Semantic Transformation-based Data Augmentation for Few-Shot Learning

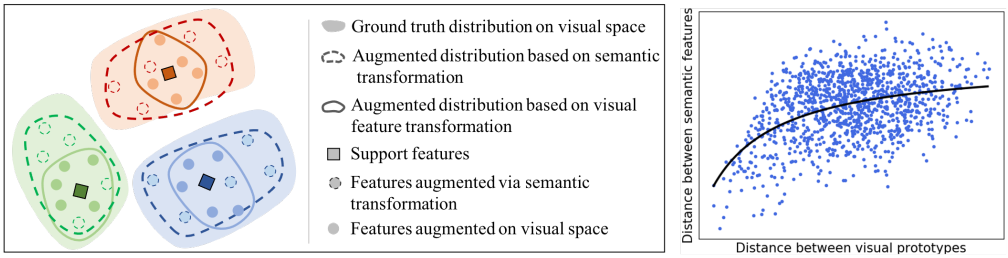
Anonymous authors
Paper under double-blind review

Abstract

Few-shot learning (FSL) as a data-scarce method, aims to recognize instances of unseen classes solely based on very few examples. However, the model can easily become overfitted due to the biased distribution formed with extremely limited training data. This paper presents a task specific data augmentation approach by transferring samples from base dataset to the novel tasks in an encoder-decoder paradigm, which guarantees generating semantically meaningful features. Specifically, the feature transfer process is carried out in semantic space. We further impose a compactness constraint to the generated features with the prototypes working as the reference points, which ensures the generated features distribute around the class centers. Moreover, we incorporate the cluster centers of the query set with the prototypes of the support set to reduce the bias of the class centers. With the supervision of the compactness loss, the model is encouraged to generate discriminative features with high inter-class dispersion and intra-class compactness. Extensive experiments show that our method outperforms the state-of-the-arts on four benchmarks, namely MiniImageNet, TieredImageNet, CUB and CIFAR-FS.

1 Introduction

Deep learning have shown impressive performance in various computer vision tasks based on massive supervisions. However, it is impractical to obtain large number of well-annotated training data in some real applications (Vartak et al., 2017; Altae-Tran et al., 2017), which motivates us to explore the data-scarce technique, called few-shot learning (FSL) to learn new concepts from very few labeled examples. FSL leverages experiences from similar tasks based on the episodic paradigm, which equips the artificial intelligence with human-like ability that continuously learns novel concept from even a single example (He et al., 2015). Data augmentation is a straightforward technique to alleviate the extremely low-data problem. It is natural that with the growing of labeled samples, the data distribution can gradually approach the ground truth distribution. However, most methods augment data on visual feature space with few labeled support samples as supervision, leading to marginal performance boosting. In (Yang et al., 2020), a DC method is proposed to calibrate the distributions of the few samples on novel classes by transferring the distribution of training dataset (base set) with large amounts of samples. The distribution transformation is implemented in the visual feature


Ground truth distribution on visual space
Augmented distribution based on semantic transformation
Augmented distribution based on visual feature transformation
Support features
Features augmented via semantic transformation
Features augmented on visual space
Distance between semantic features
Distance between visual prototypes

(*a*) Distribution comparison about the augmented features based on visual and semantic transformation.

(b) Scatter plot of the distances between visual and semantic spaces.

Figure 1: Conceptual description of STDA.

space, and the few shot labeled samples are worked as anchor. The calibrated distribution is still limited by the few labeled samples. As shown in Figure 1 (a), samples augmented in visual space distribute around the support samples, and can not cover the ground truth distribution well. The observation that samples from different classes which are semantically similar usually have similar visual representations (see Figure 1 (b)) inspires us to borrow samples from base dataset to enlarge the support set via semantic transformation. Since the semantic representation of each class is unique and fixed, it won't be affected by the number of labeled samples. Thus the augmented samples transferred from base dataset are less biased. The "semantic feature" mentioned across the whole paper denotes the word embedding of the class label (textual descriptions of categories), which is computed with GloVe (Pennington et al., 2014) method.

In this work, we present a semantic transformation-based data augmentation approach (STDA) to generate highly effective samples in an encoder-decoder paradigm (Kodirov et al., 2017). A few previous works have exploited the idea of applying encoder and decoder to augment samples. DeVries et.al (DeVries & Taylor, 2017) applies simple transformations such as adding noise, interpolating, or extrapolating to the features in a latent embedding space, and reconstruct the augmented features to the original embedding space. Schwartz et.al (Schwartz et al., 2018) propose a Delta-encoder method that extracts transferable intra-class deltas (i.e. deformations) between same-class pairs of training examples, and apply the deltas to the few samples of the novel classes. Both (DeVries & Taylor, 2017) and (Schwartz et al., 2018) employ the encoder-decoder paradigm to augment samples. They project the features to a low dimensional latent embedding space, and apply some perturbation to the latent features to generate diverse features. However, the feature diversity can be limited since the "anchor" example is the visual feature of the few novel samples. Our method augment new samples by converting a large amount of training samples of similar classes to novel classes and transform the features in semantic space rather than in an arbitrary latent space, which can preserve the diversity of the training data and generate semantically meaningful features for the novel classes.

To reduce the bias of the anchor, we propose a novel prototype rectification method based on the shortest path optimization strategy. There have been some works on prototype rectification. MetaNODE (Zhang et al., 2022) models the prototype rectification as a prototype optimization problem, that uses ODEs for meta-optimization on the mean prototypes. It involves a complex meta learning process. Liu et.al (Liu et al., 2020) propose a BD-CSPN method to rectify the prototypes by incorporating pseudo labeled query samples with the support samples. However, the pseudo labels are predicted based on the similarity to the basic prototypes. The prediction process is inevitable to introduce bias caused by the limited support samples, which will produce incorrect pseudo labels and further impose negative impact on the subsequent prototypes rectification. Different from these methods, we rectify the prototypes by incorporating the cluster centers of the query samples. The clustering process relies on the data distribution of the query data, which is agnostic to the support samples. Thus, the cluster centers won't introduce the biased information. Then the cluster centers are assigned labels based on a global optimization strategy, which can reduce the incorrect assignment.

To secure discriminability of the augmented samples, a compactness constraint (Zhu et al., 2019) is imposed to the generated features to encourage high intra-class compactness of each novel class. The compactness is defined as the distance between the prototypes and the generated features. With the prototypes as anchors, the generated features will distribute around to the "center" of their respective classes. We train the model in an end-to-end fashion and find that these cooperative losses can efficiently enhance the discriminative power of the generated features. Finally, we feed the base split samples into the trained networks including encoder, transformation network and decoder in turn, to generate features for the novel classes.

Our STDA can effectively produce diverse and discriminative features. The main contributions are summarized as follows: (1) We present a semantic transformation based data augmentation approach to transfer samples from training dataset to novel tasks in an encoder-decoder paradigm. (2) We propose a shortest path optimization strategy to rectify the prototypes by incorporating the cluster centers of the query samples with the basic prototypes computed on the support set, which can produce more representative prototypes. (3) We impose a compactness constraint to the generated features with the prototypes working as the anchor on visual space, which maximizes the compatibility between the generated features and the novel support features of respective classes. (4) The exhaustive ablation studies and extensive experiments on four benchmarks validate the efficacy of our data augmentation method and the generalization ability to other few shot learning baselines.

2 Related work

Meta learning is an effective few shot learning paradigm that leverages experiences from similar tasks based on the episodic formulation. Previous researches on few shot learning are made towards the following aspects. (1) *Metric learning methods* aim to learn a good metric space, where test samples can be classified via a simple metric rule. Works include nearest neighbor classifier with Euclidean (Snell et al., 2017) or cosine distance (Vinyals et al., 2016), learnable relation network-based method (Sung et al., 2018) and task specific metric-based method for respective task (Oreshkin et al., 2018). (2) *Gradient based methods* aim to learn models that can generalize well to unseen tasks with limited supervisions. Methods, such as MTL (Sun et al., 2019), MAML (Finn et al., 2017) and LEO (Rusu et al., 2018) fall into this category. (3) *Model-based methods* rely on the properties of specific model architectures, such as recurrent and memory-augmented networks (Mishra et al., 2018; Munkhdalai & Yu, 2017; Santoro et al., 2016).

