FSL-Rectifier: Rectify Outliers in Few-Shot Learning via Test-Time Augmentation

Yunwei Bai[1]*, Ying Kiat Tan[1], Shiming Chen[2], Yao Shu[3], Tsuhan Chen[1]

[1]National University of Singapore
[2]Mohamed bin Zayed University of Artificial Intelligence
[3]Guangdong Lab of AI and Digital Economy (SZ)
{baiyunwei, yingkiat}@u.nus.edu, shimingchen@gmail.com, shuyao@gml.ac.cn, dprtchen@nus.edu.sg

Abstract

Few-shot learning (FSL) commonly requires a model to identify images (queries) that belong to classes unseen during training, based on a few labelled samples of the new classes (support set) as reference. So far, plenty of algorithms involve training data augmentation to improve the generalization capability of FSL models, but outlier queries or support images during inference can still pose great generalization challenges. In this work, to reduce the bias caused by the outlier samples, we generate additional test-class samples by combining original samples with suitable train-class samples via a generative image combiner. Then, we obtain averaged features via an augmentor, which leads to more typical representations through the averaging. We experimentally and theoretically demonstrate the effectiveness of our method, obtaining a test accuracy improvement proportion of around 10% (e.g., from 46.86% to 53.28%) for trained FSL models. Importantly, given a pretrained image combiner, our method is training-free for off-the-shelf FSL models, whose performance can be improved without extra datasets nor further training of the models themselves.

Code —
https://github.com/WendyBaiYunwei/FSL-Rectifier-Pub

1 Introduction

Although deep learning has gained much practical success, deep neural networks (DNN) often require a large amount of data to perform well (Marcus 2018). Usually, there is a high expense for label collection, and some classes of data can be rare and practically difficult to collect (Yu et al. 2015; Marcus 2018; Wang et al. 2020; Whang et al. 2023). For example, in facial expression recognition, dataset class imbalance problems are prevalent, as happy smiley faces are generally easier to collect than facial expressions of disgust (Ciubotaru et al. 2019). Few-Shot Learning (FSL) involves classification problem tackling the issues of limited test-class labelled data. Generally, FSL models can classify unseen test classes when presented with a *support set* comprising only a few labelled test samples as reference (Snell, Swersky, and Zemel 2017; Ye et al. 2020; Sung et al. 2018; Wang et al. 2020; Dhillon et al. 2019). Nevertheless, while the data constraint, characterized by the absence of test-class data during training, is practically meaningful, it also poses generalization challenges to DNN-based FSL models (Wang et al. 2020). The challenge can be exacerbated by unconventionality of test data samples (Wang et al. 2020; Kim et al. 2019; Snell, Swersky, and Zemel 2017).

So far, different algorithms are proposed to tackle the issue of high generalization errors in FSL models. Most works involve training augmentation for enhancing the generalization capacity of FSL models (Mishra et al. 2018; Verma et al. 2018; Schwartz et al. 2018; Hariharan and Girshick 2017; Wang et al. 2018b; Gao et al. 2018). However, such augmentation involves model training over an increased amount of data, costing extra computation resources over every additional FSL model trained. Furthermore, augmentation over the training dataset can be limited in performance, since the gap between the testing data and the augmented training data may still remain wide.

In this work, we propose to tackle FSL via test-time augmentation instead of training-time augmentation. Our method is named *FSL-Rectifier*, which essentially augments the original test samples before considering the averaged augmentation during classification. For example, given a side-facing wolf test sample, we pick a suitable training sample, say a front-facing dog. Then, we convert the front-facing dog to the wolf class, producing a front-facing wolf image. A trained FSL model then makes classification based on averaged features of both the side-facing wolf and the front-facing wolf, instead of just one single side-facing wolf that may have been an outlier. According to extensive literature, feature averaging can reduce bias, effectively leading to more typical feature representations (Zhou 2012; Kimura 2024; Shanmugam et al. 2021).

To achieve this, we have three main components: 1) *image combiner*, 2) *neighbour selector* and 3) *augmentor*. For the *image combiner*, we have a generative image translator model pretrained over the training classes. The image combiner considers two images, combining the general shape (e.g., animal pose and position of eyes) of one image and the style (e.g., animal's class-defining fur style) of another via a Generative Adversarial Network (GAN) mechanism (Goodfellow et al. 2014; Liu et al. 2019a). The combination is illustrated in Figure 1(a). During testing of any off-the-

*The corresponding author.

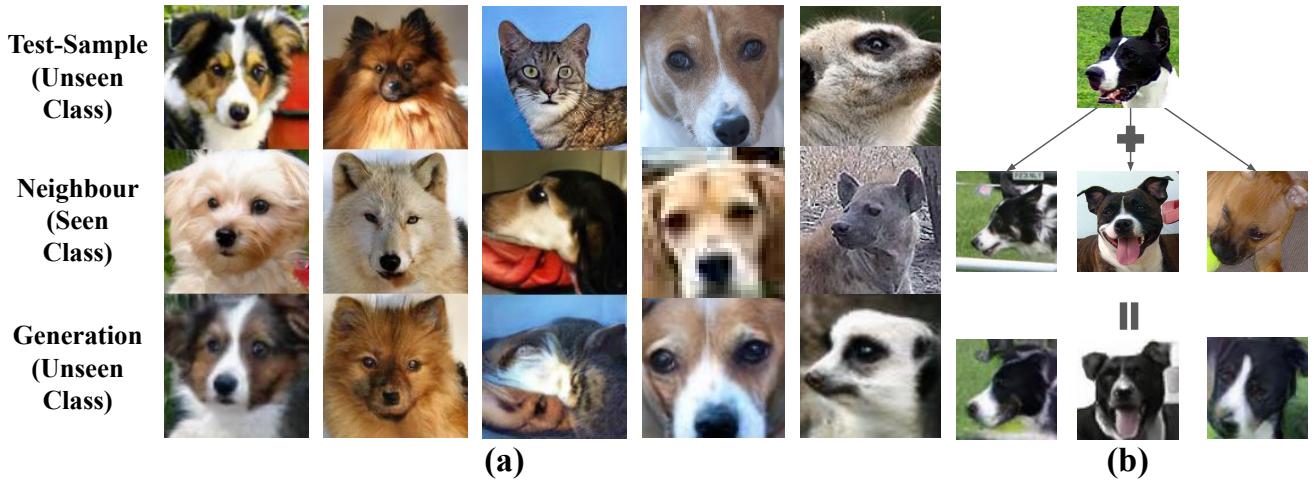

Figure 1: Illustration of our key idea. (a): The first image in each column of the animals dataset is the original test sample. The second is neighbour sample from the train class. The last image is the generation based on style of the test sample and general shape of the neighbour. (b): Given different neighbour samples, each test sample can be augmented to multiple copies.

shelf FSL models, the *neighbour selector* can select a better candidate to be combined with a test sample for generation. This test sample can either be a query to be classified, or a support set sample. Finally, through the *augmentor*, we average representations of augmented copies and the original test samples. The augmentation is illustrated in Figure 1(b). With these, our FSL-Rectifier can make the averaged representations closer to their centroids, correcting certain outlier predictions of an existing FSL model. Besides, our method is test-time-only. Given pretrained image translator, our method can be directly applied to trained FSL models.

