Improving Few-Shot Classification via Feature-Aligned AI-Generated Images

Yu-Wen Tung and Mei-Chen Yeh
Department of Computer Science and Information Engineering
National Taiwan Normal University, Taiwan
E-mail: myeh@ntnu.edu.tw Tel: +886-2-77496660

***Abstract*—This study explores the use of AI-generated images to enhance few-shot classification. While traditional data augmentation techniques——such as rotation, resizing, and cropping——create new samples from existing data, they often fail to introduce sufficient diversity. In contrast, we propose leveraging a generative AI model to produce richer and more varied training images. However, directly adding these generated samples to the training set is ineffective due to a distribution gap between the features of real and generated images. To address this problem, we propose a feature alignment strategy that adapts the feature space of generated data to better match that of real data. Experiments on multiple few-shot classification benchmarks demonstrate that our approach improves classification accuracy by both increasing the diversity of training samples and aligning their features with real image features.**

I. Introduction

To address the challenge of insufficient labeled data in resource-constrained scenarios, Few-Shot Learning (FSL) has attracted significant attention in recent years [1]. FSL aims to train models that can perform well even with only a limited number of labeled samples. This capability allows models to classify or predict new, unseen categories effectively, making them more flexible and adaptable to novel tasks.

Given that few-shot learning inherently deals with very limited data, some approaches employ data augmentation to increase the diversity of training samples and enhance classification performance. Traditional data augmentation techniques, such as rotation, scaling, and cropping, introduce variations to the data to enrich the feature space. Another strategy is to generate synthetic data. In particular, Generative Adversarial Networks (GANs) [2] have played a notable role in few-shot learning by producing high-quality, realistic images, thereby expanding the training dataset and improving model generalization. However, GAN-generated images are often constrained by the limited diversity of the small training set, which is an inherent challenge in few-shot learning.

To overcome this limitation, we leverage recent advancements in AI-based image generation to obtain additional training samples. Specifically, we utilize DALL-E [3] to generate semantically rich and diverse images for training FSL models. This approach enables the model to rapidly produce diverse data even from a small set of initial samples, thereby improving both generalization and classification performance while maintaining high visual quality. Furthermore, it reduces the cost and manual effort required to collect and annotate additional data, making it a practical solution for real-world applications.

However, a straightforward use of these AI-generated images proves insufficient: the visual feature distribution of generated images still deviates significantly from that of real images. To address this problem, we propose applying circle loss [4] to reduce this distribution gap. By aligning the visual features of generated samples more closely with those of real data, the support provided by the synthetic images becomes comparable to real samples, ultimately avoiding interference and improving model predictions.

Our main contributions are summarized as follows:

- We investigate the use of AI-generated images for enhancing few-shot learning models.
- We propose fine-tuning the generated data using circle loss to reduce the distribution gap between synthetic and real data.
- We validate our approach on four commonly used FSL datasets: Caltech101 [5], OxfordPets [6], Food101 [7], and Flowers102 [8]. The experimental results demonstrate that our method effectively leverages AI-generated data to improve model classification performance.

II. Related Work

Few-shot learning and meta-learning are closely related areas that address data scarcity by enabling models to generalize from limited labeled samples. While traditional supervised learning focuses on recognizing only seen classes from large datasets, meta-learning aims to "learn to learn" by acquiring transferable knowledge that helps models adapt quickly to new tasks [9]. In practice, meta-learning often serves as an effective framework within FSL to improve adaptability when facing unseen categories. FSL methods can be categorized from data-level augmentation (e.g., translation, rotation, scaling), model-level strategies (e.g., embedding learning, multi-task learning, memory networks), and algorithm-level approaches that guide hypothesis search using prior knowledge [1].

Transfer learning offers another strategy to address limited data and domain shifts by transferring knowledge from a source domain to a target domain [10]. Techniques include instance-based transfer, which reweights source data to match target distributions; feature-based transfer, which aligns feature spaces across domains; parameter-based transfer, which reuses

979-8-3315-7206-8/25/$31.00

trained model parameters; and relation-based transfer, which transfers logical relationships learned in the source domain. These approaches have achieved notable success across applications such as image classification——where pre-trained models on ImageNet [11] are fine-tuned for specific tasks——and natural language processing, where models like BERT transfer general language understanding to downstream applications.