Data augmentation tries to synthesize data or features by learning a generative model to alleviate the data insufficiency problem. Previous methods usually apply content preserving transformation techniques on the input images, which fail to secure the diversity of the generated samples. Recently, instead of synthesizing new image instances, a number of effective augmentation methods have been proposed in visual or semantic feature levels. Hallucination based methods (Wang et al., 2018; 2019; Hariharan & Girshick, 2017; Schwartz et al., 2018; Chen et al., 2019; Yang et al., 2020) augment the training data by hallucinating additional examples for the unseen classes. Hariharan et al. (Hariharan & Girshick, 2017) present a way of "hallucinating" additional examples for novel classes by transferring modes of variation from the base classes. In (Schwartz et al., 2018), Schwartz et al. propose an autoencoder to augment samples by encoding the intra-class deformations. Chen et al. (Chen et al., 2019) propose to directly synthesize instance features by leveraging semantics using a novel auto-encoder network. In (Yang et al., 2020), Yang et al. introduce a distribution calibration strategy to generate features for the novel tasks, which does not need extra learnable parameters. However, this method requires lots of augmented samples to achieve the improved performance, while our method does not demand many augmented samples, which greatly reduces the computational burden and computational time. Our STDA augments features based on the semantic transformation, which won't be limited by the few labeled samples. A simple classifier trained with the features generated by our STDA method along with the support data can achieve considerable performance.

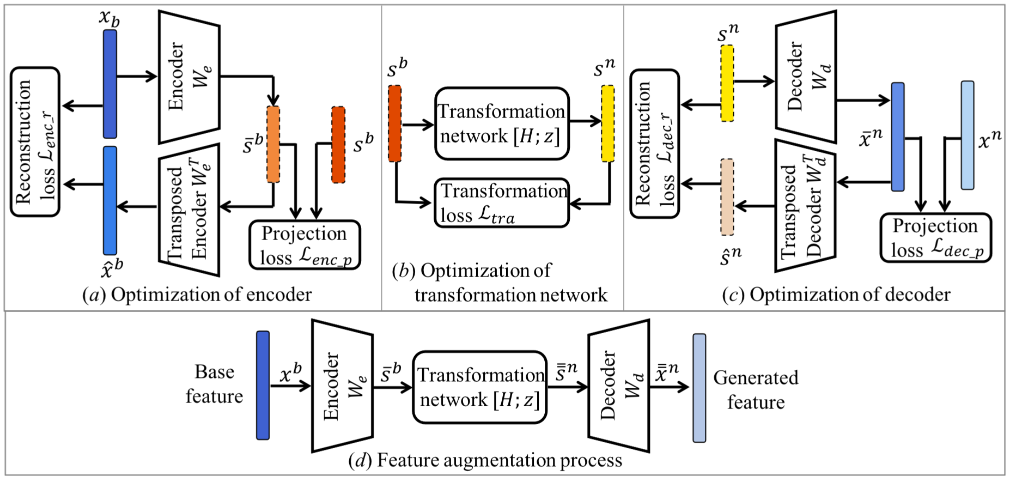
3 Method

3.1 Few shot setting

In few-shot learning, the standard N-way K-shot episodic training paradigm is adopted to learn the model. We define the training dataset as $\mathcal{D}_{base} = \left\{(x_i^b, y_i^b)|y_i^b \in \mathcal{C}_{base}\right\}$, which contains abundant labeled samples. Also, we define the test dataset as $\mathcal{D}_{novel} = \{(x_i^n, y_i^n)|y_i^n \in \mathcal{C}_{novel}\}$, which is used during the test stage. Categories of $\mathcal{D}_{novel}$ and $\mathcal{D}_{base}$ are disjoint from each other, i.e. $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \emptyset$. The model is trained and tested in task-wise manner, where each task consists a support set $\mathcal{S} = \{(x_i^n, y_i^n)\}_{i=1}^{N\times K}$ and a query set $\mathcal{Q} = \{(x_i^n)\}_{i=1}^{N\times L}$. $x_i \in \mathbb{R}^d$ is the feature vector of an image and y_i is the class label of x_i. The goal of our method is to augment the support set by transferring samples from base dataset to the novel tasks. Then a classifier can be trained with the augmented samples and the support data. Finally, the classification performance is evaluated over lots of testing tasks.

3.2 Framework Overview

The framework of our STDA approach is illustrated in Figure 2. The key idea is to transfer base data to novel tasks through semantic space in a task adaptive manner to augment support set. This process including three steps. Firstly, an encoder projects the visual features of base classes to semantic space. Then the transformation network transfer semantic features from base classes to novel classes. Afterwards, a decoder projects the semantic features back to visual space. To preserve all the information contained in the original visual features and generate semantically meaningful visual features, we impose a reconstruction constraint, that is modeled as the ridge regression loss (Kodirov et al., 2017) to the encoder and decoder. The ground truth semantic features of the novel classes work as the anchor for the visual to semantic projection. These three modules are trained in an end-to-end manner. With the trained networks, we feed samples of base classes through the encoder, transformation network and decoder in turn to generate extra samples for the novel classes.


x_b
Encoder W_e
Reconstruction loss $\mathcal{L}_{enc\_r}$
$\bar{s}^b$
s^b
Transposed Encoder W_e^T
Projection loss $\mathcal{L}_{enc\_p}$
$\hat{x}^b$
s^b
s^n
Transformation network $[H; z]$
Transformation loss $\mathcal{L}_{tra}$
s^n
Decoder W_d
Reconstruction loss $\mathcal{L}_{dec\_r}$
$\bar{x}^n$
x^n
Transposed Decoder W_d^T
$\hat{s}^n$
Projection loss $\mathcal{L}_{dec\_p}$
Base feature
x^b
Encoder W_e
$\bar{s}^b$
Transformation network $[H; z]$
$\bar{\bar{s}}^n$
Decoder W_d
$\bar{\bar{x}}^n$
Generated feature

(a) Optimization of encoder

(b) Optimization of transformation network

(c) Optimization of decoder

(d) Feature augmentation process

Figure 2: The framework of our STDA method. x^b and s^b are the visual and semantic features from the selected base classes, while x^n and s^n are the ones from the novel classes. $\hat{x}^b$ and $\bar{s}^b$ denote the reconstructed visual feature and the projected semantic feature, while $\bar{x}^n$ and $\hat{s}^n$ are the projected visual feature and the reconstructed semantic feature. $\bar{\bar{s}}^n$ is the transformed semantic feature and $\bar{\bar{x}}^n$ is the generated novel visual feature.

3.2.1 Base classes selection based on semantic similarity

Note that the semantic features in base dataset are denoted as $S^b = \left\{ s_j^b | j = 1, ..., |\mathcal{C}_{base}| \right\}$, and that in a novel task are denoted as $S^n = \{ s_i^n | j = 1, ..., N \}$. Semantic feature is the word embedding of the class label, which is computed with GloVe method. We first measure the cosine similarity between the base and novel semantic features as $p(i,j) = \frac{s_i^n s_j^b}{||s_i^n|| \cdot ||s_j^b||}$, where $i = 1, ..., N, j = 1, ..., |\mathcal{C}_{base}|$, s_j^b is the semantic feature from j^{th} base class. We select base class with the greatest similarity to the novel class. The selected classes are denoted by $I = \left\{ k_i = \underset{j=1,...,|\mathcal{C}_{base}|}{\arg\max} \; p(i,j) | i = 1, ..., N \right\}$. Then, we select K' samples per class for the novel task from their similar base classes.

$$X^{b_S} = \left\{ \bigcup_{j \in I} (x_k^b, y_k^n) | x_k^b \in select(\mathcal{D}_{base}[j], K'), y_k^n = index(I, j) \right\} \tag{1}$$

where $select(\cdot, \cdot)$ is a sample selection operator, $index(I, j)$ is an operation to return the order of j in I, i.e. the corresponding label of the base class j in the novel task. $|X^{b_S}| = N \times K'$. To encourage diversity of the generated samples, we select samples in a random way. We denote the semantic features of the selected base classes as $S^{b_N} = \left\{ s_i^b \middle| i \in I \right\}$.

3.2.2 Visual-semantic encoder formulation

We formulate the visual-to-semantic projection function as a regression loss, and denote x_i^b as the visual feature of i^{th} sample from X^{b_S} and s_i^b as the ground truth semantic feature of x_i^b. To ensure the encoder project the visual features to their respective corresponding semantic features and preserve all the information contained in the original visual features, we impose a projection constraint and a reconstruction constraint to the encoder. The constraints are formulated as $\mathcal{L}_{enc}$.

$$\min_{W_e} \mathcal{L}_{enc} = \sum_{i=1}^{N \times K'} ||s_i^b - \bar{s}_i^b||_2^2 + ||x_i^b - \hat{x}_i^b||_2^2 \;\; s.t. \;\; \bar{s}_i^b = W_e x_i^b \, , \; \hat{x}_i^b = W_e^T \bar{s}_i^b \tag{2}$$

where $\bar{s}_i^b$ is the projected semantic feature and $\hat{x}^b$ is the reconstructed visual feature, W_e denotes the trainable parameter of the encoder and W_e^T is the transposed one. By training with weight tying (Ranzato et al., 2007), the desire is to learn a set of weights where the property $W_e^T = W_e^\dagger$ is true.