Our contributions include the following:

- We propose a novel test-augmentation pipeline for FSL, including an image combiner, a neighbour selector and an augmentor. The image combiner can generate images based on general shape of one image and class-defining features of another. The neighbour selector picks suitable training images whose general shape is to be combined with the class-defining features of original test samples. The augmentor performs feature-averaging across the original test samples and the generated test samples, mitigating the outlier effect in original test samples.
- We conduct various experiments and analysis to verify the feasibility of our idea. On a dataset consisting of animal faces, we can achieve around 4% improvement for a trained FSL model, over the original baseline without any augmentation. The improvement is achieved despite limitation in our image generation quality.
- We formulate a theoretical framework and conduct mathematical analyses on our approach, demonstrating its high potential in reducing generalization errors in FSL.

2 Related Works

Few-shot learning sees pioneer works from Fe-Fei et al. (2003), which formulates a bayesian learning framework for quick model adaptation to novel classes. For a broader field of few-shot learning, mainstream algorithms include the meta-learning methodology and metric-learning methodology (Wang et al. 2020; Ribeiro, Singh, and Guestrin 2016; Sung et al. 2018). For example, meta-learning methods (Lee et al. 2019; Sun et al. 2019; Rusu et al. 2018; Bertinetto et al. 2019) like the MAML Finn, Abbeel, and Levine (2017) learn how to initialize model parameters for training so that the trained parameters can adapt to new unseen tasks quickly. Meanwhile, metric-learning methods like Sung et al. (2018); Snell, Swersky, and Zemel (2017) aim to learn data representations which are close together as the same class, and far apart as different classes. Related to our work, FSL algorithms include the Matching Network (Vinyals et al. 2016) and the Prototypical Network (ProtoNet) (Snell, Swersky, and Zemel 2017), which measure the euclidean distance or cosine similarity between query and support embeddings to identify the most probable categories of the queries. Similar algorithms include FEAT and DeepSets (Ye et al. 2020), which incorporate a set-to-set transformative layer (i.e., self-attention (Vaswani et al. 2017) and Deep Sets function (Zaheer et al. 2017)) to the Matching Network, enhancing the expressiveness of image embeddings.

Image-to-image translation is a form of generative technique. Deep generative models involve popular architectures like the Variational Auto-Encoder (VAE) and the Generative Adversarial Network (GAN) (Oussidi and Elhassouny 2018; Wang et al. 2017; Goodfellow et al. 2014), while the Adversarial Auto-Encoder combines VAE and GAN. Famous image-to-image translation models include the CycleGAN (Zhu et al. 2017), which merges two images through a symmetric pair of GAN networks. Other related works such as the StyleGAN (Karras, Laine, and Aila 2019), Coco-Funit/Funit (Saito, Saenko, and Liu 2020; Liu et al. 2019b) and more (Park et al. 2019; Wang et al. 2018a) can also achieve realistic results in generating new images conditioned

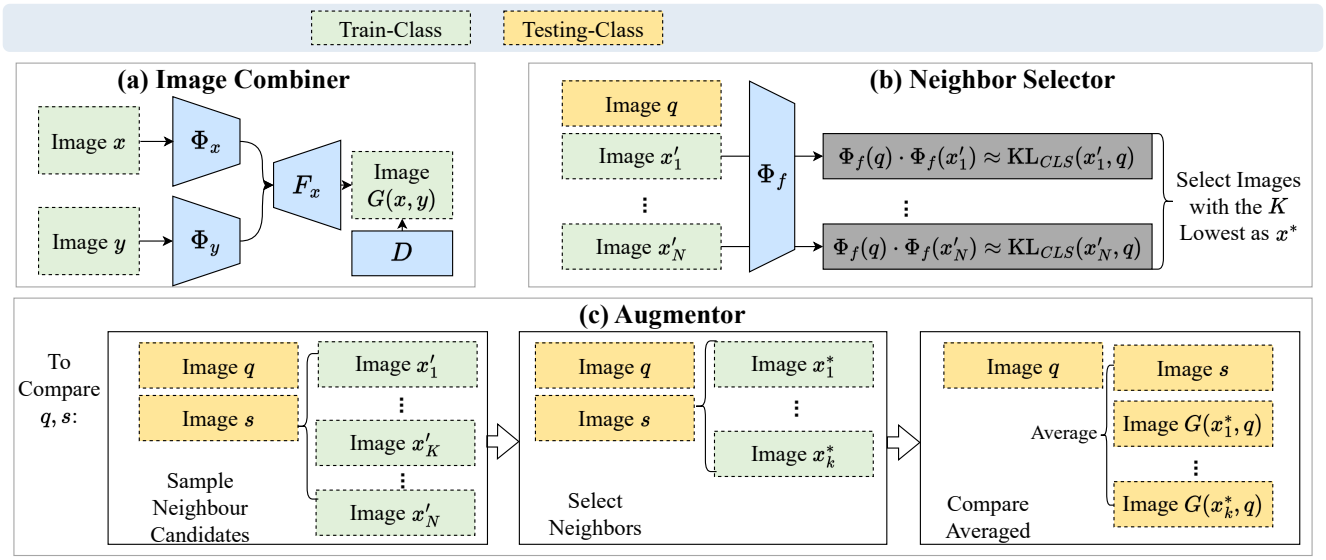

Figure 2: Architecture of FSL-Rectifier.

on different user requirements.

Data augmentation for FSL efforts mainly orient towards training data augmentation. Earlier, a "congealing" method grafts variations from similar training classes to a separate training class (Wang et al. 2020). Similar ideas are also seen in FSL works like Mishra et al. (2018); Verma et al. (2018); Schwartz et al. (2018); Hariharan and Girshick (2017); Wang et al. (2018b); Gao et al. (2018), where the authors augment the training dataset to achieve better generalization among FSL models. Another work similar to ours is Kim et al. (2019), where the authors correct the angles of traffic signs through VAE and logo prototype images. Although most prior works focus on training-phase augmentation, we only focus on testing-phase augmentation for models that are already trained, which is a novel idea to our best knowledge.