Despite advances, traditional deep models often suffer from limited generalization beyond the domains on which they were trained, partly due to overfitting and dataset-specific biases [12]. CLIP [13] mitigates these issues by jointly learning visual and textual representations in a shared semantic space through contrastive learning. Its architecture, which combines a Vision Transformer (ViT) [14] and a text encoder, enables powerful zero-shot capabilities. By leveraging diverse image-text pairs, CLIP captures richer, more transferable features, and has been successfully applied to tasks like zero-shot image generation [3] and few-shot generation [15].

Generative models have also advanced significantly, starting with Generative Adversarial Networks (GANs) [2], which use adversarial training to produce high-quality images. Although GANs achieve impressive realism, they face challenges including unstable training and limited diversity. Variational Autoencoders (VAEs) [16] offer an alternative probabilistic approach but often generate blurrier outputs. Both methods typically require task-specific training and have limited controllability over the generated content.

Recent transformer-based models such as DALL-E [3] represent a new paradigm by framing image generation as a joint vision-and-language task. By conditioning on textual descriptions, DALL-E produces semantically rich, diverse, and controllable images without additional task-specific fine-tuning. This makes it particularly promising for augmenting data in FSL. However, directly using AI-generated images can introduce distribution gaps between synthetic and real data. We address this challenge in this work.

III. Method

Our method was developed based on Tip-Adapter [17], which improved the few-shot performance of CLIP [13] without the need for computationally expensive backpropagation-based training. Tip-Adapter constructs a key-value cache from the few-shot training set. At inference, it retrieves information from this cache to adjust CLIP's predictions. This non-parametric approach allows Tip-Adapter to maintain CLIP's training-free advantage while significantly improving few-shot classification accuracy. Moreover, its performance can be further enhanced by fine-tuning the initialized adapter for only a few epochs, leading to rapid convergence and competitive results across several benchmarks.

While Tip-Adapter effectively leverages real few-shot training samples, it still depends on a small real validation set to tune hyper-parameters such as cache scale and blending coefficients. This reliance on real images partially breaks the strict few-shot learning (FSL) setting, where only a few labeled samples are assumed to be available. Our method addresses this limitation by replacing the real validation images with AI-generated images produced by a powerful generative model (e.g., DALL-E [3]). This modification ensures that hyper-parameters are estimated in a strictly training-free and data-scarce manner, aligning better with the core principles of FSL.

Beyond this replacement, we introduce an additional component called *G-cache*, which functions similarly to the original cache mechanism but is constructed from AI-generated images instead of real images. The intuition is that these synthetic images, while imperfect, can provide complementary semantic diversity and improve the robustness of predictions.

A. Model Architecture

Our approach builds on the original Tip-Adapter architecture and consists of the following steps:

1) **Feature extraction:** We first use a pre-trained vision-language model (e.g., CLIP) to extract visual features for both the few real labeled samples and the AI-generated images.
2) **Cache construction:** Following Tip-Adapter, we construct a key-value cache from the real few-shot samples. Additionally, we build the *G-cache* from the AI-generated images. Both caches store feature vectors as keys and corresponding class labels as values.
3) **Prediction adjustment:** At inference, we retrieve relevant information from both caches. Specifically, the original cache contributes directly from the few real samples, while the G-cache provides complementary signals derived from the synthetic data. We combine these contributions through weighted blending, where the weights are determined by hyper-parameters tuned on AI-generated validation images rather than real ones.
4) **Feature alignment:** To further reduce the distribution gap between generated and real data, we apply circle loss [4] to align the visual features extracted from AI-generated images with those from real samples. This step helps the G-cache support the model's predictions more effectively, improving overall classification accuracy.

Fig. 1 displays the overall architecture of our method. An input image is first processed by a feature extractor, and then passed into R-Cache, G-Cache, and the enhanced CLIP classifier to obtain the predicted label. This method retains the efficiency nature of Tip-Adapter while improving alignment with the few-shot learning setting. Furthermore, by leveraging AI-generated data we enhance classification performance without requiring extra real validation images.