3.2.3 Semantic transformation from base to novel classes

In this section, we aim to learn a transformation network $g(\cdot; [H, z])$: $s^b \rightarrow s^n$, which is optimized by minimizing the Euclidean distance between the novel semantic features s^n and the generated semantic features $\bar{s}^n$. We denote this objective as the transformation loss.

$$\begin{aligned} &\min_{H;z} || \, s^n - \bar{s}^n||_2^2 \quad s.t. \quad \bar{s}^n = Hs^b + z \\ &\mathcal{L}_{tra} = || \, s^n - \bar{s}^n||_2^2 \end{aligned} \tag{3}$$

3.2.4 Semantic-visual decoder formulation

The decoder employs similar optimization process as the encoder. The difference is that the decoder reverses the projection direction: from semantic space to visual space. Given a support set $\mathcal{S} = (x_i^n, y_i^n)_{i=1}^{N\times K}$ and the corresponding semantic feature set $S^n = \{\, s_i^n |\, i = 1, ..., N\}$, we project the semantic features to visual space with the support features as the reference points. The decoder is constrained by minimizing the distance between the projected visual features $\bar{x}^n$ and the novel visual features x^n from support set. To ensure the decoder generate semantically meaningful visual features, we further impose a reconstruction constraint to the decoder by minimizing the distance between the ground truth semantic features s^n of the novel classes and the reconstructed semantic features $\hat{s}^n$. These constraints are formulated as $\mathcal{L}_{dec}$.

$$\min_{W_d} \mathcal{L}_{dec} = \sum_{i=1}^{N\times K} ||x_i^n - \bar{x}_i^n||_2^2 + ||s_i^n - \hat{s}_i^n||_2^2 \;\; s.t. \;\; \bar{x}_i^n = W_d s_i^n, \hat{s}_i^n = W_d^T \bar{x}_i^n \tag{4}$$

where W_d is the parameter of the decoder and W_d^T denotes the transposed one. We consider tied weights (Kodirov et al., 2017) for the decoder, which has the property of $W_d^T = W_d^\dagger$.

3.2.5 Compactness constraint to the generated features

Given a novel task $\mathcal{T} = \{\mathcal{S}, \mathcal{Q}\}$, where $\mathcal{S} = (x_i^n, s_i^n, y_i^n)_{i=1}^{N\times K}$, we select N base classes for the novel task by the way in section 3.2.1. The selected dataset is denoted as $X^{b_S} = \left\{(x_i^b, y_i^n)\right\}_{i=1}^{N\times K'}$. We generate novel visual features $\bar{\bar{x}}_i^n$ from X^{b_S} and augment the support set as follow:

$$\bar{\bar{x}}_i^n = W_d(HW_e x_i^b + z) \qquad \mathcal{S}_{aug} = \{(\bar{\bar{x}}_i^n, y_i^n)\}_{i=1}^{N\times K'} \tag{5}$$

where K' is the number of generated features for each novel class. In order to get discriminative features, we impose a compactness constraint to the generated features, which is formulated as follow:

$$\mathcal{L}_{cpt} = \frac{1}{|X^{b_S}|} \sum_{j=1}^{N} \sum_{y_i=j} ||\bar{\bar{x}}_i^n - m_j||_2^2 \tag{6}$$

where m_j denotes the prototype of class j, which works as the reference point for the generated features. This constraint encourages the generated features of the same class gathering around the class centers, which improves the discrimination of the generated features.

3.2.6 Prototype rectification

Our method can be implemented in inductive and transductive settings. For the inductive setting, we compute the prototypes by averaging the labeled features of each class. For the transductive setting, we propose a novel prototype rectification method by incorporating the cluster centers of the query samples with the basic prototypes computed from support set. Firstly, the query samples are clustered into N clusters with a simple clustering algorithm $K-$means. Since clustering is an unsupervised classification method, the clusters themselves cannot tell which category they belong to. To infer class label for each cluster, we model the label assignment problem as the path planning problem, which is formulated as 7.

$$min \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{ij} \qquad s.t. \begin{cases} \sum_{i=1}^{N} x_{ij} = 1, \quad j = 1, ..., N \\ \sum_{j=1}^{N} x_{ij} = 1, \quad i = 1, ..., N \end{cases} \tag{7}$$

where $d_{ij} = ||m_i - c_j||_2^2, i = 1, ..., N, j = 1, ..., N$ is a distance matrix between the basic prototypes and the cluster centers. x_{ij} is a binary decision variable that decidedes which class the cluster belongs to. m_i denotes the i^{th} basic prototype and c_j denotes the j^{th} cluster center. The objective is to minimize the total distance between the matched prototypes and cluster centers. The constraints ensure that each class label must be only assigned to one cluster center. This model can produce global optimal solution for the label assignment, thus avoid the situation that two or more clusters share a common class label due to the close distribution of samples from different classes. Then we compute the rectified prototype of k^{th} class as follow:

$$\hat{m_k} = \frac{m_k + c_{I(x[k,:]==1)}}{2} \tag{8}$$

where $I(\cdot)$ is a index function to return the position of the nonzero value. This prototype rectification method can reduce the bias caused by limited labeled samples and further improve the quality of the generated features.

3.3 Train the STDA model and the classifier

Overall, the proposed model is trained in an end-to-end manner with the joint loss as follow:

$$\mathcal{L} = \mathcal{L}_{cpt} + \lambda_1 \mathcal{L}_{enc} + \lambda_2 \mathcal{L}_{tra} + \lambda_3 \mathcal{L}_{dec} \tag{9}$$

where the balance factors λ_1, λ_2 and λ_3 are consistently set to 1 in all experiments. Now that the feature generation model has been trained, it is used to generate features to augment the support set, see Eq. 5. Then we train a task-specific classifier based on the generated features and the original support set. The training task follows the standard cross-entropy loss optimization:

$$\min_{\theta} \sum_{(x,y) \in \mathcal{S} \cup \mathcal{S}_{aug}} -log\ p(y|x;\theta) \tag{10}$$

where θ is the parameter of the classifier. The trained classifier is then used to predict labels for the query samples. We summarize the process of our method in Algorithm 1, see Appendix A.

4 Experiments

4.1 Experimental setup

Dataset We evaluate our semantic transformation-based data augmentation approach on MiniImageNet (Ravi & Larochelle, 2016), TieredImageNet (Ren et al., 2018), Caltech-UCSD Birds 200 (Welinder et al., 2010) (CUB) and CIFAR-FS (Chen et al., 2018). More details about these benchmarks are provided in the supplementary material, see Appendix B.

Network Architecture We use two popular networks: the ResNet12 and WideResNet (WRN-28-10) (Zagoruyko & Komodakis, 2016) to extract features for all the datasets. We first pretrain the feature extractor on the base dataset. The trained extractor outputs 640-dimensional visual feature vectors for the images. The encoder is constructed with a fully connected layer without the bias term, the transformation network is modeled as a fully connected layer, and the decoder is implemented as a fully connected layer without the bias term as well.

4.2 Results and analysis

4.2.1 Comparison to the state-of-the-art methods

To evaluate the performance of our STDA method, we conduct a comparison of our approach to state-of-the-art methods. Table 1 and Table 2 present the averaged classification accuracies over 600 novel tasks on 5-way 1-shot and 5-way 5-shot cases, with each task containing 15 randomly sampled query samples. We take the logistic regression (LR) and support vector machine (SVM) as the classifiers to show the generalizability of our STDA to different classifiers. The comparison results are shown in Figure 7, see Appendix G. From Table 1, Table 2 and Figure 7, we can observe that our proposed STDA performs better than the state-of-the-art few-shot classification methods, consistently across all few-shot benchmarks, classifiers, improving over both inductive and transductive settings. The performance gain is especially considerable in the $1-$shot case, which demonstrates that our data augmentation method is fairly reliable and can handle extremely low-shot classification problem better.

Table 1: The few-shot classification accuracy results on MiniImageNet and TieredImageNet datasets. They present the mean accuracy on 600 novel episodes with a 95% confidence interval. **In.** and **Tran.** denote the inductive and transductive settings, respectively. "$-$" signifies the result is unavailable. Semantic-based methods are marked with *.

