \phi(x_{q,j})).$$

2.2 The Support-Query Shift and Optimal Transportation

The conventional few-shot learning methods assume the support set and the query set lie in the same distribution. A more realistic setting is that the support set $\mathcal{S}$ and the query set $\mathcal{Q}$ follow different distributions, i.e., the support-query shift [2]. While these two sets are sampled from different distributions μ_s and μ_q, the embeddings for the support set $\mathcal{S}$ (i.e., $\phi(x_s)$) and the query set $\mathcal{Q}$ (i.e., $\phi(x_q)$) are likely to lie in different embedding spaces. Thus, it would lead to a wrong classification result via the comparison module $M(\phi(x_s), \phi(x_q))$ [2].

To tackle with the support-query shift, optimal transportation [9] is one of the effective techniques to align different distributions by a transportation plan $\pi(\mu_s, \mu_q)$, which can formally be written as follows.

$$W(\mu_s, \mu_q) = \inf_{\pi \in \Pi(\mu_s, \mu_q)} \int w(x_s, x_q) d\pi(x_s, x_q), \tag{1}$$

where $\Pi(\mu_s, \mu_q)$ is the set of transportation plans (or couplings) and w is the cost function, and W is the overall cost of transporting distribution μ_s to μ_q. In our practice, we select l_2-norm of the embedding vector, i.e., $\|\phi(x_s) - \phi(x_q)\|_2^2$, as our distance function w.

Since there are only finite samples for both the support set $x_{s,i} \in \mathcal{S}$ and the query set $x_{q,j} \in \mathcal{Q}$, the discrete optimal transportation adopts the empirical distributions to estimate the probability mass function $\hat{\mu}_s = \sum \delta_{s,i}$, and $\hat{\mu}_q = \sum \delta_{q,j}$, where $\delta_{s,i}$ and $\delta_{q,j}$ is the Dirac distribution. We obtain

$$\pi^* = \arg\min_{\pi} \sum_{\substack{x_{s,i} \sim \hat{\mu}_s \\ x_{q,j} \sim \hat{\mu}_q}} w(x_{s,i}, x_{q,j}) \pi(x_{s,i}, x_{q,j}) \tag{2}$$

Then, Sinkhorn's algorithm [10] is adopted to solve the optimal transportation plan π^*.

Equipped with the optimal plan π^*, we transport the embeddings of the support set $\phi(x_{s,i})$ to $\hat{z}_{s,i}$ by barycenter mapping [9] to adapt the support set to the query set.

$$\hat{\phi}(x_{s,i}) = \frac{\sum_{x_{q,j} \in \mathcal{Q}} \pi^*(x_{s,i}, x_{q,j}) \phi(x_{q,j})}{\sum_{x_{q,j} \in \mathcal{Q}} \pi^*(x_{s,i}, x_{q,j})}. \tag{3}$$

$\hat{\phi}(x_{s,i})$ denotes the transported embedding of $x_{s,i}$. Therefore, we can correctly measure the distance metric $M(\hat{\phi}(x_{s,i}), \phi(x_{q,j}))$ in a shared embedding space.

3 Methodology

Here, we first investigate the misestimation of optimal transport of perturbed images. Then, we illustrate our framework, namely *Perturbation-Guided Adversarial Alignment (PGADA)*, which relieves the perturbation in the images and derives a more robust embedding model. In addition, a regularized optimal transportation is introduced to align the support set and the query set better, which takes more data points from the query set as anchors to enhance the error tolerance.

3.1 Motivation

We observe that optimal transportation has a challenge, which comes from the quality of the embedding $\phi(x)$, i.e., the perturbation in the image may misguide the transportation plan. For example, clean images' embeddings may give a better transportation plan than those of foggy images. With some derivation, we formally present such misestimation as follows.

Lemma 1. *(Proof in Appendix A.1.) The error of the transportation cost is*

$$W_\sigma(\mu_s, \mu_q) \leq W(\mu_s, \mu_q) \leq W_\sigma(\mu_s, \mu_q) + \sqrt{d(\sigma_s^2 + \sigma_q^2)},$$

where $W_\sigma(\mu_s, \mu_q) := W(\mu_s * \mathcal{N}_{\sigma_s}, \mu_q * \mathcal{N}_{\sigma_q})$ denotes the original support and query set distributions μ_s and μ_q being perturbed with Gaussian noises σ_s and σ_q.

Note that $|\cdot|$ is the absolute value, and $*$ is the convolution operator. Based on Lemma 1, we estimate the error of transported embedding $\hat{\phi}(x_{s,i})$ in Eq. (3),

Theorem 1. *(Proof in Appendix A.2.) The error of the transported embedding is*

$$E[\|\hat{\phi}(x_{s,i}) - \hat{\phi}_\sigma(x_{s,i})\|_2^2] = \sqrt{d(\sigma_s^2 + \sigma_q^2)},$$

where $\hat{\phi}_\sigma(x_{s,i})$ is the transported embedding from the perturbed distribution $W_\sigma(\mu_s, \mu_q)$.

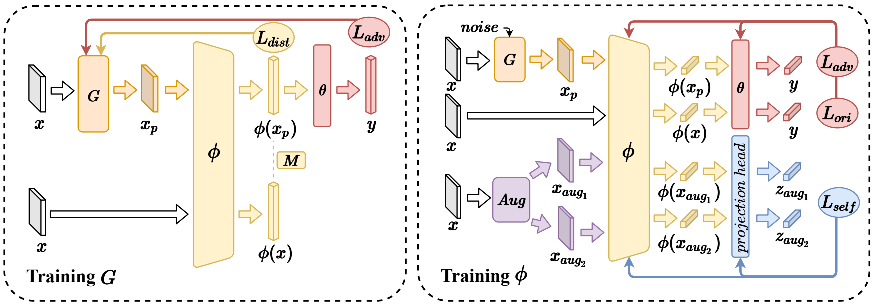

Fig. 2: Illustration of *Perturbation-Guided Adversarial Alignment (PGADA)*.