B. Image Generation

To address the issue of limited samples in the support set and to replace the validation set from real data, our study uses the pre-trained DALL-E Mega model to generate a large number of images. To obtain categorical descriptions, we use the template "What is a [CLASS]?" as input to Chat-GPT, which generates descriptive sentences for each category. In addition, we also collect descriptions from various websites and subjectively filter them to representative ones. The DALL-E model generates

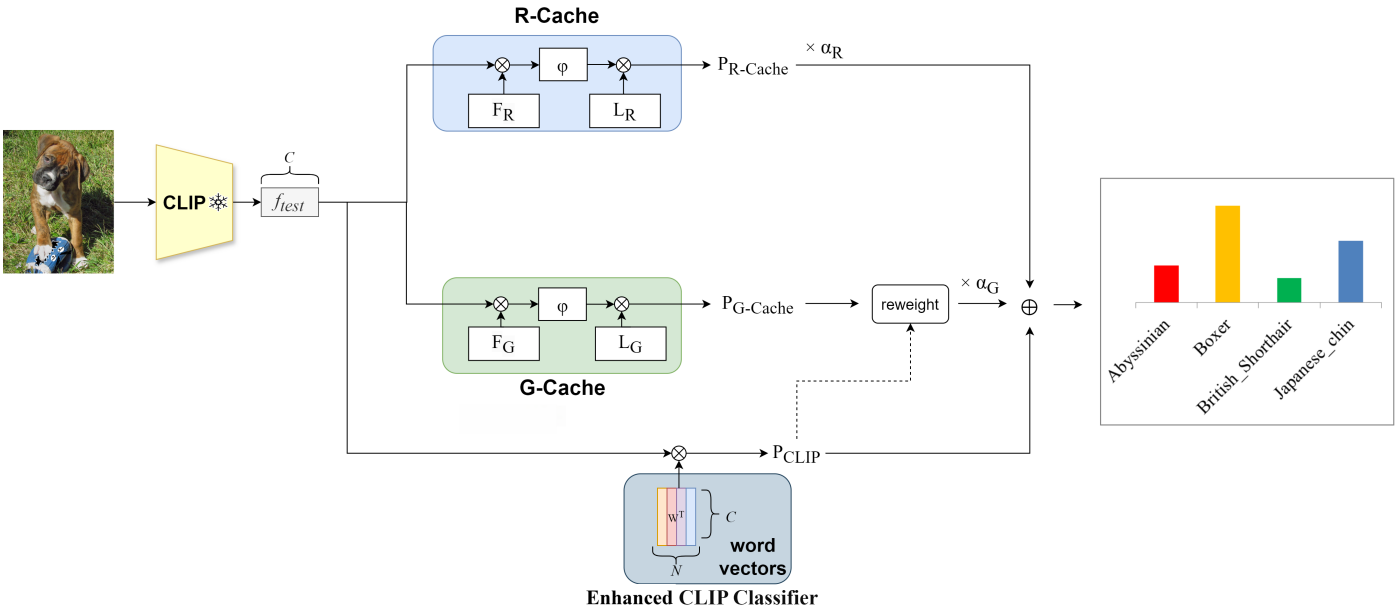


Fig. 1. **Overview of the proposed model architecture.** The model consists of three modules: (1) an enhanced CLIP classifier that uses descriptive sentences to compute text features; (2) the R-Cache, constructed from a few real training samples; and (3) the G-Cache, built from AI-generated images whose features are aligned to real data through a projection module. At inference, predictions from the classifier and both caches are combined to produce the final output.

images based on these descriptions, expanding the original dataset.

In this study, DALL-E generates $2 \times M$ images for each category. One hundred images from half of them are randomly selected to form the G-Cache. The remaining half are used entirely to create a new validation set, replacing the real data in the validation set. This setup ensures that only the R-Cache uses N-way K-shot real samples, with no other real data involved in the model training. This adheres to the standards of few-shot learning, ensuring fairness in the training and validation processes.

C. Cache Models and Enhanced CLIP Classifier

Our method leverages both real and AI-generated data to construct two separate caches: *R-Cache* and *G-Cache*. The R-Cache is built from a small number of real training samples available in the few-shot setting, while the G-Cache is constructed from synthetic images generated by the DALL-E model.

To extract visual features, we utilize the pre-trained CLIP model [13], which was trained on a large-scale dataset of paired images and text descriptions. This model can produce representative and semantically rich feature vectors without the need for additional fine-tuning. Each image, whether real or generated, is passed through the CLIP visual encoder to obtain its feature vector, which serves as a key in the cache model. For the R-Cache, we collect features from $N \times K$ real images (where N is the number of classes and K is the number of few-shot samples per class). For the G-Cache, we similarly use features from $N \times M$ generated images per class. The corresponding labels of these images are converted into one-hot encoded vectors, which act as the values in each cache.