Model	Backbone	MiniImageNet 5-way		TieredImageNet 5-way	
		1-shot	5-shot	1-shot	5-shot
In.					
ProtoNet (Snell et al., 2017)	ResNet12	60.37 ± 0.83	78.02 ± 0.57	65.65 ± 0.92	83.40 ± 0.65
Meta Navigator (Zhang et al., 2021b)	ResNet12	67.14 ± 0.80	83.82 ± 0.51	74.58 ± 0.88	86.73 ± 0.61
AM3-TADAM* (Xing et al., 2019)	ResNet12	65.30 ± 0.49	78.10 ± 0.36	69.08 ± 0.47	82.58 ± 0.31
TriNet* (Chen et al., 2019)	ResNet18	58.12 ± 1.37	76.92 ± 0.69	–	–
MultiSem* (Schwartz et al., 2022)	Dense-121	67.30	82.10	–	–
FSLKT* (Peng et al., 2019)	ConvNet128	64.42 ± 0.72	74.16 ± 0.56	–	–
PSST (Chen et al., 2021)	WRN-28	64.16 ± 0.44	80.64 ± 0.32	–	–
CA (Afrasiyabi et al., 2020)	WRN-28	65.92 ± 0.60	82.85 ± 0.55	74.40 ± 0.68	86.61 ± 0.59
MetaQDA (Zhang et al., 2021c)	WRN-28	67.83 ± 0.64	84.28 ± 0.69	74.33 ± 0.65	89.56 ± 0.79
DC+LR (Yang et al., 2020)	WRN-28	68.57 ± 0.55	82.88 ± 0.42	**78.19 ± 0.25**	**89.90 ± 0.41**
STDA+LR*	WRN-28	**76.98 ± 0.72**	**86.62 ± 0.51**	77.80 ± 0.82	89.58 ± 0.56
STDA+SVM*	WRN-28	**76.18 ± 0.71**	**86.48 ± 0.47**	77.38 ± 0.84	89.05 ± 0.57
Tran.					
ICI (Wang et al., 2020)	ResNet12	72.39 ± 0.62	83.27 ± 0.33	77.48 ± 0.62	86.84 ± 0.36
POODLE (Le et al., 2021)	ResNet12	77.56	85.81	79.67	86.96
ProtoComNet* (Zhang et al., 2021a)	ResNet12	73.13 ± 0.85	82.06 ± 0.54	81.04 ± 0.89	87.42 ± 0.57
SSR (Shen et al., 2021)	WRN-28	72.40 ± 0.60	80.20 ± 0.40	79.50 ± 0.60	84.80 ± 0.40
AIM (Lee et al., 2021)	WRN-28	71.22 ± 0.57	82.25 ± 0.34	–	–
LaplacianShot (Ziko et al., 2020)	WRN-28	74.86 ± 0.19	84.13 ± 0.14	80.18 ± 0.21	87.56 ± 0.15
TIM (Boudiaf et al., 2020)	WRN-28	77.80	87.40	82.10	89.80
BD-CSPN (Liu et al., 2020)	WRN-28	70.31 ± 0.93	81.89 ± 0.60	78.74 ± 0.95	86.92 ± 0.63
STDA+LR*	WRN-28	**80.46 ± 0.72**	**87.54 ± 0.49**	**82.26 ± 0.86**	**89.98 ± 0.55**
STDA+SVM*	WRN-28	**80.21 ± 0.71**	86.77 ± 0.47	81.52 ± 0.81	**90.35 ± 0.50**

4.2.2 Impact of the number of generated features

Figure 3 shows the analysis about the classification accuracy in terms of the number of generated features for each class during test. We can observe that the performance keeps boosting with the increase of the number of generated features at the beginning, and then it turns stable. The linear regression is used as the baseline method. The baseline performance from $2-$shot to $11-$shot works as the upper bound of our method with the generated samples from 1 to 10 (along with the support data, there are 2 to 11 samples per class) at the $1-$shot case, while for the $5-$shot case, the baseline performance from $6-$shot to $15-$shot is the upper bound of our method with the generated samples from 1 to 10 (along with the support data, there are 6 to 15 samples per class). It can be noticed that, when one extra sample is added, our method achieves almost the $(K+1)-$shot performance on the baseline classifier, K is the number of labeled samples per class on support set. To further prove that our data augmentation method can indeed work, we design an experiment named "Without STDA", which borrows data directly from base classes that are similar to the corresponding novel classes without any transformation. As the number of base samples added to the novel classes increases, the performance degrads drastically (show in blue curve), which indicates that adding samples from similar base classes directly will descend the performance of the classifier. We list the quantitative results on Table 5 show in Appendix D. To qualitatively demonstrate the effectiveness of our approach, we show the t-SNE visualization (Van der Maaten & Hinton, 2008) of the generated features in Figure 4, see Appendix E. It can be observed that the generated features exhibit clear

Table 2: The few-shot classification accuracy results on CIFAR-FS and CUB datasets.

Model	Backbone	CIFAR-FS 5-way		CUB 5-way	
		1-shot	5-shot	1-shot	5-shot
In.					
ProtoNet (Snell et al., 2017)	ResNet12	72.20 ± 0.70	83.50 ± 0.50	71.90	87.40
Meta Navigator (Zhang et al., 2021b)	ResNet12	74.63 ± 0.91	86.45 ± 0.59	–	–
CA (Afrasiyabi et al., 2020)	ResNet18	–	–	74.22 ± 1.09	88.65 ± 0.55
TriNet* (Chen et al., 2019)	ResNet18	–	–	69.61 ± 0.46	84.10 ± 0.35
MultiSem* (Schwartz et al., 2022)	Dense-121	–	–	76.10	82.90
PSST (Chen et al., 2021)	WRN-28	77.02 ± 0.38	88.45 ± 0.35	–	–
MetaQDA (Zhang et al., 2021c)	WRN-28	75.83 ± 0.88	88.79 ± 0.75	–	–
DC+LR (Yang et al., 2020)	WRN-28	–	–	79.56 ± 0.87	90.67 ± 0.35
STDA+LR*	WRN-28	**80.60 ± 0.76**	**88.89 ± 0.54**	**82.50 ± 0.77**	**91.49 ± 0.42**
STDA+SVM*	WRN-28	**79.64 ± 0.83**	87.86 ± 0.57	**83.37 ± 0.69**	**91.10 ± 0.42**
Tran.					
ICI (Wang et al., 2020)	ResNet12	79.19 ± 0.63	86.66 ± 0.36	–	–
ProtoComNet* (Zhang et al., 2021a)	ResNet12	–	–	**93.20 ± 0.45**	**94.90 ± 0.31**
LaplacianShot (Ziko et al., 2020)	ResNet18	–	–	80.96	88.68
TIM (Boudiaf et al., 2020)	ResNet18	–	–	82.20	90.80
SSR (Shen et al., 2021)	WRN-28	81.60 ± 0.60	86.0 ± 0.40	–	–
AIM (Lee et al., 2021)	WRN-28	80.20 ± 0.55	87.34 ± 0.36	–	–
STDA+LR*	WRN-28	**84.47 ± 0.74**	**89.30 ± 0.53**	84.48 ± 0.72	91.90 ± 0.42
STDA+SVM*	WRN-28	**84.18 ± 0.80**	**88.56 ± 0.55**	84.95 ± 0.71	91.32 ± 0.41

clustering structure, and is compatible with the support and query samples. We also compare the rectified prototypes (denoted as the transductive prototypes) with the support prototypes (denoted as the inductive prototypes) in the last column of Figure 4. It can be observed that the transductive prototypes exhibit closer to the centers of the query set than the inductive ones, demonstrating the effectiveness of our prototype rectification method.

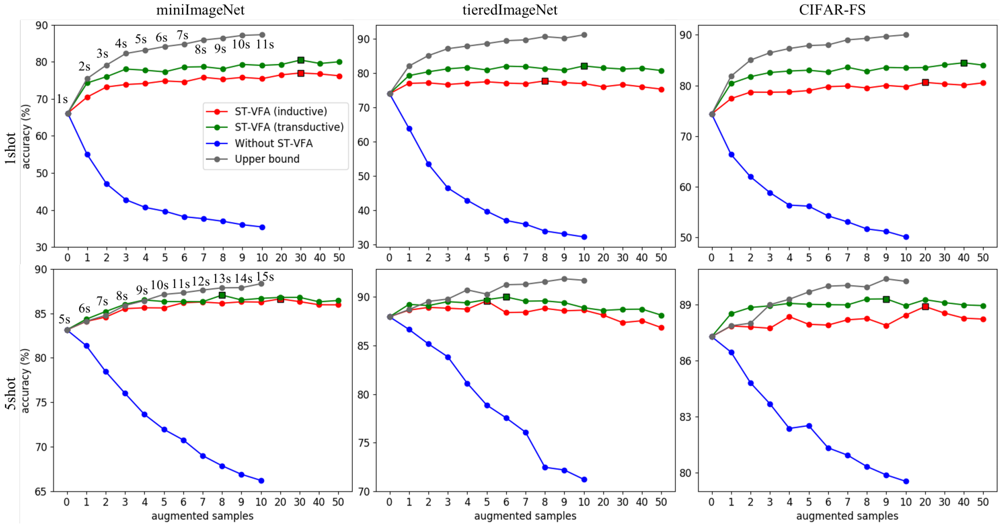


Figure 3: Illustration of the performance change with varied number of augmented samples on MiniImageNet, TieredImageNet and CIFAR-FS datasets. The square markers with black edge denote the best performance points. All the classification is implemented with the linear regression (LR) classifier.