3 FSL-Rectifier

3.1 Preliminary: Few-Shot Learning

Consider an FSL classifier h, which learns from train classes C^{train} with complete supervision and is then tested on a set of novel classes C^{test}. The training classes and test classes do not overlap (i.e., $C^{\text{train}} \cap C^{\text{test}} = \varnothing$). Commonly, during both training and testing, FSL classification tasks follow an N-way-K-shot set-up; N represents the number of classes being classified and K denotes the number of labelled samples per class. These $K \times N$ samples, denoted as $S = \{x_1^{(1)}, x_2^{(1)}, \ldots, x_{K-1}^{(N)}, x_K^{(N)}\}$, are the labelled support set (Snell, Swersky, and Zemel 2017; Ye et al. 2020; Sung et al. 2018; Wang et al. 2020; Dhillon et al. 2019). During *training*, the classifier h is presented with the labelled support set S and unlabelled queries q sub-sampled from the N classes. The classifier h has to learn or predict which categories q belong to by referring to S. Here, the N classes are a subset of train classes C^{train}. During *testing*, h has to predict classes of q based on labels of S as well, but both q and S are sampled from the novel test classes C^{test} (Snell, Swersky, and Zemel 2017; Ye et al. 2020; Sung et al. 2018; Wang et al. 2020; Dhillon et al. 2019).

3.2 Method

Overall Pipeline. As illustrated in Figure 2, our method, FSL-Rectifier, consists of three main components: 1) image combiner, 2) neighbour selector, and 3) augmentor. During stage 1, we train a generative image combiner that takes in a training sample and a test sample, producing a test-class sample based on the general shape of the training sample. At stage 2, we train a neighbour selector, which can take in a test sample and a set of candidate training samples. It can return a suitable training sample to be combined with the test sample for generation. The suitable sample is termed *neighbour*. At stage 3, we perform test-time augmentation. For each test sample, we pick its neighbours via the neighbour selector. Then, we produce generated test samples based on neighbours picked and the original test sample, through our image combiner trained at stage 1. For FSL classification with the augmentation setting, we consider the average image embeddings of the original test samples and the generated test samples, instead of just the single original sample.

Image Combiner. Let x denote the image from which we extract the general shape. Let y denote an image from which we extract the class-defining style. The image combiner G consists of a shape encoder Φ_x, a style encoder Φ_y and a decoder F_x. The style encoder captures the style and appearance of an image, and the shape encoder captures the general shape of another image. The image combiner essentially produces images $G(x, y) = F_x(\Phi_x(x), \Phi_y(y))$.

During training, we solve the following minimax optimization problem:

$$\min_D \max_G \mathcal{L}_{\text{GAN}}(D, G) + k_R \mathcal{L}_R(G) + k_{\text{FM}} \mathcal{L}_{\text{FM}}(G). \quad (1)$$

where D is the discriminator in the GAN network, k_R and k_{FM} are hyperparemters, $\mathcal{L}_{\text{GAN}}$ is the GAN-loss, $\mathcal{L}_R$ is the reconstruction loss and $\mathcal{L}_{\text{FM}}$ is the feature matching loss (Salimans et al. 2016). We elaborate on each loss below.

Firstly, the GAN-loss drives the training of both the generator and the discriminator to compete through the following objective function:

$$\mathcal{L}_{\text{GAN}}(D, G) = \mathbb{E}_x[-\log D(x)] + \mathbb{E}_{x,y}[\log(1 - D(G(x, y))]. \tag{2}$$

Here, the discriminator tries to discern between real and images produced by the generator, while encouraging the generation to become the same class as the original test sample class through a classifier h_d coupled with the discriminator encoder Φ_d, where $D = h_d \circ \Phi_d$. h_d is a fully connected layer with output size equal to the number of train classes. At the same time, the generator tries to fool the discriminator.

The reconstruction loss $\mathcal{L}_R$ helps generator G generate outputs that resemble the shape of the target images, which are designed as the input images themselves in the loss function:

$$\mathcal{L}_R(G) = \mathbb{E}_x[\|x - G(x, x)\|_1]. \tag{3}$$

The feature matching loss helps regularize the training, generating new samples that possess the style of the image y:

$$\mathcal{L}_{\text{FM}}(G) = \mathbb{E}_{x,y}\left[\|\Phi_d\left(G(x, y)\right) - \Phi_d(y)\|_1\right]. \tag{4}$$

Here, Φ_d is the feature extractor, which is obtained from removing the last layer (the classifier layer h_d) from the discriminator D.

Note that this image combiner, including the reconstruction loss and feature matching loss, follows prior techniques (Saito, Saenko, and Liu 2020; Liu et al. 2019a; Liu, Breuel, and Kautz 2017; Park et al. 2019; Wang et al. 2018a; Salimans et al. 2016). The decoder F_x includes a few adaptive instance normalization (AdaIN) residual blocks (Huang and Belongie 2017; Liu et al. 2019a). Lastly, during training of the image combiner, we only use the train-split of a dataset, leaving the test-split dataset unseen.

Neighbour Selector. The image combiner can at times produce poor results. To ensure that good-quality generations are used during the testing phase, we design the neighbour selector which, when presented with a pool of candidate neighbour images, can return the better candidates for the generation of new test samples. Intuitively, a naive way to implement this neighbour selector is to generate a set of new test samples and select the better based on a measurement of generation quality. However, it can be computationally expensive to generate the actual images for quality assessment. Therefore, we aim to ensure the quality while skipping the actual generation.

Reusing the trained image combiner, we aim to update the discriminator encoder Φ_d to become Φ_f such that, for a pair of image samples $\{x, y\}$, $\Phi_f(x)^\top \Phi_f(y)$ returns a "generation quality" score, which estimates the quality of generation for $G(x, y)$. To achieve this, during training of Φ_f, we feed combinations of train-class image samples and neighbour candidates $\{x, y\}$ and minimize the following objective function $\mathcal{L}_{KL}$, defined by:

$$\mathbb{E}_{x,y}\left[|\Phi_f(x)^\top \Phi_f(y) - KL(\sigma(h_d(G(x, y))) \parallel \sigma(h_d(y)))|\right]. \tag{5}$$

Here, σ is the softmax function, $h_d(\cdot)$ represents the logits output from the trained and frozen classifier h_d in discriminator D. For example, when there are 64 train classes in a dataset, there are 64 logit scores associated with each class, as returned by the classifier h_d. We measure the KL-divergence score between the class logit distributions of a potential generation and the target-class sample. When the KL-divergence score is low, the generation is desirable, since the generation $G(x, y)$ tends to be the same in class as its target y. We try to train image embeddings that, when multiplied in a pairwise manner, indicate the degree of class difference between the generation and the original sample. Therefore, during actual testing, a candidate neighbour sample x^*, whose feature leading to the lowest divergence score when multiplied with the original test sample feature $\Phi_f(y)$, is selected from a pool of neighbour candidates $\{x'\}$:

$$x^* = \underset{x'}{\operatorname{argmin}}\, \Phi_f(x')^\top \Phi_f(y). \tag{6}$$

As seen in Equation 6, we can eliminate the generation of actual new test samples compared with directly using the KL-divergence in Equation 5 to measure the generation quality. Meanwhile, the same configuration for the image combiner is used. We finetune the neighbour selector based on existing model parameters of D learnt from stage 1, keeping the training for neighbour selector simplistic.