Since the visual feature distributions of real and generated images can differ, we introduce a lightweight projection module within the G-Cache to transform the generated features closer to the distribution of real features. This helps reduce the domain gap and ensures that the synthetic data provide meaningful support during inference. The details of how this projection module is trained will be described in the next subsection.

In zero-shot learning scenarios, the model must recognize categories it has never encountered during training. This typically requires additional semantic information—-such as category names or textual descriptions—-to guide the model in understanding the relationships among classes. Inspired by this idea, our approach incorporates category descriptions as text prompts used during DALL-E image generation. By leveraging CLIP's capability to jointly embed visual and textual data into a shared feature space, the model can effectively retrieve and align relevant semantic information, improving classification without extra supervised training.

Therefore, unlike Tip-Adapter, which directly uses discrete labels as semantic features in the CLIP classifier, our enhanced CLIP classifier leverages richer descriptive sentences for each category. This design aligns better with the text prompts used in the generative process and helps the model better capture semantic nuances between categories, ultimately improving the prediction performance.

D. Model Training

Our method involves two stages of model training. The first stage trains the projection module, which aims to project the visual features of AI-generated images into the feature space of real images, thereby reducing the distribution gap between

them. Since both the support set and test data come from real-world images, the R-Cache naturally provides more reliable predictive information than the G-Cache.

To align the generated features with real features, we adopt circle loss [4]. Circle loss is designed to maximize within-class similarity while minimizing between-class similarity. By doing so, it effectively reduces the feature distance between generated and real data of the same class and increases the separation between features of different classes.

The loss operates by measuring the similarity scores of each sample relative to its class center, applying different penalty intensities depending on these scores. Concretely, the Euclidean distance of each feature vector to its class center is computed and compared against within-class radii and between-class margins. If a feature vector falls within its designated "class circle," the penalty is zero; otherwise, the penalty increases proportionally to its distance. The circle loss is formally defined as:

$$L_{CL} = \log\Big[1 + \big(\sum_{j=1}^{L} \exp\big(\gamma(s_n^j + m)\big)\big)\big(\sum_{i=1}^{K} \exp\big(-\gamma s_p^i\big)\big)\Big], \tag{1}$$

where s_p and s_n denote the within-class and between-class similarity scores, respectively; γ is a scaling factor controlling convergence speed; and m is the margin that adjusts the separation between positive and negative similarities. K and L represent the number of within-class and between-class similarity scores, respectively.

In the second stage, we fine-tune the cache models following the approach in Tip-Adapter-F [17]. Fine-tuning the cache allows the model to further aggregate similar features within each class and enhance inter-class separation, leading to improved classification performance. Specifically, we treat both the R-Cache and the G-Cache (after adjustment by the projection module) as learnable weights in a linear model. This enables the model to utilize prior knowledge captured in both caches to improve the final classifier.

The fine-tuning process optimizes a standard cross-entropy loss:

$$L_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} p_{ij}\log(q_{ij}), \tag{2}$$

where N is the total number of training samples, C is the total number of classes, p_{ij} is the one-hot ground-truth label for sample i and class j, and q_{ij} is the predicted probability that sample i belongs to class j.

Through these two stages, our method effectively reduces the distribution gap between real and generated data and refines the classification boundaries, leading to better few-shot learning performance.

E. Inference

At inference time, our method follows the overall framework of Tip-Adapter [17] by integrating information from both cache models (R-Cache and G-Cache) and the CLIP classifier to produce final predictions. Specifically, given a test image, its visual feature is first extracted using the pre-trained CLIP visual encoder. This feature is then compared against the entries stored in the R-Cache and G-Cache to obtain two sets of similarity-based predictions. Simultaneously, the feature is also passed through the CLIP classifier, which leverages the category text descriptions to produce semantic predictions.

The final prediction is computed as a weighted combination of these three components. Unlike Tip-Adapter, which searches for optimal values of the hyper-parameters using a real validation set, our method determines these hyper-parameters using an AI-generated validation set instead. This design ensures that the entire inference pipeline adheres to the few-shot learning principle of not relying on additional real validation data.

IV. Experiments

A. Datasets

We evaluate our method on four widely used benchmark datasets: Caltech101 [5], OxfordPets [6], Food101 [7], and Flowers102 [8]. These datasets cover both coarse-grained and fine-grained classification tasks, allowing us to comprehensively assess the generalization capability of our approach.