4.2.3 Application of our STDA on other baselines

To further test the generalization of our data augmentation technique on other baseline methods, we apply our STDA on two few-shot classification algorithms: prototypical network (Snell et al., 2017) and MTL (Sun et al., 2019). Results in Table 3 show that the baseline methods equipped with our STDA technique outperform the baselines by significant margins, and the boosting in the 1-shot case is especially considerable, which demonstrates that the baseline methods can benefit from training with the augmented data generated by our STDA method.

Table 3: Performance comparison of baselines with and without STDA technique on MiniImageNet.

Backbone	Methods	Without STDA 5-way		With STDA 5-way	
		1-shot	5-shot	1-shot	5-shot
ResNet12	ProtoNet (Snell et al., 2017)	61.35 ± 0.87	79.54 ± 0.58	**63.99 ± 0.82**	**80.17 ± 0.54**
	MTL (Sun et al., 2019)	60.82 ± 0.80	75.96 ± 0.62	**63.18 ± 0.80**	**76.92 ± 0.65**
WRN-28	ProtoNet (Snell et al., 2017)	65.50 ± 0.81	82.95 ± 0.56	**76.45 ± 0.73**	**86.62 ± 0.49**
	MTL (Sun et al., 2019)	65.57 ± 0.80	79.76 ± 0.60	**73.69 ± 0.78**	**83.09 ± 0.56**

4.2.4 ABLATION STUDY

Ablation study on the constraints. To study the impact of each constraint on the final performance, we ablate the loss term: the reconstruction loss of encoder $\mathcal{L}_{enc\_r}$, the reconstruction loss of decoder $\mathcal{L}_{dec\_r}$, the compactness loss $\mathcal{L}_{cpt}$ and all these three constraints, respectively. The projection constraints of the encoder and decoder are the skeletons of our framework, which cannot be ablated. From Table 4, we can observe performance decline when any constraint is ablated. The degradation is especially severe in the $1-$shot case. We further qualitatively analyze the impact of the constraints on the model by showing the t-SNE visualization of the generated features under ablation study. Results shown in Figure 5 indicate that when any constraint is ablated, there is a deviation between the generated features and the support features of the novel task. This phenomenon is in consistent with the quantitative results in Table 4.

Table 4: Ablation study on the constraints for both inductive and transductive settings.

$\mathcal{L}_{enc\_r}$	$\mathcal{L}_{dec\_r}$	$\mathcal{L}_{cpt}$	MiniImageNet 5-way			
			In		Tran	
			1-shot	5-shot	1-shot	5-shot
✓	✓	✓	**76.98 ± 0.72**	**86.62 ± 0.51**	**80.46 ± 0.72**	**87.54 ± 0.49**
	✓	✓	71.46 ± 0.77	84.66 ± 0.51	75.93 ± 0.85	85.66 ± 0.54
✓		✓	76.03 ± 0.84	85.80 ± 0.46	78.70 ± 0.78	86.23 ± 0.47
✓	✓		69.13 ± 0.82	84.53 ± 0.51	74.91 ± 0.90	85.94 ± 0.57
			64.45 ± 0.80	82.71 ± 0.56	64.70 ± 0.80	83.10 ± 0.54

Ablation study on the networks and balancing parameters. To disentangle the impact of different depth networks for the modules on the STDA model, we further try a non-linear deeper network for the encoder, transformation network and decoder, respectively. Results shown in Table 7 demonstrate that the non-linear network performs worse than the linear one. To select the best hyperparameters for the objective function Eq. 9, we carry out an ablation study on the balancing parameters λ_1, λ_2 and λ_3. The plots shown in Figure 8 demonstrate that 1.0 is the best value for these three parameters. The detailed implementations are described in Appendix H and Appendix I, respectively.

Running time comparison. To show the efficiency of method, we record the running time of STDA at different number of generated samples, and compare that with DC (Yang et al., 2020) in Table 8 (see Appendix J). Results show that our method is time-saving for both 1-shot and 5-shot cases.

5 CONCLUSION

In this paper, we propose a simple data augmentation framework to solve the extremely low-shot classification problem. Our STDA method can effectively transfer samples from base dataset to the novel task based on semantic transformation between similar base and novel classes. Since the prototypes of the novel task are worked as anchor for the generated samples, we further introduce a shortest path optimization based prototype rectification approach by incorporating the cluster centers of the query set with the basic prototypes of support set to reduce the bias. Finally, a simple classifier is adopted to predict labels for the test samples. It is trained with the support samples along with the features generated by STDA method. The exhaustive ablation studies and comparison experiments show our method outperforms the state-of-the-arts by significant margins, which proves the generalization and effectiveness of our method.

REFERENCES

Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In *European Conference on Computer Vision*, pp. 18–35. Springer, 2020.

Han Altae-Tran, Bharath Ramsundar, Aneesh S Pappu, and Vijay Pande. Low data drug discovery with one-shot learning. *ACS central science*, 3(4):283–293, 2017.

Malik Boudiaf, Imtiaz Ziko, Jérôme Rony, José Dolz, Pablo Piantanida, and Ismail Ben Ayed. Information maximization for few-shot learning. *Advances in Neural Information Processing Systems*, 33:2445–2457, 2020.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2018.

Zhengyu Chen, Jixie Ge, Heshen Zhan, Siteng Huang, and Donglin Wang. Pareto self-supervised training for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13663–13672, 2021.

Zitian Chen, Yanwei Fu, Yinda Zhang, Yu-Gang Jiang, Xiangyang Xue, and Leonid Sigal. Multi-level semantic feature augmentation for one-shot learning. *IEEE Transactions on Image Processing*, 28 (9):4594–4605, 2019.

Terrance DeVries and Graham W Taylor. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*, 2017.

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

Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3018–3027, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Elyor Kodirov, Tao Xiang, and Shaogang Gong. Semantic autoencoder for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3174–3183, 2017.

Duong Le, Khoi Duc Nguyen, Khoi Nguyen, Quoc-Huy Tran, Rang Nguyen, and Binh-Son Hua. Poodle: Improving few-shot learning via penalizing out-of-distribution samples. *Advances in Neural Information Processing Systems*, 34:23942–23955, 2021.

Eugene Lee, Cheng-Han Huang, and Chen-Yi Lee. Few-shot and continual learning with attentive independent mechanisms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9455–9464, 2021.

Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. In *European Conference on Computer Vision*, pp. 741–756. Springer, 2020.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pp. 2554–2563. PMLR, 2017.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018.

Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. Few-shot image recognition with knowledge transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 441–449, 2019.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Marc'Aurelio Ranzato, Y-Lan Boureau, Yann Cun, et al. Sparse feature learning for deep belief networks. *Advances in neural information processing systems*, 20, 2007.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2018.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pp. 1842–1850. PMLR, 2016.

Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. *Advances in Neural Information Processing Systems*, 31, 2018.

Eli Schwartz, Leonid Karlinsky, Rogerio Feris, Raja Giryes, and Alex Bronstein. Baby steps towards few-shot learning with multiple semantics. *Pattern Recognition Letters*, 2022.

Xi Shen, Yang Xiao, Shell Xu Hu, Othman Sbai, and Mathieu Aubry. Re-ranking for image retrieval and transductive few-shot classification. *Advances in Neural Information Processing Systems*, 34: 25932–25943, 2021.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 403–412, 2019.

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

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

Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. A meta-learning perspective on cold-start recommendations for items. *Advances in neural information processing systems*, 30, 2017.

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

Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12836–12845, 2020.

Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7278–7286, 2018.

Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. Implicit semantic data augmentation for deep networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

Chen Xing, Negar Rostamzadeh, Boris Oreshkin, and Pedro O O Pinheiro. Adaptive cross-modal few-shot learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations*, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

Baoquan Zhang, Xutao Li, Yunming Ye, Zhichao Huang, and Lisai Zhang. Prototype completion with primitive knowledge for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3754–3762, 2021a.