As the noise level, i.e., σ_s, and σ_q, increases, it is more likely to mislead the transportation plan and alleviate the model's performance. Therefore, it's non-trivial to learn a better embedding model ϕ having a better capability of *noise tolerance* such that $\phi(x_p) \approx \phi(x)$, where x_p is original image x with small perturbation.

3.2 Perturbation-Guided Adversarial Alignment (PGADA)

According to the Theorem 1, the optimal transport can be easily misguided by considering perturbed images. The goal of PGADA is to generate a set of augmented data to derive a more robust embedding model and relieve the perturbation in images. Recently, MaxUp [14] synthesized augmented data by *minimizing the maximum loss* over the augmented data x_p, which can be formally written as follows.

$$\min_{\phi,\theta} E_{\{x,y\}\sim\mathcal{D}}[\max_{x_p} L(\theta(\phi(x_p)), y)], \tag{4}$$

and can be easily minimized with stochastic gradient descent (SGD). Specifically, MaxUp samples a batch of augmented data x_p and compute the gradient of the data point which has the highest loss L. Therefore, the model would learn the hardest example over all augmented data x_p.

However, it's hard to collect sufficient labels for each class in few-shot learning. Therefore, instead of maximizing the empirical risk of the labeled data by Eq. (4), we introduce a self-supervised learning-based objective.

$$\min_{\phi} E_{\{x\}\sim\mathcal{D}}[\max_{x_p} M(\phi(x_p), \phi(x))], \tag{5}$$

where M is the comparison module in few-shot learning, e.g., l_2-norm. By maximizing the distance between $\phi(x_p)$ and $\phi(x)$, PGADA generates perturbed image x_p, which is less similar to the original image x in embedding space, as a *hard* example. To effectively generate the perturbed data, we introduce a semantic-aware perturbation generator to synthesize the augmented data.

$$x_p = G(x)\text{ , } s.t.\ \|x_p - x\|_2^2 \leq \epsilon, \tag{6}$$

where G is a model to generate the perturbed image x_p.[1] Besides, we utilize dropout [15] to provide the randomness of our model. Compared to conventional adversarial training

[1] We employ a 3-layer convolutional neural network as our G.

techniques [25,33], which usually sample the perturbed images from an i.i.d distribution $x_p \sim P(\cdot|x)$, e.g., Gaussian distribution, our method can encode the semantic of the image x without requiring many samples to achieve convergence [14].

At the training phase (illustrated in the left of Fig. 2), we also minimize the empirical risk of the perturbed data x_p to ensure that the generator persists enough information to predict the original label y with KL divergence [23]. The objective is,

$$\max_{G} M(\phi(G(x)), \phi(x)) - KL(\theta(\phi(G(x))), y). \tag{7}$$

Then, we adopt stochastic gradient descent (SGD) [3] to train our generator G. It is worth noting that we fix the parameters of the embedding model ϕ and θ when training the generator G to stabilize the training process [1].

In addition, we minimize the KL divergence of the original data x and the perturbed data point x_p to train the embedding model (illustrated in the right of Fig. 2).

$$L_{ori} = KL(\theta(\phi(x)), y)), \quad L_{adv} = KL(\theta(\phi(x_p)), y)).$$

To enhance the generalizability of the embeddings, we also leverage the unlabeled data by the auxiliary contrastive self-supervised learning [7]. At each iteration, we sample N images either from training or testing set,[2] and generate 2 augmentations for each images, i.e., each augmented image has 1 positive example and $2N-1$ corresponding negative examples with the self-supervised loss defined as follows.

$$L_{self} = \frac{1}{2N}\sum_{k=1}^{N}[\ell(2k-1, 2k) + \ell(2k, 2k-1)]. \tag{8}$$

ℓ denotes the *NT-Xent* Loss [7], which can be written as,

$$\ell(i,j) = -\log \frac{\exp\left(cos(z_i, z_j)/\tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{k\neq i} \exp\left(cos(z_i, z_k)/\tau\right)},$$

where z_* is defined as $W\phi(x_*)$, and W is a trainable projection matrix. Summing up, the overall all objective becomes

$$\min_{\phi,\theta} L_{ori} + \lambda_1 L_{adv} + \lambda_2 L_{self}, \tag{9}$$

where λ_1 and λ_2 are the trade-off parameters between each loss. Similarly, the classifier θ is trained by minimizing L_{ori} and L_{adv}. The pseudo code is presented in Algorithm 1.

3.3 Regularized Optimal Transportation

After deriving a robust embedding model, we extend the original transportation (Eq. 2) with negative entropy regularization to align the support set and the query set. Thus, the transport plan is penalized as follows.

$$\pi^* = \arg\min_{\pi} \sum_{\substack{x_{s,i}\sim\hat{\mu}_s \\ x_{q,j}\sim\hat{\mu}_q}} \beta w(x_{s,i}, x_{q,j})\pi(x_{s,i}, x_{q,j}) + (1-\beta)\pi(x_{s,i}, x_{q,j}) \log \pi(x_{s,i}, x_{q,j}), \tag{10}$$

[2] Note that it is valid to access the images from testing set in few-shot learning, which is named transductive few-shot learning [23].