Augmentor. Suppose we generate K new samples for N-way-1-shot classification, and $h \circ \Phi$ is the trained FSL model with encoder Φ and classifier h. The augmentor $\gamma_K(y)$ translates one image sample y to K copies of generated images, before considering their average embeddings:

$$\begin{aligned} &\gamma_K : y \rightarrow \frac{1}{K+1} \sum \\ &(\Phi(G(y, x_1^*)) + \ldots \Phi(G(y, x_K^*)) + \Phi(G(y, y))). \end{aligned} \tag{7}$$

Here, y is the test sample and $\{x_i^*\}$ are neighbours. Note that one can tune embedding weights during the embedding summation. Final FSL predictions can be rendered based on classifier h:

$$\hat{y} = h(\gamma_K(q)|\{\gamma_K(s_1), \ldots \gamma_K(s_n)\}). \tag{8}$$

where q represents the query image and s represents each support set image.

4 Theoretical Guarantee

Here we present how our method with an Support Vector Machine (SVM) (Cortes and Vapnik 1995) classifier tends to achieve better performance with one augmentation, a simplified case, from a theoretical perspective. Through Proposition 1, we highlight the high probability of outlier (defined by data points with high-value feature norm) reduction. Via Theorem

1, we associate our method with a usually tighter generalization bound for trained models consisting of a general-architecture encoder and an SVM classifier. This formulation is meaningful for trained FSL models of many configurations, since we do not assume a fixed model encoder architecture (e.g., image feature distribution); for the classifier, it has been formally established that gradient descent iterations on logistic loss and separable datasets converge to hard margin SVM solutions (Soudry et al. 2024; Rosset, Zhu, and Hastie 2003). Recently, strong connection between attention-based (Vaswani et al. 2017) classifiers and SVM is also established (Tarzanagh et al. 2024).

4.1 Problem Formulation

We assume that we have a SVM classifier, h, trained on pairwise feature differences from the training image embedding differences $\{\mathbf{\Omega} \coloneqq |\Phi(a) - \Phi(b)|; a, b \in \mathcal{D}_{\text{train}}\}$, with labels l defined as:

$$l(\mathbf{\Omega}) = \begin{cases} 1 & \text{if } y_a = y_b, \\ -1 & \text{otherwise.} \end{cases} \tag{9}$$

Note that $\mathbf{\Omega}$ has the same dimension as $\Phi(\cdot)$ under the operation of element-wise absolute difference. Consider an N-way-1-shot classification task consisting of a query q and support set samples $\{s_1, s_2, \ldots, s_N\}$ during testing. Given pairwise feature differences $\mathbf{\Omega} = \{|\Phi(q) - \Phi(s_1)|, |\Phi(q) - \Phi(s_2)|, \ldots, |\Phi(q) - \Phi(s_N)|\}$, after SVM training, few-shot prediction is rendered as $\text{argmax}_i\, h(\mathbf{\Omega}_i)$ for $1 \leq i \leq N$. With this setup, we perform FSL classification via a one-vs-all-classifier (Mohri, Rostamizadeh, and Talwalkar 2018), where each comparison is reduced to a binary classification. In essence, each pairwise feature difference signifies how much two data points differ from each other (Bai et al. 2024).

4.2 Generalization Bound Analysis

We assume that the test set and augmentation set data are of the same *i.i.d.* data distribution. The test-split feature difference data distribution is defined as $S_1 \coloneqq \{\mathbf{\Omega} : |\Phi(a) - \Phi(b)|;\ \ a, b \in \mathcal{D}_{\text{test}}\}$. Similarly, S_2 is defined for the augmentation test pairs (i.e., $S_2 \coloneqq \{\mathbf{\Omega}' : |\Phi(a') - \Phi(b')|;\ \ a', b' \in \mathcal{D}_{\text{aug}}\}$). The maximum norm r in S_1 is associated with test images a and b (i.e., $r = \max_{S_1} \|\mathbf{\Omega}\| = \||\Phi(a) - \Phi(b)|\|$, $|\Phi(a) - \Phi(b)| \in S_1$).

Proposition 1. *When the cardinality of S_1 is v, we have* $\mathbb{P}_{S_1,S_2}(\max_{S_1} \|\mathbf{\Omega}\| > \|\mathbf{\Omega}'\|) = \mathbb{P}_{S_1,S_2}(r > \|\mathbf{\Omega}'\|) = 1 - \frac{1}{2^v}$, $\forall \mathbf{\Omega}' \in S_2$.

See proof in Appendix Section §B. This proposition states that the maximum norm r in S_1 is larger than any norm in S_2 (i.e., $\|\mathbf{\Omega}'\|$) with high probability. Furthermore, the norm of our rectified combination defined as $\|\hat{\mathbf{\Omega}}\|$ can be bounded as follows:

$$\begin{aligned} \|\hat{\mathbf{\Omega}}\| &= \||(0.5\Phi(a) + 0.5\Phi(a')) - (0.5\Phi(b) + 0.5\Phi(b'))|\| \\ &= \||0.5(\Phi(a) - \Phi(b)) + 0.5(\Phi(a') - \Phi(b'))|\| \\ &\leq \|0.5|\Phi(a) - \Phi(b)| + 0.5|\Phi(a') - \Phi(b')|\| \\ &= 0.5r + 0.5\|\mathbf{\Omega}'\|. \end{aligned} \tag{10}$$

Therefore, $\|\hat{\mathbf{\Omega}}\|$ can only be larger than r when $\|\mathbf{\Omega}'\| > r$. Since $\|\mathbf{\Omega}'\|$ tends to be smaller than r according to Proposition 1, the proposition implies that our method tends to combine the original input in S_1 with augmentation in S_2 associated with norm smaller than r, and as a result, the new norm tends to be smaller than r with high probability. Through our method, the maximum norm tends to reduce regardless of data distribution, especially when v is large.