Baoquan Zhang, Xutao Li, Shanshan Feng, Yunming Ye, and Rui Ye. Metanode: Prototype optimization as a neural ode for few-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9014–9021, 2022.

Chi Zhang, Henghui Ding, Guosheng Lin, Ruibo Li, Changhu Wang, and Chunhua Shen. Meta navigator: Search for a good adaptation policy for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9435–9444, 2021b.

Xueting Zhang, Debin Meng, Henry Gouk, and Timothy M Hospedales. Shallow bayesian meta learning for real-world few-shot recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 651–660, 2021c.

Yizhe Zhu, Jianwen Xie, Zhiqiang Tang, Xi Peng, and Ahmed Elgammal. Semantic-guided multi-attention localization for zero-shot learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Imtiaz Ziko, Jose Dolz, Eric Granger, and Ismail Ben Ayed. Laplacian regularized few-shot learning. In *International Conference on Machine Learning*, pp. 11660–11670. PMLR, 2020.

A Algorithm of STDA

Algorithm 1: Proposed STDA algorithm

```
Input: Base set $\mathcal{D}_{base}$, novel task $\mathcal{T} = \{\mathcal{S}; \mathcal{Q}\}$, iteration $T$.
Output: The generated novel features.
Select $N$ base classes for the novel task;
Sample $K'$ features from each selected base class by Eq. 1;
for $t$ in $1$ to $T$ do
    Impose projection and reconstruction constraints to the encoder by Eq. 2;
    Optimize the semantic transformation network $g(\cdot; [H, z])$ by Eq. 3;
    Impose projection and reconstruction constraints to the decoder by Eq. 4;
    Impose compactness constraint to the generated features by Eq. 6;
    Optimize parameters $\{W_e; [H, z]; W_d\}$ of the model by minimizing Eq. 9;
end
```

B Detailed description of datasets

MiniImageNet is a subset of ILSVRC-2012 (Russakovsky et al., 2015), which consists of 100 classes that are split into 64, 16 and 20 classes for training, validation and test respectively. Each class contains 600 images of size $84 \times 84 \times 3$.

TieredImageNet is a larger subset of ILSVRC-12 dataset (Russakovsky et al., 2015), which contains 608 classes sampled from hierarchical category structure. The average number of images in each class is 1281 of size $84 \times 84 \times 3$. We use 351, 97, and 160 classes for training, validation, and test, respectively.

CIFAR-FS is a dataset derived from CIFAR-100 with images of size $32 \times 32 \times 3$. It contains 100 categories with 600 instances in each class. All the classes are split into 64, 16 and 20 for training, validation and test.

Caltech-UCSD Birds 200 is a fine-grained dataset consisting of 200 classes of birds with a total number of 11,788 images of size $84 \times 84 \times 3$. Following the split in previous works (Chen et al., 2018), we use 100, 50 and 50 classes for training, validation and test, respectively. This dataset also provides 312-dimensional semantic attribute vectors on a per-class level.

C Implementation details

Pytorch is used to implement all our experiments. At the model training stage, Adam is used as the optimizer. The initial learning rate is 0.001 and decreased to 0.2 times of the previous stage per 10 steps. The model is trained for 100 iterations per task. During the test stage, we use the logistic regression (LR) and the support vector machine (SVM) as the classifier. The number of generated features for different datasets are determined by the experiment in section 4.2.2. The visual features are extracted by the pre-trained network. And the semantic features are computed by the Glove method in an average manner of the word embeddings of each class.

D Results on different number of augmented features

Table 5 presents the quantitative results of our STDA method at different number of generated features. LR denotes the linear regression method, which is used as the classifier to evaluate our data augmentation technique. The accuracies shown in the first line denote the performance of LR at different support shots, while for the other lines, they present the performance of the data augmentation based methods at different number of augmented samples. The results shown in Table 5 illustrate that our STDA method is better than DC (Yang et al., 2020) method in accuracy at different number of augmented samples. "Without STDA" is a method that we design to prove our semantic feature transformation process is the reason for the performance boosting. In this experiment, we borrow samples directly from the base classes (which is semantically similar to the corresponding novel classes) without any transformation. Results of this experiment show drastic performance degradation as the number of base samples increases, which demonstrates that the semantic feature transformation process is essential, thus proves that our data transformation technique can indeed work.

Table 5: Performance comparison on different number of augmented samples on MiniImageNet, where features are extracted with the WRN28 backbone.

Shot	Methods	Accuracy (%) on various number of generated samples/shots						
		1	5	10	20	30	40	50
1shot	LR (baseline)	66.73	83.14	86.76	89.10	90.03	90.49	90.97
	Without STDA	55.01	39.66	35.43	32.82	32.02	31.65	30.68
	DC+LR (Yang et al., 2020)	61.31	66.49	66.72	66.21	67.06	67.74	67.31
	STDA+LR(In)	70.46	74.83	75.43	76.47	76.98	76.71	76.21
	STDA+LR(Tran)	73.81	78.64	78.98	79.47	80.46	79.81	79.72
5shot	Without STDA	81.37	71.97	66.24	60.63	56.57	54.94	53.31
	DC+LR (Yang et al., 2020)	83.0	82.22	83.24	83.0	83.15	83.49	83.30
	STDA+LR(In)	84.39	85.60	86.26	86.62	86.31	85.96	85.95
	STDA+LR(Tran)	84.69	86.69	86.53	86.80	86.82	86.59	86.10

E t-SNE VISUALIZATION OF THE GENERATED FEATURES

To intuitively observe the distribution of the augmented samples, we show the t-SNE visualization of the augmented samples at different numbers. From Figure 4, we can observe that the generated features exhibit clear clustering structure, which can help train a discriminative classifier. From the last column of Figure 4 we can observe that the transductive prototypes exhibit closer to the centers of the query set than the inductive ones, which proves the effectiveness of our prototype rectification method in reducing bias.

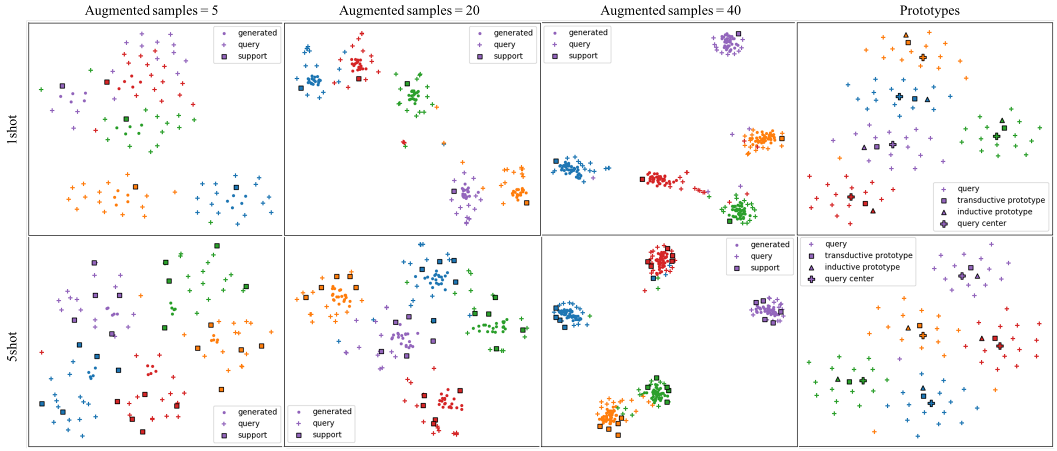


Figure 4: t-SNE visualization of the augmented samples. The transductive prototype is computed with Eq. 8, while the inductive prototype is computed by averaging samples of each class from the original support set. Features are extracted with WRN28-10 backbone.

F t-SNE VISUALIZATION OF THE GENERATED FEATURES UNDER ABLATION STUDY

We show the qualitative results of the ablation study in t-SNE visualization. As shown in Figure 5 and Figure 6, we can observe that there exists a deviation between the generated samples and the support and query samples when any constraint is ablated. As analyzed before, without the reconstruction constraint on the encoder, it will lose some important information contained in the original visual features during the projection from visual to semantic space. While without the reconstruction constraint on the decoder, it will produce semantically meaningless features. Taking a closer look at the t-SNE visualization in third line, we can observe that the features generated without the compactness loss exhibit more biased than that without the encoder or decoder reconstruction constraint, which indicates that the compactness constraint contributes more to the final performance. Furthermore, when these three constraints are removed from the optimization objective, the generated features are far away from the support and query samples. This phenomenon is in consistent with the quantitative results. The visualizations prove that these constraints can indeed help reduce the distribution bias, resulting in valid augmented samples.