Algorithm 1 PGADA

```
Require: Dataset $\mathcal{D}$, comparison module $M$, learning rate $\eta$, trade-off parameters $\lambda_1$ and $\lambda_2$.
Ensure: Embedding model $\phi$
Initialize generator $G$, embedding model $\phi$, and classifier $\theta$.
for $\{x, y\}$ in $\mathcal{D}$ do
    # fixed $\phi, \theta$, update $G$
    $x_p = G(x),\ L_{dist} = -M(\phi(x_p), \phi(x)),\ L_{adv} = KL(\theta(\phi(x_p)), y))$
    $G \leftarrow G - \eta\nabla(L_{dist} + L_{adv})$ # Generated less similar data points with perturbations.
    # fixed G, update $\phi, \theta$
    $x_p = G(x),\ L_{ori} = KL(\theta(\phi(x)), y)),\ L_{adv} = KL(\theta(\phi(x_p)), y))$
    $\phi \leftarrow \phi - \eta\nabla\ (L_{ori} + \lambda_1 L_{adv} + \lambda_2 L_{self})$
    $\theta \leftarrow \theta - \eta\nabla\ (L_{ori} + \lambda_1 L_{adv})$ # classifying the generated samples correctly.
```

where β is the weight parameter to determine the smoothness of the transportation plan. In other words, the data points in the support set take more data points in the query set as anchors for alignment. Accordingly, we can better align the support set to the query set because each labeled data point is representative, especially with limited data points and labels.

4 Experiment

We compare PGADA to eleven baselines, including conventional few-shot learning methods and adversarial data augmentation methods on three real datasets. Due to the space constraints, additional experimental results are presented in Appendix B.

4.1 Experiment Setup

Datasets. Following [2], we validate our framework on three benchmark datasets in few-shot learning. 1) *CIFAR100* [21] consists of $60,000$ images, evenly distributed in 100 classes (64 classes for training, 10 classes for validation, and 25 classes for testing). 2) *miniImageNet* [31] is a subset of ImageNet, with $60,000$ images from 100 classes (64 classes for training, 16 classes for validation, and 20 classes for testing). 3) *FEMNIST* [6] is a dataset with $805,263$ handwritten characters in 62 classes (42 classes for training, 10 classes for validation, and 10 classes for testing).

Evaluation. Following [2], the average top-1 accuracy scores with 95% confidence interval from 2000 runs are reported. All experiments are under the 5-way setting. Note that we conduct the tasks of 1-shot and 5-shot with 8-target and 16-target, i.e., 1 or 5 instances per class in the support set and 8 or 16 instances in query set, in CIFAR100, and miniImageNet. While in FEMNIST, we only adopt tasks of 1-shot and 1-target, restricted by the setting of the dataset.

Implementation. We use a 4-layer convolutional network as the embedding model ϕ on CIFAR100 and FEMNIST, and ResNet18 for miniImageNet. As a general framework, we combine PGADA with two classifiers, i.e., ProtoNet and MatchingNet. The hyperparameters are selected by grid search with $\eta = 1e - 3$, $b = 128$, $d = 128$, $\lambda_1 = 1$, $\lambda_2 = 1$, $\beta = 0.5$, respectively. Note that we also employ transductive batch normalization [22]. Besides, SGD optimizer [20] is adopted to train all models in 200