Definition 1 (Margin Loss Function)**.** *For any $\rho > 0$, the ρ-margin loss is the function $L_\rho : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ defined for all $y, y' \in \mathbb{R}$ by $L_\rho(y, y') = \tau_\rho(yy')$ with*

$$\tau_\rho(x) = \begin{cases} 1 & \text{if } x \leq 0, \\ 1 - \frac{x}{\rho} & \text{if } 0 \leq x \leq \rho, \\ 0 & \text{if } \rho \leq x. \end{cases} \tag{11}$$

The parameter $\rho > 0$ can be understood as the required confidence margin associated with hypothesis h. Suppose we have a margin loss characterized by ρ, we have the following generalization bound:

Theorem 1. *Let $h = \mathbf{\Omega} \mapsto \mathbf{w} \cdot \mathbf{\Omega} : \|\mathbf{w}\| \leq \Lambda$ and $S \subseteq S_1$. Fix $\rho > 0$, then, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S of size m, the following holds for h :*

$$R(h) \leq \widehat{R}_{S,\rho}(h) + \underbrace{2\sqrt{\frac{r^2\Lambda^2/\rho^2}{m}}}_{\textit{Decreases with } r} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

See proof in Appendix Section §C. Note that $R(h) \leq \widehat{R}_{S,\rho}(h)$ is the expected risk with respect to the margin loss characterized by ρ, which equals $\mathbb{E}_{S \sim \mathcal{D}_{\text{test}}}[\widehat{R}_{S,\rho}(h)]$. The empirical risk $\widehat{R}_{S,\rho}(h)$ equals $\frac{1}{m}\sum_{i=1}^{m} \tau_\rho(l_i h(\mathbf{\Omega}_i))$, and $l_i \in \{-1, 1\}$ refers to the label associated with $\mathbf{\Omega}_i$. During testing, with augmentation, r tends to decrease with outliers rectified, as implied by Proposition 1. Therefore, the second term $2\sqrt{\frac{r^2\Lambda^2/\rho^2}{m}}$ tends to decrease, resulting in a lower generalization error.

5 Implementation and Experiments

5.1 Datasets

In this work, we use the Animals dataset, which is sampled from the ImageNet dataset (Russakovsky et al. 2015). The Animals dataset contains carnivorous animal facial images (Liu et al. 2019a). The train-test split follows prior works (Liu et al. 2019a). For further analysis, we also consider a mammal animal dataset, or the Mammals dataset (Asaniczka 2023) consisting of 45 testing classes.

5.2 Models and Training

To obtain trained FSL models used in this work, we use the ProtoNet (Snell, Swersky, and Zemel 2017), the FEAT, and the DeepSet (Ye et al. 2020) with either an euclidean-distance or cosine-similarity classifier for experiments. We consider a 4-layer-CNN encoder (Conv4), which we adopt from Ye et al. (2020). We pretrain the encoder before training

	Euclidean Distance Classifier on Animals (%)			Cosine Similarity Classifier on Animals (%)		
	ProtoNet	FEAT	DeepSet	ProtoNet	FEAT	DeepSet
No Augmentation	54.64 ± 0.60	46.86 ± 0.59	56.48 ± 0.59	54.56 ± 0.60	46.80 ± 0.59	55.26 ± 0.59
Oracle	80.20 ± 0.56	69.96 ± 0.57	73.92 ± 0.57	79.74 ± 0.56	69.92 ± 0.57	74.38 ± 0.57
Rotate	53.68 ± 0.60	50.60 ± 0.59	55.24 ± 0.59	54.68 ± 0.60	50.36 ± 0.59	53.79 ± 0.59
ColorJitter	52.25 ± 0.60	46.71 ± 0.58	54.24 ± 0.58	52.24 ± 0.60	46.64 ± 0.58	54.01 ± 0.58
Affine	52.25 ± 0.60	46.07 ± 0.60	52.12 ± 0.60	49.68 ± 0.60	46.68 ± 0.60	49.41 ± 0.60
Mix-Up	54.44 ± 0.63	47.16 ± 0.59	53.32 ± 0.59	54.60 ± 0.63	47.76 ± 0.59	53.19 ± 0.59
FSL-Rectifier (Ours)	57.90 ± 0.60 (*3.26*↑)	53.28 ± 0.56 (*6.42*↑)	60.18 ± 0.58 (*3.70*↑)	58.12 ± 0.60 (*3.56*↑)	52.80 ± 0.56 (*6.00*↑)	58.74 ± 0.58 (*3.48*↑)

Table 1: 5-way-1-shot accuracy on Animals datasets, under different trained FSL models.

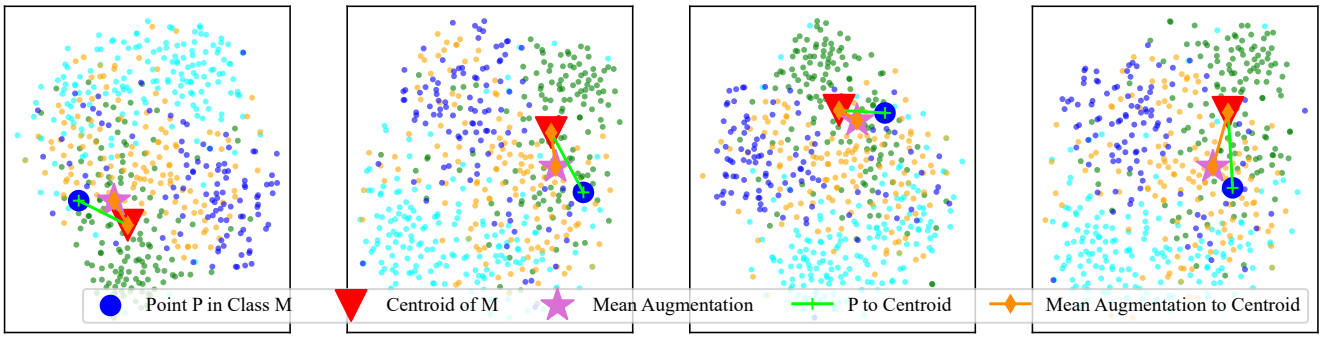

Figure 3: TSNE plot indicating that average augmentation of a random point P (★) stay closer to the class centroid (▼), compared to the random point P (●) on its own. Best viewed in colors.

it further on different FSL algorithms. We directly employ the pretrained image translator models from Liu et al. (2019a); Saito, Saenko, and Liu (2020). When training the neighbour selector, we clone a copy of the trained image combiner. The learning rate is set to 1×10^{-4}, and the maximum number of training iterations is 10,000. When testing our augmentation against the baseline, the neighbour selector considers 20 candidates for each neighbour selection.