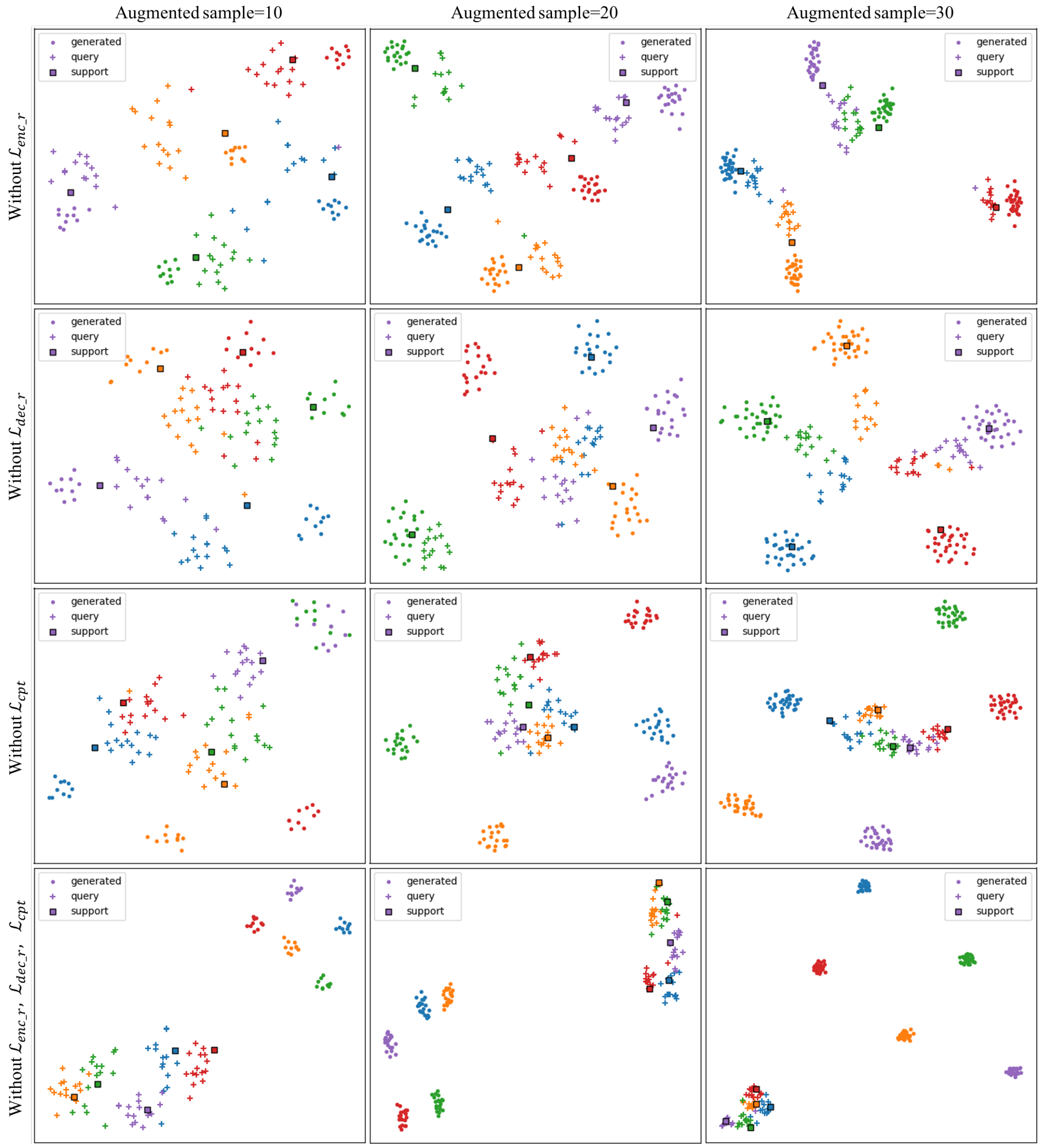


Figure 5: t-SNE visualization of the ablation study on the constraints: the reconstruction loss of encoder $\mathcal{L}_{enc\_r}$, the reconstruction loss of decoder $\mathcal{L}_{dec\_r}$, the compactness loss $\mathcal{L}_{cpt}$ and all these three constraints, respectively. Experiments are conducted on a 5-way 1-shot task from MiniImageNet dataset.