Dataset	CIFAR100				miniImageNet				FEMNIST
	8-target		16-target		8-target		16-target		1-target
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot
Few-shot Learning									
MatchingNet [32]	$30.71_{\pm 0.38}$	$41.15_{\pm 0.45}$	$31.00_{\pm 0.34}$	$41.83_{\pm 0.39}$	$35.26_{\pm 0.50}$	$44.75_{\pm 0.55}$	$37.20_{\pm 0.48}$	$44.22_{\pm 0.52}$	$84.25_{\pm 0.71}$
ProtoNet [28]	$30.02_{\pm 0.40}$	$42.77_{\pm 0.47}$	$30.29_{\pm 0.33}$	$42.52_{\pm 0.41}$	$36.37_{\pm 0.50}$	$47.58_{\pm 0.57}$	$35.69_{\pm 0.45}$	$46.29_{\pm 0.53}$	$84.31_{\pm 0.73}$
TransPropNet [22]	$34.15_{\pm 0.39}$	$47.39_{\pm 0.42}$	$34.20_{\pm 0.40}$	$44.31_{\pm 0.38}$	$24.10_{\pm 0.27}$	$27.24_{\pm 0.33}$	$25.38_{\pm 0.30}$	$28.05_{\pm 0.30}$	$86.42_{\pm 0.76}$
FTNET [11]	$28.91_{\pm 0.37}$	$37.28_{\pm 0.40}$	$28.66_{\pm 0.31}$	$37.37_{\pm 0.33}$	$39.02_{\pm 0.46}$	$51.27_{\pm 0.45}$	$39.70_{\pm 0.40}$	$52.00_{\pm 0.37}$	$86.13_{\pm 0.71}$
TP [2]	$34.00_{\pm 0.46}$	$49.71_{\pm 0.47}$	$35.55_{\pm 0.41}$	$50.24_{\pm 0.39}$	$40.49_{\pm 0.54}$	$59.85_{\pm 0.49}$	$43.83_{\pm 0.51}$	$55.87_{\pm 0.42}$	$93.63_{\pm 0.63}$
Adversarial Data Augmentation									
MixUp [36]	$37.82_{\pm 0.47}$	$52.57_{\pm 0.47}$	$38.52_{\pm 0.42}$	$53.33_{\pm 0.40}$	$42.98_{\pm 0.54}$	$57.22_{\pm 0.48}$	$43.64_{\pm 0.48}$	$57.33_{\pm 0.42}$	$97.22_{\pm 0.46}$
CutMix [35]	$39.36_{\pm 0.48}$	$54.76_{\pm 0.48}$	$40.05_{\pm 0.44}$	$55.44_{\pm 0.40}$	$35.50_{\pm 0.52}$	$45.50_{\pm 0.56}$	$35.78_{\pm 0.48}$	$44.85_{\pm 0.52}$	$96.89_{\pm 0.49}$
Autoencoder [26]	$39.05_{\pm 0.50}$	$53.24_{\pm 0.47}$	$39.82_{\pm 0.44}$	$53.88_{\pm 0.40}$	$45.36_{\pm 0.56}$	$57.69_{\pm 0.51}$	$45.65_{\pm 0.52}$	$57.39_{\pm 0.44}$	$96.53_{\pm 0.43}$
AugGAN [17]	$39.54_{\pm 0.50}$	$53.05_{\pm 0.47}$	$39.50_{\pm 0.45}$	$53.42_{\pm 0.39}$	$44.65_{\pm 0.55}$	$57.55_{\pm 0.50}$	$44.91_{\pm 0.49}$	$57.10_{\pm 0.42}$	$96.42_{\pm 0.52}$
MaxEntropy [38]	$38.14_{\pm 0.40}$	$51.02_{\pm 0.56}$	$38.21_{\pm 0.34}$	$51.33_{\pm 0.52}$	$48.21_{\pm 0.36}$	$57.67_{\pm 0.63}$	$48.99_{\pm 0.21}$	$59.01_{\pm 0.44}$	$97.19_{\pm 0.51}$
MaxUp [14]	$34.84_{\pm 0.44}$	$47.51_{\pm 0.46}$	$35.20_{\pm 0.40}$	$47.63_{\pm 0.39}$	$37.62_{\pm 0.55}$	$48.65_{\pm 0.58}$	$38.13_{\pm 0.50}$	$49.19_{\pm 0.51}$	$96.48_{\pm 0.53}$
Ours									
PGADA (ProtoNet)	$42.16_{\pm 0.52}$	$\mathbf{56.52}_{\pm \mathbf{0.47}}$	$\mathbf{42.73}_{\pm \mathbf{0.46}}$	$\mathbf{56.83}_{\pm \mathbf{0.40}}$	$55.44_{\pm 0.61}$	$\mathbf{67.34}_{\pm \mathbf{0.49}}$	$55.69_{\pm 0.62}$	$\mathbf{66.90}_{\pm \mathbf{0.50}}$	$\mathbf{97.98}_{\pm \mathbf{0.40}}$
PGADA (MatchingNet)	$\mathbf{42.25}_{\pm \mathbf{0.53}}$	$50.98_{\pm 0.45}$	$42.60_{\pm 0.45}$	$51.80_{\pm 0.39}$	$\mathbf{56.15}_{\pm \mathbf{0.61}}$	$63.08_{\pm 0.49}$	$\mathbf{56.12}_{\pm \mathbf{0.57}}$	$63.61_{\pm 0.45}$	$97.96_{\pm 0.39}$

Table 1: Accuracy comparison of the three datasets with two types of baselines.

epochs with early stopping. All experiments are implemented in a server with a Intel(R) Core(TM) i9-9820X CPU@3.30GHz, and a GeForce RTX 3090 GPU.

4.2 Quantitative Analysis

Few-shot learning. We first compare five state-of-the-art few-shot learning methods, including *i) MatchingNet [32]*, *ii) ProtoNet [28]*, *iii) TransPropNet [22]*, *iv) FTNET [11]*, and *v) Transported Prototypes (TP) [2]*. As shown in Table 1, PGADA outperforms the best baseline (TP) by at least 13.12% in CIFAR100, 12.51% in miniImageNet, and 4.65% in FEMNIST, respectively. Note that PGADA achieves the best improvement on miniImageNet, since the image size of miniImageNet is the largest one compared to CIFAR100 and FEMNIST. In addition, PGADA achieves better improvement in the tasks of 1-shot (i.e., 22.46%) than the tasks of 5-shot (14.77%) averagely, which demonstrates the robustness of our method, especially with limited labeled data. Compared with the first four baselines, i.e., ProtoNet, MatchingNet, TransPropNet, and FTNET, our method achieves consistent improvement, 41.48%, 39.78%, 71.61%, and 54.85%, respectively, since these methods do not consider the inherent distribution shift between the support and query set. While TP also utilizes optimal transport to align the support and query set, it still shows relatively weak performance compared to PGADA because the transportation plan of TP is misguided by the small perturbations in the images, as proved in Theorem 1. Since PGADA is a model agnostic adversarial alignment framework, we equip PGADA with different classifiers, e.g., ProtoNet and MatchingNet. Our framework outperforms the baseline by 39.78% and 21.81% in ProtoNet and MatchingNet, respectively, manifesting the generability of PGADA. We observe that ProtoNet outperforms MatchingNet in the 5-shot case as the prototypes reduce the bias by averaging the embedding vectors, which is more robust.