5.3 Evaluation Protocol and Results

We compare our method with other main test-time augmentation techniques (Kimura 2024). Our baselines include: **1) No Augmentation**: based on original support set and queries without any augmentation; **2) Oracle**: based on the average of four real test samples for respective support set and queries, which is similar to the 5-way-4-shot evaluation benchmark; **3-5) Rotate/Affine/Color-Jitter**: based on various augmentations adopted from the Pytorch transform functions (Paszke et al. 2019). Rotate is RandomRotation with degree options from $\{0, 180\}$; Affine is RandomAffine with degree options from $\{30, 70\}$, translate options from $\{0.1, 0.3\}$ and scale options from $\{0.5, 0.75\}$; Color-Jitter is ColorJitter with brightness, contrast and saturation as 0.2, hue as 0.1. Besides, we perform **6) Mix-Up** defined as the pixel-level average between the original test sample and its neighbour. For our method, the combined weight of all augmentations and that of the original samples are both set to 0.5. To avoid repeated inference, we save augmentation for each original image, where each generation based on 20 neighbour candidates takes around 0.47 seconds to be generated via one NVIDIA RTX A5000. We pass 25,000 5-way-1-shot queries to test a trained FSL model. 5-way-1-shot accuracy and 95% confidence intervals are reported. Table 1 consists of experimental results of our proposed method over the Animals dataset. Based on the results, our method can improve the original FSL model performance by around 4% on average, which is more effective compared to other common test-time augmentation techniques. Note that the FEAT is not trained till full convergence under the given set of hyper-parameters, and Table 5 includes additional results over the Mammals dataset without the neighbour selector.

6 Further Discussions and Ablation Studies

Near to Centroids and Reduced Norm? Since centroids are defined by the averaged representation of a class, we speculate that our test sample features become closer to class centroids when we consider the average of more same-class copies instead of just one copy. We randomly sample images from 4 classes and visualize the TSNE (van der Maaten and Hinton 2008) plot of the sampled images. Figure 3 summarises our visualization. The blue point P is a randomly sampled point in class M, and we always observe that its rectified version (i.e., averaged with augmentations) comes closer to the centroid. Intuitively, when the points to be classified become nearer to the centroids, the performance degradation caused by the outlier effect is reduced.

Moreover, as discussed in Section 4, the maximum norm of pairwise feature difference tends to reduce through our method. We randomly sample 4,000 image feature pairs and measure the maximum pairwise norm (i.e., $\mathbf{\Omega}$), calculated before and after 3 copies of augmentation. We repeat the sampling and calculation 15 times. Although the augmented

Figure 4: Illustration of the effect of neighbour selector. The first row are the original test samples, the second row are neighbours picked by either neighbour selector or inverse neighbour selector. The third row are new test samples generated by the image combiner based on neighbours picked by different neighbour selectors.

copies are of a different data distribution, where a higher maximum norm of difference is expected, we still observe a reduced maximum norm for our method over all trials. As indicated in Table 2, the average maximum norm reduces from 21.37 to 19.08. The reduction is consistent with our theoretical analyses.

	Mean	STD
Norm Before	21.37	0.81
Norm After	19.08	0.57

Table 2: Maximum norm of randomly sampled absolute-valued feature difference, before and after rectification.

Without Neighbour Selector? To study the importance of our neighbour selector, we first visualize the neighbours and generations rendered by an *inverse neighbour selector* and the original neighbour selector. To recap, the neighbour selector sorts a set of candidate neighbours from good to bad. The inverse neighbour selector is the same but with reversed sorting. The contrast is presented in Figure 4, demonstrating better generation returned by the neighbour selector. Meanwhile, as indicated in Table 3, the inverse neighbour selector downgrades the ProtoNet performance on Animals dataset from 54.64% to around 51%. The random neighbour selector can only improve the original FSL model accuracy score to 56.70%. In this study, we set the number of neighbour candidates to consider for each generation to 20.

How Much Augmentation? To study how many samples to augment, we augment each input query with less than or equal to 3 additional copies. Table 4 summarises our results on the trained ProtoNet model of a euclidean-distance classifier. We observe that the best result (i.e., 57.90%) is achieved by 3 augmentation copies for both the support set samples

	5-Way Accuracy (%)
No Augmentation	54.64
Neighbour Selector	57.90
Inverse Neighbour Selector	51.08
Random Neighbour Selector	56.70

Table 3: Effect of neighbour selector and its ablative variants.

Augmentation Copies	(Support) 0	1	2	3
(Query) 0	**54.64**	56.60	57.31	57.61
1	53.46	54.24	55.95	55.63
2	54.27	56.01	56.47	57.10
3	54.30	56.30	57.18	**57.90**

Table 4: How sizes of augmentation affect euclidean-distance-based ProtoNet accuracy.

and the queries. Meanwhile, it is important to augment the queries when we augment the support samples, otherwise the support augmentation may backfire (e.g., Degradation from 54.64% to 53.46% when there is 1 query augmentation but no support augmentation.). This phenomenon is expected, since the augmentation copies have different data distributions compared to the original images (e.g., different image resolutions). Thus, the augmentation sizes for support-query pairs should match to achieve consistency and optimality.

7 Conclusions

On the whole, we propose a method to improve the performance of the trained FSL model through test-time augmentation, during which the image combiner converts suitable training samples, picked by the neighbour selector, to test classes. The averaged representation becomes more typical especially when compared to outliers, thus improving a

trained FSL model without costing extra dataset or training. This approach explores the alternative possibility that differs from previous training augmentation techniques, and our experiments and theoretical analyses demonstrate its feasibility and effectiveness.

A Additional Results on Mammals

Table 5 summarises additional results over the Mammals dataset without the neighbour selector.

	Cosine Similarity Classifier on Mammals (%)	
	ProtoNet	DeepSet
No Augmentation	43.07 ± 0.55	44.47 ± 0.55
Oracle	76.02 ± 0.50	76.70 ± 0.50
Rotate	42.04 ± 0.55	43.04 ± 0.54
ColorJitter	42.03 ± 0.54	42.36 ± 0.54
Affine	42.39 ± 0.53	42.66 ± 0.53
Mix-Up	41.76 ± 0.56	41.32 ± 0.52
FSL-Rectifier (Ours)	44.07 ± 0.55	46.20 ± 0.54

Table 5: 5-way-1-shot accuracy on Mammals datasets, under different trained FSL models.

B Proof of Proposition 1.