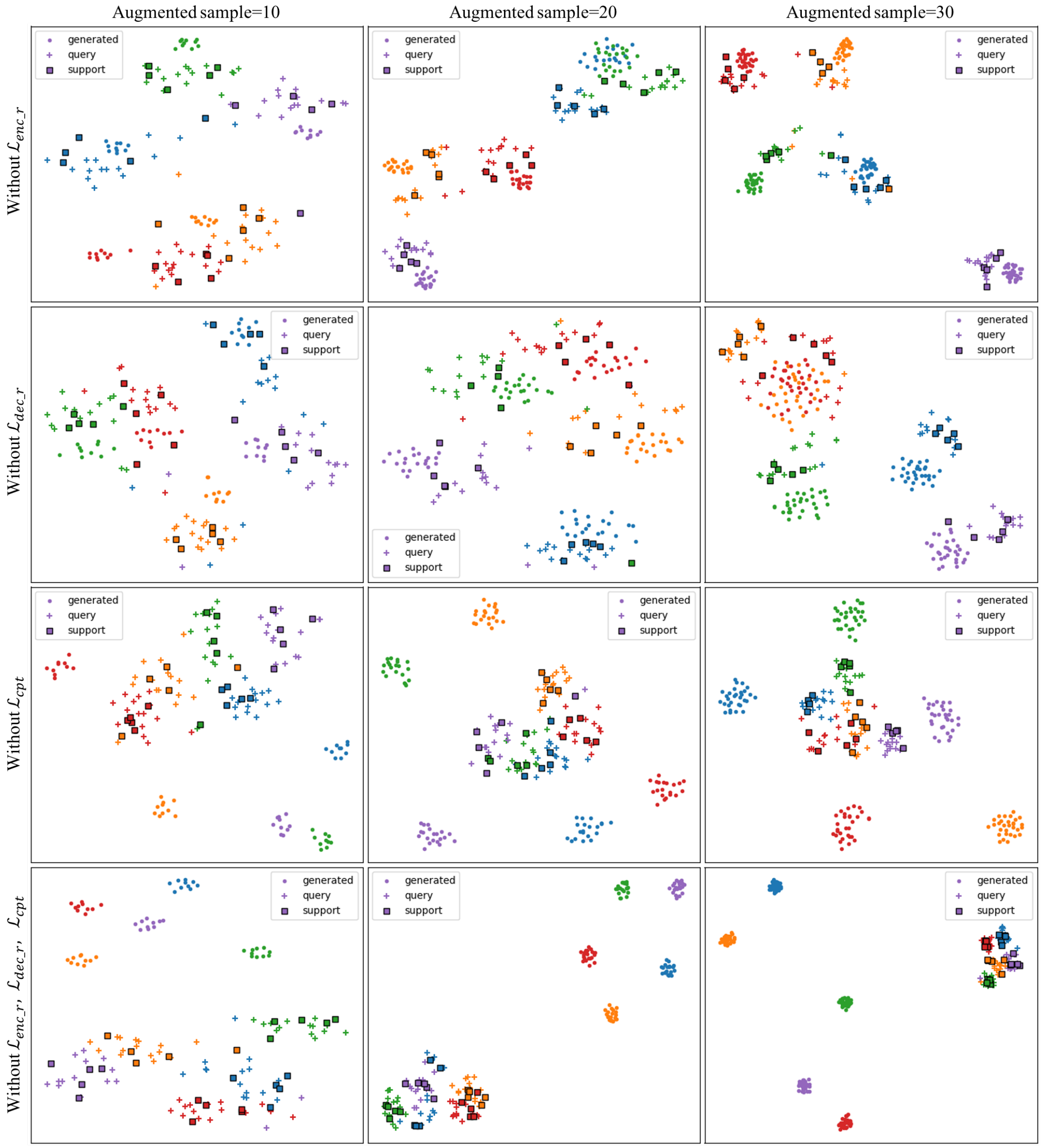


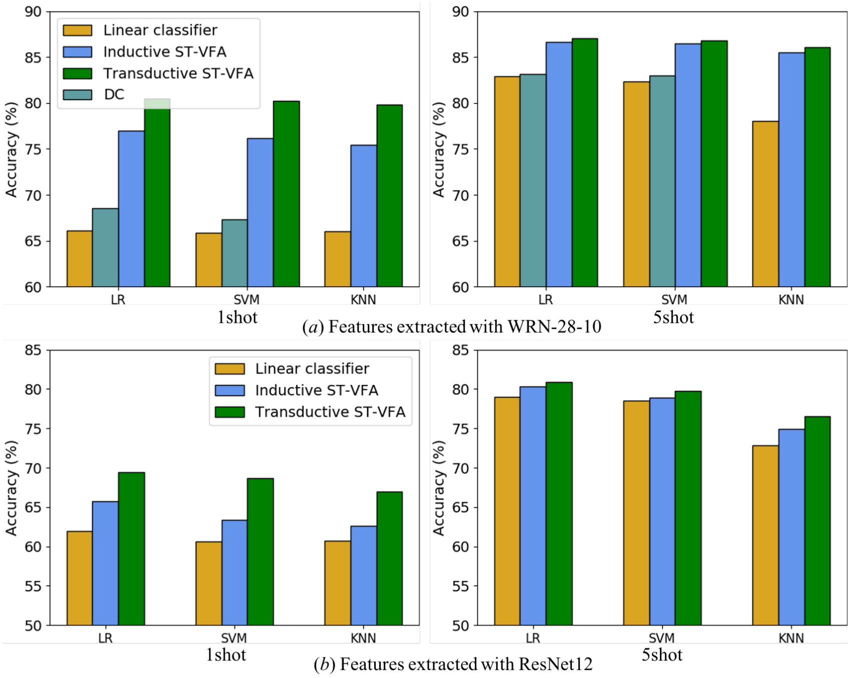
Figure 6: t-SNE visualization of the ablation study on the constraints: the reconstruction loss of encoder $\mathcal{L}_{enc\_r}$, the reconstruction loss of decoder $\mathcal{L}_{dec\_r}$, the compactness loss $\mathcal{L}_{cpt}$ and all these three constraints, respectively. Experiments are conducted on a 5-way 5-shot task from MiniImageNet dataset.

G Comparison on different classifiers

Figure 7 shows the comparison about the recognition accuracy with respect to different classifiers. Comparing with the distribution calibration method (DC), our STDA method consistently yields performance gain on different classifiers (i.e., LR, SVM and KNN), which further validates the effectiveness of our data augmentation technique. Meanwhile, We conduct an experiment on a shallower feature extractor ResNet12. Results shown in Table 6 demonstrate that our method is applicable for the shallower feature space as well.

Table 6: The few-shot classification accuracy on ResNet12 feature extractor. The superscripts denote the settings: ^{I}Inductive, ^{T}Transductive.

Model	MiniImageNet 5way		TieredImageNet 5-way		CIFAR-FS 5-way	
	1shot	5shot	1shot	5shot	1shot	5shot
LR (baseline)	61.93 ± 0.81	78.98 ± 0.59	68.58 ± 0.84	83.26 ± 0.67	67.64 ± 0.90	82.79 ± 0.65
STDA+LR^I	65.75 ± 0.71	80.36 ± 0.55	68.67 ± 0.89	83.80 ± 0.63	71.30 ± 0.84	84.12 ± 0.62
STDA+LR^T	69.40 ± 0.82	80.90 ± 0.55	71.37 ± 0.94	84.14 ± 0.67	75.56 ± 0.84	84.19 ± 0.64
SVM (baseline)	60.61 ± 0.84	78.50 ± 0.55	66.39 ± 0.95	83.26 ± 0.67	66.72 ± 0.91	81.89 ± 0.68
STDA+SVM^I	63.34 ± 0.79	78.90 ± 0.59	67.32 ± 0.94	83.53 ± 0.60	68.74 ± 0.89	82.45 ± 0.67
STDA+SVM^T	68.66 ± 0.86	79.78 ± 0.57	69.74 ± 0.92	84.25 ± 0.64	72.99 ± 0.90	83.76 ± 0.65



(*a*) Features extracted with WRN-28-10

(*b*) Features extracted with ResNet12

Figure 7: Comparison of our STDA method at different classifiers and backbones on miniImageNet dataset.

H Performance comparison of different depth networks

To explicitly reveal the impact of different depth networks on the performance, we try non-linear networks for the encoder, transformation network and the decoder, respectively, where each of them is composed of two fully connected layers and one activation function. We compare the performance of the model constructed with linear shallow networks and non-linear deeper networks. As can be seen in Table 7, the non-linear network performs worse than the linear one. Since the training data is very scarce, deeper networks will introduce more parameters, which makes the model easy to overfit the limited training data.

Table 7: Performance comparison of the model constructed with linear networks vs non-linear networks in inductive setting on MiniImageNet dataset. Features are extracted with WRN28 backbone.

Model	MiniImageNet 5way	
	1shot	5shot
STDA+LR(linear)	76.98 ± 0.72	86.62 ± 0.51
STDA+LR(nonlinear)	70.66 ± 0.75	84.32 ± 0.50

I Ablation study on the balancing parameters

To study the impact of the balancing parameters λ_1, λ_2 and λ_3 on the total optimization objective eq. 9, we conduct a variable-controlling experiment on λ_1, λ_2 and λ_3 to show the performance changing at different values. λ_1 (λ_2 and λ_3) is varied from 0.1 to 1 with 0.1 as interval, while the other two parameters are fixed to 1. From Figure 8 we can observe that the performance in terms of λ_1, λ_2 and λ_3 keeps a trend of growing as the increase of λ values in $1-$shot case. While for the $5-$shot case, the performance in terms of the varying values of λ_1 and λ_2 keeps smooth in 5-shot case. The growing trend in terms of the increasing value of λ_3 is especially obvious for both $1-$shot and $5-$shot cases. This experiment demonstrates that setting λ_1, λ_2 and λ_3 to 1.0 is reasonable.

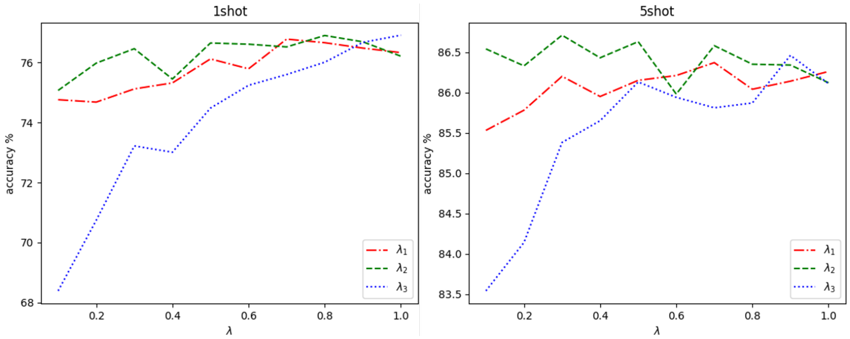


Figure 8: Performance change about the classification accuracy in terms of the hyperparameters λ_1, λ_2 and λ_3. The experiment is conducted on MiniImageNet dataset with WRN28 as the feature extractor.

J Comparison of running-time

Previous data augmentation works focus on learning data distribution, and generating a large number of samples from the distribution, which is memory and time consuming. Different from these methods, our method can achieve great performance gain by augmenting few samples, which is time saving and sample effective.

In Table 8, we record the average running time of per few-shot task for the 5-way 5-shot and 5-way 1-shot tasks on MiniImageNet. Results show that with our STDA method, the 5-shot task is a little slower than the 1-shot task, while the experiments on transductive and inductive settings require similar running-time. For the DC (Yang et al., 2020) method, it exhibits great running-time gap between the $1-$shot and $5-$shot tasks. The running-time increases as the number of augmented samples grows. Our experiments are run on the GTX TITAN X GPU.

Table 8: Per few-shot task running-time comparison for the $5-$way $1-$shot and $5-$way $5-$shot tasks on MiniImageNet.

Setting	Methods	Run-time (s) for varied number of generated samples per task						
		1	5	10	20	30	40	50
Tran	STDA+LR(1shot)	3.1	3.5	3.6	4.5	5.3	5.9	6.6
	STDA+LR(5shot)	3.6	3.9	3.6	4.8	5.5	6.0	6.6
In	STDA+LR(1shot)	3.0	3.6	3.6	4.5	5.2	5.9	6.6
	STDA+LR(5shot)	3.5	3.8	3.9	4.9	5.5	6.0	7.2
	DC+LR (Yang et al., 2020) (1shot)	3.5	3.1	3.8	3.7	3.7	4.1	3.7
	DC+LR (Yang et al., 2020) (5shot)	9.6	9.6	9.0	9.6	9.6	10.2	12.0