Adversarial data augmentation. In addition, as our method is closed to adversarial data augmentation, we also compares six adversarial data augmentation methods, including *vi) MixUp* [36], *vii) CutMix* [35], *viii) Autoencoder* [26], *ix) AugGAN* [17], *x) MaxEntropy* [38], and *xi) MaxUp* [14]. According to Table 1, PGADA outperforms MixUp and CutMix by 9.12% and 8.84% on average. Compared with the Autoencoder and AugGAN, our method improved by 11.77% and 12.41% because they synthesize similar patterns to the original images to alleviate the data sparsity problem, which

Dataset	CIFAR100				miniImageNet				FEMNIST
	8-target		16-target		8-target		16-target		1-target
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot
PGADA	$\mathbf{42.16}_{\pm 0.52}$	$\mathbf{56.52}_{\pm 0.47}$	$\mathbf{42.73}_{\pm 0.46}$	$\mathbf{56.83}_{\pm 0.40}$	$\mathbf{55.44}_{\pm 0.61}$	$\mathbf{67.34}_{\pm 0.49}$	$\mathbf{55.69}_{\pm 0.62}$	$\mathbf{66.90}_{\pm 0.50}$	$\mathbf{97.98}_{\pm 0.40}$
Generator									
fixed G	$38.58_{\pm 0.48}$	$52.41_{\pm 0.47}$	$39.26_{\pm 0.43}$	$52.67_{\pm 0.39}$	$43.50_{\pm 0.55}$	$55.65_{\pm 0.50}$	$43.48_{\pm 0.51}$	$55.42_{\pm 0.43}$	$96.41_{\pm 0.52}$
w/o noise	$37.16_{\pm 0.47}$	$50.12_{\pm 0.46}$	$37.73_{\pm 0.41}$	$50.50_{\pm 0.38}$	$44.06_{\pm 0.56}$	$56.97_{\pm 0.48}$	$44.42_{\pm 0.49}$	$56.96_{\pm 0.42}$	$96.89_{\pm 0.48}$
w/o KL	$37.30_{\pm 0.47}$	$50.79_{\pm 0.46}$	$37.91_{\pm 0.42}$	$51.35_{\pm 0.39}$	$44.22_{\pm 0.54}$	$55.04_{\pm 0.49}$	$44.21_{\pm 0.49}$	$53.96_{\pm 0.41}$	$96.49_{\pm 0.48}$
Regularized Optimal Transport (OT)									
w/o OT	$35.76_{\pm 0.41}$	$54.06_{\pm 0.45}$	$35.66_{\pm 0.35}$	$54.09_{\pm 0.38}$	$44.30_{\pm 0.52}$	$61.23_{\pm 0.53}$	$44.15_{\pm 0.46}$	$60.86_{\pm 0.48}$	$94.03_{\pm 0.48}$
TP [2]	$34.00_{\pm 0.46}$	$49.71_{\pm 0.47}$	$35.55_{\pm 0.41}$	$50.24_{\pm 0.39}$	$40.49_{\pm 0.54}$	$59.85_{\pm 0.49}$	$43.83_{\pm 0.51}$	$55.87_{\pm 0.42}$	$93.63_{\pm 0.63}$
TP w/o OT	$33.07_{\pm 0.38}$	$50.99_{\pm 0.44}$	$32.96_{\pm 0.32}$	$50.71_{\pm 0.37}$	$38.07_{\pm 0.45}$	$55.31_{\pm 0.51}$	$37.94_{\pm 0.41}$	$55.11_{\pm 0.44}$	$91.84_{\pm 0.56}$
Self-supervised Learning (SSL)									
w/o SSL	$39.33_{\pm 0.50}$	$53.66_{\pm 0.47}$	$40.31_{\pm 0.44}$	$54.23_{\pm 0.40}$	$47.96_{\pm 0.57}$	$61.38_{\pm 0.49}$	$48.70_{\pm 0.52}$	$61.44_{\pm 0.43}$	$97.07_{\pm 0.48}$

Table 2: The results of ablation studies.

cannot create new information that is not included in the given data [14]. In contrast, PGADA is able to explore the perturbed data that is most likely to confuse the model, which can be regarded as hard examples. Compared to MaxEntropy, which uses opposite gradients to search the worst data point, we observe that PGADA outperforms MaxEntropy by 6.52% on average since PGADA adds noise to the generator, increasing the uncertainty of the generated data. Then, by adding these data points to the training phase, the model can defense against more unknown perturbations attacking, resulting in higher accuracy. In addition, our method also significantly outperforms MaxUp because the selected worst data point in MaxUp is still near the original images, resulting in less robustness. In addition, PGADA is more suitable for few-shot learning, as it generates the perturbations in a self-supervised manner by comparing the embedding of original images and perturbed images to explore hard examples.

4.3 Ablation Studies

We conduct ablation studies to evaluate the importance of different modules in PGADA. Note that we only present the results of PGADA with ProtoNet as PGADA with MatchingNet shows similar ones.

Effect of Generator. We compare PGADA with three different variants of the generator, including i) *fixed G*, fixing the parameters of the generator, ii) *w/o noise*, removing the noise function of the generator, and iii) *w/o KL*, removing the classification loss of perturbed data. When we fix the parameters of the generator, we can observe that the performance of PGADA drops by 8.46%, 24.31%, and 1.63% in three datasets, respectively. It demonstrates that our trainable generator is able to extract the information from original images x to generate meaningful perturbed images x_p. As we remove the noise, it also shows a consistent decline in performance, which demonstrates that leveraging some randomness in the training process is helpful to explore the perturbation. Lastly, our method without classification loss (i.e., *w/o KL*) still outperforms TP [2]. It demonstrates that the self-supervised-learning-based objective function (Eq. 5) works well to learn the inherent information from the original data, leading to a robust embedding model. However, the classification loss of the perturbed data also boosts the model capability as it regularizes the generator by preserving useful information to predict a correct label, rather than exploration via random walk, which is less efficient.