Given that S_1 and S_2 are of the same *i.i.d.* distribution, via the law of symmetry, we have $\mathbb{P}_{S_1,S_2}(\|\mathbf{\Omega}\| \leq \|\mathbf{\Omega}'\|) = \frac{1}{2}$, $\forall\ \mathbf{\Omega}$ and $\mathbf{\Omega}'$ in S_1 and S_2 respectively. Consider a set S_1 consisting of v samples. We have:

$$\begin{aligned}
&\mathbb{P}_{S_1,S_2}(\max_S \|\mathbf{\Omega}\| > \|\mathbf{\Omega}'\|) \\
&\quad = 1 - \mathbb{P}_{S_1,S_2}(\max_S \|\mathbf{\Omega}\| \leq \|\mathbf{\Omega}'\|) \\
&\quad = 1 - (\mathbb{P}_{S_1,S_2}(\|\mathbf{\Omega}_1\| \leq \|\mathbf{\Omega}'\|) \times \\
&\qquad \mathbb{P}_{S_1,S_2}(\|\mathbf{\Omega}_2\| \leq \|\mathbf{\Omega}'\|) \ldots \times \\
&\qquad \mathbb{P}_{S_1,S_2}(\|\mathbf{\Omega}_v\| \leq \|\mathbf{\Omega}'\|)) \\
&\quad = 1 - (\mathbb{P}_{S_1,S_2}(\|\mathbf{\Omega}\| \leq \|\mathbf{\Omega}'\|))^v \\
&\quad = 1 - \frac{1}{2^v}.
\end{aligned} \tag{12}$$

□

C Proof of Theorem 1.

Let $\mathcal{H}$ be our hypothesis set. Consider the family of functions taking values in $[0, 1]$:

$$\tilde{\mathcal{H}} = \{\tau_\rho \circ lh : h \in \mathcal{H}\}. \tag{13}$$

By the Rademacher Complexity Generalization Bound in Mohri, Rostamizadeh, and Talwalkar (2018), with probability at least $1 - \delta$, for all $g \in \tilde{\mathcal{H}}$,

$$\mathbb{E}[g(\mathbf{\Omega})] \leq \frac{1}{m}\sum_{i=1}^{m} g\left(\mathbf{\Omega}_i\right) + 2\Re_m(\tilde{\mathcal{H}}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}. \tag{14}$$

Note that $g(\mathbf{\Omega}) = \tau_\rho \circ lh(\mathbf{\Omega})$, $\Re_m(\tilde{\mathcal{H}}) = \frac{1}{m}\mathbb{E}_{\sigma,S}[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i\tau_\rho(l_i h\left(\mathbf{\Omega}_i\right))]$, m is the cardinality of sample set S drawn from S_1 (i.e., absolute feature differences associated with original test samples) and σ takes a random value from $\{-1, 1\}$ with uniform probability.

$\Re_m(\tilde{\mathcal{H}})$ can be rewritten as:

$$\begin{aligned}
\Re_m(\tilde{\mathcal{H}}) &= \frac{1}{m}\mathbb{E}_{\sigma,S}\left[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i\tau_\rho(l_i h\left(\mathbf{\Omega}_i\right))\right] \\
&= \frac{1}{m}\mathbb{E}_{\sigma,S}\left[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i\tau_\rho(h\left(\mathbf{\Omega}_i\right))\right] = \Re_m(\tau_\rho\circ\mathcal{H}).
\end{aligned} \tag{15}$$

Let $\widehat{R}_{S,\rho}(h)$ equal $\frac{1}{m}\sum_{i=1}^{m} g\left(\mathbf{\Omega}_i\right)$. For all $h \in \mathcal{H}$:

$$\mathbb{E}\left[\tau_\rho(lh(\mathbf{\Omega}))\right] \leq \widehat{R}_{S,\rho}(h) + 2\Re_m\left(\tau_\rho\circ\mathcal{H}\right) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}. \tag{16}$$

Since $1_{u\leq 0} \leq \tau_\rho(u)$ for all $u \in \mathbb{R}$, we have $R(h) = \mathbb{E}\left[1_{lh(\mathbf{\Omega})\leq 0}\right] \leq \mathbb{E}\left[\tau_\rho(lh(\mathbf{\Omega}))\right]$, thus:

$$R(h) \leq \widehat{R}_{S,\rho}(h) + 2\Re_m\left(\tau_\rho\circ\mathcal{H}\right) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}. \tag{17}$$

Since τ_ρ is $1/\rho$-Lipschitz, by the Talagrand's Lemma in Mohri, Rostamizadeh, and Talwalkar (2018), we have $\Re_m\left(\tau_\rho\circ\mathcal{H}\right) \leq \frac{1}{\rho}\Re_m(\mathcal{H})$. Therefore:

$$R(h) \leq \widehat{R}_{S,\rho}(h) + \frac{2}{\rho}\Re_m(\mathcal{H}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}. \tag{18}$$

Note that $S \subseteq \{\mathbf{\Omega} : \|\mathbf{\Omega}\| \leq r\}$ is a sample set of size m drawn from S_1 and $\mathcal{H} = \{\mathbf{\Omega} \mapsto \mathbf{w}\cdot\mathbf{\Omega} : \|\mathbf{w}\| \leq \Lambda\}$. Then, the empirical Rademacher complexity of $\mathcal{H}$ can be bounded as:

$$\begin{aligned}
\Re_m(\mathcal{H}) &= \frac{1}{m}\mathbb{E}_{\sigma,S}\left[\sup_{\|\mathbf{w}\|\leq\Lambda}\sum_{i=1}^{m}\sigma_i\mathbf{w}\cdot\mathbf{\Omega}_i\right] \\
&= \frac{1}{m}\mathbb{E}_{\sigma,S}\left[\sup_{\|\mathbf{w}\|\leq\Lambda}\mathbf{w}\cdot\sum_{i=1}^{m}\sigma_i\mathbf{\Omega}_i\right] \\
&\leq \frac{\Lambda}{m}\mathbb{E}_{\sigma,S}\left[\left\|\sum_{i=1}^{m}\sigma_i\mathbf{\Omega}_i\right\|\right] \\
&\leq \frac{\Lambda}{m}\left[\mathbb{E}_{\sigma,S}\left[\left\|\sum_{i=1}^{m}\sigma_i\mathbf{\Omega}_i\right\|^2\right]\right]^{\frac{1}{2}} \\
&= \frac{\Lambda}{m}\left[\mathbb{E}_{\sigma,S}\left[\sum_{i,j=1}^{m}\sigma_i\sigma_j\left(\mathbf{\Omega}_i\cdot\mathbf{\Omega}_j\right)\right]\right]^{\frac{1}{2}} \\
&\leq \frac{\Lambda}{m}\left[\mathbb{E}_{S}\left[\sum_{i=1}^{m}\|\mathbf{\Omega}_i\|^2\right]\right]^{\frac{1}{2}} \leq \frac{\Lambda\sqrt{mr^2}}{m} = \sqrt{\frac{r^2\Lambda^2}{m}}.
\end{aligned} \tag{19}$$

The first inequality uses the Cauchy-Schwarz inequality and the bound on $\|\mathbf{w}\|$, the second inequality uses the Jensen's inequality, the third inequality uses $\mathbb{E}\left[\sigma_i\sigma_j\right] = \mathbb{E}\left[\sigma_i\right]\mathbb{E}\left[\sigma_j\right] = 0$ for $i \neq j$, and the last inequality uses $\|\mathbf{\Omega}_i\| \leq r$. □

Acknowledgements

This research is supported in part by the Ministry of Education, Singapore, under its MOE AcRF TIER 3 Grant (MOE-MOET32022-0001). The authors would like to thank Jiangwei Chen, Xiangming Gu, Dexter Neo, Miao Xiong, Junfeng Hu, Xiangyu Peng and Haonan Wang for their feedback.