Effect of Regularized Optimal Transportation. Here, we investigate the effect of regularized optimal transportation (OT), which plays a crucial role during the evaluation phase in few-shot learning under the support-query shift. It shows that OT significantly improves the model's capability since it aligns the distributions of the support and query set. Even though the performance of PGADA drops as we remove OT. PGADA still outperforms the state-of-the-art method TP [2], which also utilizes OT by 4.86%. Also, compared TP to TP w/o OT, we observe that optimal transportation in TP does not perform well as in PGADA. This observation echoes the motivation of this work (Theorem 1), i.e., the small perturbations misguide the optimal transportation plan. Also, this result manifests that a robust embedding model indeed alleviates the support-query shift in few-shot learning.

Effect of Self-Supervised Learning. Last, we evaluate the effect of self-supervised learning. Equipped with the contrastive loss, PGADA improves by 5.83%, 12.13%, and 0.94% in three datasets, respectively, since the model leverages the structural information of the unlabeled data in the training phase and thus leaps in model performance. The results also demonstrate that PGADA and self-supervised learning can incorporate together to explore the information from the training set and testing set to improve the generality of the embedding model. Understanding why this combination performs well is interesting for future works.

5 Related Work

Few-shot Learning. Few-shot learning can be divided into three main categories, including optimization-based method [12], hallucination-based method [16], and metric-based method [32]. On the one hand, the optimization-based methods aim to train a quickly adaptive model and quickly adapt to other tasks by fine-tuning [12,4]. On the other hand, the hallucination-based methods hallucinate the scarce labeled data by synthesizing representations [16] with Generative Adversarial Networks (GANs) [1]. Our work is most relevant to the metric-based methods where Vinyals et al. [32] and Snell et al. [28] introduce the pairwise (and classwise) metrics to determine the label of the query set according to the support set. Sung et al. [29] model the non-linear relation between class representations and queries by neural networks. Moreover, cross-domain few-shot learning [8] argues that the training and testing sets may not lie in the same distribution. Thus, several techniques such as optimal transport [9], self-supervised learning [37,23] have been introduced to relieve the domain shift. While the domain shift not only occurs between the training set and the testing set but also between the support set and the query set [2]. In addition, the optimal transport plan can be easily misguided by the perturbation in the images and leads to unacceptable performance.

Data Augmentation and Adversarial Training. Data augmentation has been widely used in machine learning with various transformations on a single image [27], e.g., resizing, flipping, rotation, cropping, or multiple images, e.g., MixUp [36] and CutMix [35]. Another line of studies [7,34] works on several self-training schemes by maximizing the pairwise similarity of augmented data. However, the aforementioned methods cannot create new information not included in the given data [30] since they synthesize and train on similar images. In contrast, adversarial training [25,38] has been developed to defend against adversarial attacks, which conducts the attack to generate perturbed examples from clean data that the model misclassifies. Then, the generated examples are used as training data to compensate for the model weaknesses, contributing to model robustness. A similar work, MaxUP [14], generates a set of random augmented data and searches the hardest example that would maximize the classification loss. However, MaxUp requires numerous random augmented data to explore more information, thus leading to ineffectiveness. In contrast, PGADA directly generates hard samples in a self-supervised manner, which is simpler and computationally efficient.

6 Conclusion

In this paper, we propose *Perturbation-Guided Adversarial Alignment (PGADA)* to solve the support-query shift in few-shot learning. Our key idea is to generate perturbed images that are hard to classify and then train on these perturbed data to derive a more robust embedding model and alleviate the misestimation of optimal transportation. In addition, a negative entropy regularization is introduced to obtain a smooth transportation plan. The experiment results manifest that PGADA outperforms eleven baselines by at least 13.12% in CIFAR100, 12.51% in miniImageNet, and 4.65% in FEMNIST, respectively. Future works include applying PGADA to other computer vision tasks and incorporating it with other data augmentation schemes.

Acknowledgement

S. Jiang is supported by the science and technology plan project in Huizhou (No. 2020SD0402030)

References

1. Antoniou, A., Storkey, A., Edwards, H.: Data augmentation generative adversarial networks. In: ICLR (2017)
2. Bennequin, E., Bouvier, V., Tami, M., Toubhans, A., Hudelot, C.: Bridging few-shot learning and adaptation: New challenges of support-query shift. ECML-PKDD (2021)
3. Bottou, L.: Stochastic gradient descent tricks. In: Neural networks: Tricks of the trade, pp. 421–436. Springer (2012)
4. Boudiaf, M., Masud, Z.I., Rony, J., Dolz, J., Piantanida, P., Ayed, I.B.: Transductive information maximization for few-shot learning. arXiv preprint arXiv:2008.11297 (2020)
5. Bronskill, J., Gordon, J., Requeima, J., Nowozin, S., Turner, R.: Tasknorm: Rethinking batch normalization for meta-learning. In: ICML. PMLR (2020)
6. Caldas, S., Duddu, S.M.K., Wu, P., Li, T., Konečnỳ, J., McMahan, H.B., Smith, V., Talwalkar, A.: Leaf: A benchmark for federated settings. NeurIPS (2019)
7. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML. pp. 1597–1607. PMLR (2020)
8. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C.F., Huang, J.B.: A closer look at few-shot classification. arXiv preprint arXiv:1904.04232 (2019)
9. Courty, N., Flamary, R., Tuia, D., Rakotomamonjy, A.: Optimal transport for domain adaptation. IEEE TPAMI (2016)
10. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. NeurIPS **26**, 2292–2300 (2013)
11. Dhillon, G.S., Chaudhari, P., Ravichandran, A., Soatto, S.: A baseline for few-shot image classification. In: ICLR (2019)
12. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017)
13. Garcia, V., Bruna, J.: Few-shot learning with graph neural networks. In: ICLR (2017)
14. Gong, C., Ren, T., Ye, M., Liu, Q.: Maxup: Lightweight adversarial training with data augmentation improves neural network training. In: CVPR. pp. 2474–2483 (2021)

15. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. NeurIPS **27**, 2672–2680 (2014)
16. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: CVPR. pp. 3018–3027 (2017)
17. Huang, S.W., Lin, C.T., Chen, S.P., Wu, Y.Y., Hsu, P.H., Lai, S.H.: Auggan: Cross domain adaptation with gan-based data augmentation. In: ECCV. pp. 718–731 (2018)
18. Jiang, S., Chen, H.W., Chen, M.S.: Dataflow systolic array implementations of exploring dual-triangular structure in qr decomposition using high-level synthesis. In: ICFPT (2021)
19. Jiang, S., Yao, X., Long, Q., Chen, J., Jiang, H.: Fund investment decision in support vector classification based on information entropy. Review of Economics & Finance (2019)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2014)
21. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
22. Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S.J., Yang, Y.: Learning to propagate labels: Transductive propagation network for few-shot learning. In: ICLR (2018)
23. Phoo, C.P., Hariharan, B.: Self-training for few-shot transfer across extreme task differences. In: ICLR (2020)
24. Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised few-shot classification. arXiv preprint arXiv:1803.00676 (2018)
25. Samangouei, P., Kabkab, M., Chellappa, R.: Defense-gan: Protecting classifiers against adversarial attacks using generative models. In: ICLR (2018)
26. Schonfeld, E., Ebrahimi, S., Sinha, S., Darrell, T., Akata, Z.: Generalized zero-and few-shot learning via aligned variational autoencoders. In: CVPR. pp. 8247–8255 (2019)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
28. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: NeurIPS (2017)
29. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: CVPR. pp. 1199–1208 (2018)
30. Theagarajan, R., Chen, M., Bhanu, B., Zhang, J.: Shieldnets: Defending against adversarial attacks using probabilistic adversarial robustness. In: CVPR. pp. 6988–6996 (2019)
31. Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Evci, U., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P.A., et al.: Meta-dataset: A dataset of datasets for learning to learn from few examples. arXiv preprint arXiv:1903.03096 (2019)
32. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: NeurIPS (2016)
33. Wang, Y.X., Girshick, R., Hebert, M., Hariharan, B.: Low-shot learning from imaginary data. In: CVPR. pp. 7278–7286 (2018)
34. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: CVPR. pp. 10687–10698 (2020)
35. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: ICCV. pp. 6023–6032 (2019)
36. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: ICLR (2017)
37. Zhao, A., Ding, M., Lu, Z., Xiang, T., Niu, Y., Guan, J., Wen, J.R.: Domain-adaptive few-shot learning. In: WACV. pp. 1390–1399 (2021)
38. Zhao, L., Liu, T., Peng, X., Metaxas, D.: Maximum-entropy adversarial data augmentation for improved generalization and robustness. arXiv preprint arXiv:2010.08001 (2020)

A Proof

A.1 Proof of Lemma 1

The left side inequality immediately follows because W is non-increasing under convolutions, since $\mathcal{N}_{\sqrt{\sigma_s^2+\sigma_q^2}} = \mathcal{N}_{\sigma_s} * \mathcal{N}_{\sigma_q}$, where $*$ is the convolution operator.

In the right side of the inequality, we adopt Kantorovich-Rubinstein duality to write the optimal transport as follows.

$$W(\mu_s, \mu_q) = \sup_{\|w\|_{Lip} \leq 1} E_{\mu_s}[w] - \mathrm{E}_{\mu_q}[w]$$

$$W_\sigma(\mu_s, \mu_q) = \sup_{\|w\|_{Lip} \leq 1} E_{\mu_s * \mathcal{N}_{\sigma_s}}[w_\sigma] - \mathrm{E}_{\mu_q * \mathcal{N}_{\sigma_q}}[w_\sigma]$$

where $\|\cdot\|_{Lip}$ is the Lipschitz norm. Letting w^* be optimal for $W(\mu_s, \mu_q)$, we obtain,

$$W_\sigma(\mu_s, \mu_q) = \sup_{\|\phi\|_{Lip} \leq 1} E_{\mu_s * \mathcal{N}_{\sigma_s}}[w^*] - \mathrm{E}_{\mu_q * \mathcal{N}_{\sigma_q}}[w^*]. \tag{11}$$

Let $X_s \sim \mu_s$, $Z_s \sim N_{\sigma_s}$ as independent random variables, we have,

$$\begin{aligned} &|E_{\mu_s}[w^*] - E_{\mu_s * N_{\sigma_s}}[w^*]| \\ =&E[w^*(X_s)] - E[w^*(X + Z_s)] \\ \leq&E[\|Z_s\|_2^2] = d\sigma_s. \end{aligned} \tag{12}$$

where the last in equality uses $\|w^*\|_{Lip} \leq 1$. d is the dimension of the embedding vector. Similarly, $X_q \sim \mu_q$, $Z_q \sim N_{\sigma_q}$ as independent random variables, we have,

$$\begin{aligned} &|E_{\mu_q}[w^*] - E_{\mu_q * N_{\sigma_q}}[w^*]| \\ =&E[w^*(X_q)] - E[w^*(X + Z_q)] \\ \leq&E[\|Z_q\|_2^2] = d\sigma_q. \end{aligned} \tag{13}$$

By inserting Eq. (12) and Eq. (13) to Eq. (11), and Cauchy-Schwarz inequality, we concludes the proof.