References

Asaniczka. 2023. Mammals Image Classification Dataset (45 Animals).

Bai, Y.; Cai, B. Y.; Tan, Y. K.; Zheng, Z.; Chen, S.; and Chen, T. 2024. FSL-QuickBoost: Minimal-Cost Ensemble for Few-Shot Learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 8326–8335.

Bertinetto, L.; Henriques, J. F.; Torr, P.; and Vedaldi, A. 2019. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*.

Ciubotaru, A.-N.; Devos, A.; Bozorgtabar, B.; Thiran, J.-P.; and Gabrani, M. 2019. Revisiting Few-Shot Learning for Facial Expression Recognition. arXiv:1912.02751.

Cortes, C.; and Vapnik, V. 1995. Support-vector networks. *Machine learning*, 20: 273–297.

Dhillon, G. S.; Chaudhari, P.; Ravichandran, A.; and Soatto, S. 2019. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*.

Fe-Fei, L.; et al. 2003. A Bayesian approach to unsupervised one-shot learning of object categories. In *proceedings ninth IEEE international conference on computer vision*, 1134–1141. IEEE.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Gao, H.; Shou, Z.; Zareian, A.; Zhang, H.; and Chang, S.-F. 2018. Low-shot learning via covariance-preserving adversarial augmentation networks. *Advances in Neural Information Processing Systems*, 31.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Networks. arXiv:1406.2661.

Hariharan, B.; and Girshick, R. 2017. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE international conference on computer vision*, 3018–3027.

Huang, X.; and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, 1501–1510.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.

Kim, J.; Oh, T.-H.; Lee, S.; Pan, F.; and Kweon, I. S. 2019. Variational prototyping-encoder: One-shot learning with prototypical images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9462–9470.

Kimura, M. 2024. Understanding Test-Time Augmentation. arXiv:2402.06892.

Lee, K.; Maji, S.; Ravichandran, A.; and Soatto, S. 2019. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10657–10665.

Liu, M.-Y.; Breuel, T.; and Kautz, J. 2017. Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, 30.

Liu, M.-Y.; Huang, X.; Mallya, A.; Karras, T.; Aila, T.; Lehtinen, J.; and Kautz., J. 2019a. Few-shot Unsueprvised Image-to-Image Translation. In *arxiv*.

Liu, M.-Y.; Huang, X.; Mallya, A.; Karras, T.; Aila, T.; Lehtinen, J.; and Kautz, J. 2019b. Few-Shot Unsupervised Image-to-Image Translation. arXiv:1905.01723.

Marcus, G. 2018. Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.

Mishra, A.; Krishna Reddy, S.; Mittal, A.; and Murthy, H. A. 2018. A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2188–2196.

Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2018. *Foundations of machine learning*. MIT press.

Oussidi, A.; and Elhassouny, A. 2018. Deep generative models: Survey. In *2018 International conference on intelligent systems and computer vision (ISCV)*, 1–8. IEEE.

Park, T.; Liu, M.-Y.; Wang, T.-C.; and Zhu, J.-Y. 2019. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2337–2346.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*.

Rosset, S.; Zhu, J.; and Hastie, T. 2003. Margin Maximizing Loss Functions. In Thrun, S.; Saul, L.; and Schölkopf, B., eds., *Advances in Neural Information Processing Systems*, volume 16. MIT Press.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2018. Meta-learning with latent embedding optimization. *arXiv preprint arXiv:1807.05960*.

Saito, K.; Saenko, K.; and Liu, M.-Y. 2020. Coco-funit: Few-shot unsupervised image translation with a content conditioned style encoder. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, 382–398. Springer.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29.

Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Kumar, A.; Feris, R.; Giryes, R.; and Bronstein, A. 2018. Delta-encoder: an effective sample synthesis method for few-shot object recognition. *Advances in neural information processing systems*, 31.

Shanmugam, D.; Blalock, D.; Balakrishnan, G.; and Guttag, J. 2021. Better aggregation in test-time augmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1214–1223.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Soudry, D.; Hoffer, E.; Nacson, M. S.; Gunasekar, S.; and Srebro, N. 2024. The Implicit Bias of Gradient Descent on Separable Data. arXiv:1710.10345.

Sun, Q.; Liu, Y.; Chua, T.-S.; and Schiele, B. 2019. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 403–412.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Tarzanagh, D. A.; Li, Y.; Thrampoulidis, C.; and Oymak, S. 2024. Transformers as Support Vector Machines. arXiv:2308.16898.

van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9: 2579–2605.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. arXiv:1706.03762.

Verma, V. K.; Arora, G.; Mishra, A.; and Rai, P. 2018. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4281–4289.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching Networks for One Shot Learning.

Wang, K.; Gou, C.; Duan, Y.; Lin, Y.; Zheng, X.; and Wang, F.-Y. 2017. Generative adversarial networks: introduction and outlook. *IEEE/CAA Journal of Automatica Sinica*, 4(4): 588–598.

Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Tao, A.; Kautz, J.; and Catanzaro, B. 2018a. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8798–8807.

Wang, Y.; Yao, Q.; Kwok, J. T.; and Ni, L. M. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3): 1–34.

Wang, Y.-X.; Girshick, R.; Hebert, M.; and Hariharan, B. 2018b. Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7278–7286.

Whang, S. E.; Roh, Y.; Song, H.; and Lee, J.-G. 2023. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal*, 32(4): 791–813.

Ye, H.-J.; Hu, H.; Zhan, D.-C.; and Sha, F. 2020. Few-Shot Learning via Embedding Adaptation with Set-to-Set Functions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8808–8817.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R. R.; and Smola, A. J. 2017. Deep sets. *Advances in neural information processing systems*, 30.

Zhou, Z.-H. 2012. *Ensemble methods: foundations and algorithms*. CRC press.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2223–2232.