A.2 Proof of Theorem 1

Base on Lemma 1, barycentric coordinate is defined as follows,

$$\hat{\pi}_i^* = \frac{\pi^*(x_{s,i}, x_{q,j})}{\sum_{x_{q,j} \in \mathcal{Q}} \pi^*(x_{s,i}, x_{q,j})} \sim \mathcal{N}_{\sigma_s} \tag{14}$$

Let $X_q \sim \mu_q$, $X_q^\sigma \sim \mu_q * N_{\sigma_q}$ as independent random variables,

$$E[X_q^{\sigma(t)} - X_q^{(t)}] = \sigma_q, \tag{15}$$

where $X_q^{\sigma(t)}$ and $X_q^{(t)}$ denotes the t-th dimension of random variable X_q^σ and X_q, respectively.

Combining Eq. (14) and Eq. (15), the projected distribution $\hat{X}_s \sim \mu_s * N_{\sigma_s} * N_{\sigma_q} = \mu_s * N_{\sqrt{\sigma_s^2+\sigma_q^2}}$.

$$E[\hat{X}_s - X_s] = d\sqrt{\sigma_s^2 + \sigma_q^2}. \tag{16}$$

The theorem follows.

Dataset	CIFAR100				miniImageNet				FEMNIST
	8-target		16-target		8-target		16-target		1-target
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot
PGADA w/ TBN	$\mathbf{42.16}_{\pm\mathbf{0.52}}$	$\mathbf{56.52}_{\pm\mathbf{0.47}}$	$\mathbf{42.73}_{\pm\mathbf{0.46}}$	$\mathbf{56.83}_{\pm\mathbf{0.40}}$	$\mathbf{55.44}_{\pm\mathbf{0.61}}$	$\mathbf{67.34}_{\pm\mathbf{0.49}}$	$\mathbf{55.69}_{\pm\mathbf{0.62}}$	$\mathbf{66.90}_{\pm\mathbf{0.50}}$	$\mathbf{97.71}_{\pm\mathbf{0.42}}$
PGADA w/ CBN	$39.23_{\pm0.50}$	$52.78_{\pm0.47}$	$39.75_{\pm0.45}$	$52.96_{\pm0.41}$	$45.31_{\pm0.56}$	$57.42_{\pm0.48}$	$45.63_{\pm0.50}$	$57.34_{\pm0.43}$	$96.12_{\pm0.54}$

Table 3: Comparsion of TBN and CBN under the support-query shift.

Dataset		CIFAR100				miniImageNet			
		8-target		16-target		8-target		16-target	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
w/ SQS	PGADA	$42.16_{\pm0.52}$	$56.52_{\pm0.47}$	$42.73_{\pm0.46}$	$56.83_{\pm0.40}$	$55.44_{\pm0.61}$	$67.34_{\pm0.49}$	$55.69_{\pm0.62}$	$66.90_{\pm0.50}$
	TP	$34.00_{\pm0.46}$	$49.71_{\pm0.47}$	$35.55_{\pm0.41}$	$50.24_{\pm0.39}$	$40.49_{\pm0.54}$	$59.85_{\pm0.49}$	$43.83_{\pm0.51}$	$55.87_{\pm0.42}$
w/o SQS	PGADA	$90.32_{\pm0.18}$	$94.26_{\pm0.17}$	$84.61_{\pm0.20}$	$87.24_{\pm0.19}$	$86.83_{\pm0.27}$	$83.58_{\pm0.27}$	$81.46_{\pm0.27}$	$79.75_{\pm0.28}$
	TP	$85.67_{\pm0.26}$	$88.52_{\pm0.17}$	$82.40_{\pm0.20}$	$84.15_{\pm0.17}$	$64.27_{\pm0.39}$	$75.22_{\pm0.30}$	$69.68_{\pm0.59}$	$72.67_{\pm0.52}$

Table 4: The case study of the support-query shift.

B Additional Experiment Results

B.1 Effect of the Transductive Batch Normalization

Here, we explore the effect of transductive batch normalization (TBN) [5] and conventional batch normalization (CBN). TBN is first proposed in MAML [12]. Ren *et al.*[24] point out that TBN leverages a query set as a whole to bring a significant boost in performances compared to CBN. Therefore, we both deploy TBN and CBN to evaluate our method. From Table 3, we observe that PGADA gets unignorable benefit from TBN, which is consistent with prior works. Specifically, PGADA w/ TBN outperforms PGADA w/ CBN, 7.21%, 19.57%, 1.06% on CIFAR100, miniImageNet, FEMNIST respectively. Once the self-supervised loss is removed, we observe that PGADA w/ TBN has a similar performance with PGADA w/ CBN. It demonstrates that TBN and self-supervised learning can achieve better performance by exploring the inherent structure information from the unlabeled data. Summing up, the prior works [12,31] point out that TBN is efficient in Meta-Learning, where TBN can explore whole query distribution cross tasks to achieve improvement. Our work obverse that the combination of TBN and self-supervised learning also performs well as they can learn more information from unlabeled data.

B.2 Case Study

Table 4 presents a case study to understand the effect of the support-query shift on CIFAR100 and miniImageNet. As mentioned in Sec. 4.1, we strengthen the support-query shift by adopting different transformation functions on the support and query sets. We find that the support-query shift indeed significantly affects the behavior of PGADA, i.e., the performance drops by at least 40% in both PGADA and TP. The decrease in performance shows the importance of this work in the real world. On the other hand, the performance of PGADA without the support-query shift also outperforms that of TP since our framework learns a more robust feature extractor. Summing up, PGADA can still provide promising performance despite the absence of the support-query shift, showing PGADA is a general framework for few-shot learning.