- **Caltech101** contains 101 object categories, covering a diverse range of classes with large inter-class differences, making it a coarse-grained dataset. It includes 4,128 training images and 2,465 test images.
- **OxfordPets** consists of 37 categories of pet breeds, many of which share visually similar features, posing a fine-grained classification challenge. It comprises 2,944 training images and 3,669 test images.
- **Food101** includes 101 types of food, also considered fine-grained due to subtle visual differences between categories. The dataset contains 50,500 training images and 30,300 test images.
- **Flowers102** features 102 flower categories that are visually challenging to distinguish, making it another fine-grained dataset. It includes 4,093 training images and 2,463 test images.

We adopt the same dataset splits and few-shot settings as Tip-Adapter [17] for fair comparison. For each dataset, a few-shot subset is randomly sampled from the training set to construct the R-Cache. We evaluate performance under three settings: 1-shot, 5-shot, and 16-shot. Regardless of the number of real samples, the G-Cache always uses 100 synthetic images per class generated by the DALL-E model.

B. Implementation Details

In our experiments, we use the pre-trained CLIP model with ViT-B/32 as the visual encoder. The parameters of CLIP remain frozen throughout training to retain its pre-trained semantic knowledge.

The projection module that maps generated visual features to the real feature space consists of a single linear layer followed by a ReLU activation function, forming a lightweight architecture. For training this module, the margin m is set to

TABLE I
FEW-SHOT CLASSIFICATION ACCURACY (%) OF OUR METHOD COMPARED WITH TIP-ADAPTER WITHOUT CACHE FINE-TUNING

Dataset	Caltech101			OxfordPets			Food101			Flowers102		
Shot	1	5	16	1	5	16	1	5	16	1	5	16
Tip-Adapter	91.85	**93.79**	93.55	87.41	87.71	88.28	80.49	80.61	81.03	**79.66**	87.94	**91.64**
Ours	**93.02**	93.59	**93.59**	**89.02**	**89.18**	**89.67**	**80.77**	**80.91**	**81.29**	76.70	**88.39**	90.70

TABLE II
FEW-SHOT CLASSIFICATION ACCURACY (%) OF OUR METHOD COMPARED WITH TIP-ADAPTER-F WITH CACHE FINE-TUNING

Dataset	Caltech101			OxfordPets			Food101			Flowers102		
Shot	1	5	16	1	5	16	1	5	16	1	5	16
Tip-Adapter-F	93.10	94.81	95.42	89.15	89.15	90.98	80.96	81.33	82.16	**80.88**	89.77	**94.32**
Ours	**93.63**	**94.89**	**95.46**	**89.97**	**89.40**	**91.25**	**81.02**	**81.46**	**82.20**	80.47	**90.78**	93.95

0.4, and the scaling factor γ is set to 80. We train for 30 epochs using a learning rate of 0.0001 and a batch size of 2.

In the second stage, we fine-tune the cache models following the setup in Tip-Adapter-F [17]. The fine-tuning process uses a batch size of 256, 20 training epochs, and a learning rate of 0.001. All experiments are implemented in PyTorch and conducted on an NVIDIA RTX 4090 GPU.

C. Results

Table I compares the classification performance of our proposed method with Tip-Adapter [17], where neither method involves cache fine-tuning. Overall, our model consistently outperforms Tip-Adapter, demonstrating the effectiveness of incorporating AI-generated data combined with feature alignment. However, on the Flowers102 dataset, our method shows a slightly lower classification accuracy (approximately 3% lower) compared to Tip-Adapter in the 1-shot setting. This can be attributed to the fine-grained nature of Flowers102, which contains many visually similar categories. As a result, it poses a greater challenge for image generation models to produce synthetic samples that capture the subtle intra-class differences required for accurate classification.

Table II presents the classification results after cache fine-tuning for both our method and Tip-Adapter (named Tip-Adapter-F in [17]). Our model achieves higher accuracy on the Caltech101, OxfordPets, and Food101 datasets across different shot settings. However, on Flowers102, it only surpasses Tip-Adapter-F in the 5-shot setting, while showing slightly weaker performance in the 1-shot and 16-shot scenarios.

This observation can be attributed to the same challenge noted earlier: the fine-grained nature of Flowers102 makes it particularly difficult for the image generation process to produce synthetic samples that capture subtle intra-class differences. While cache fine-tuning helps to some extent, it does not fully resolve this issue.

D. Ablation Studies

This section investigates the contribution of each component in our model, examining the effect of incorporating the G-Cache (which uses AI-generated data) and the projection module (which performs feature alignment). To this end, we conduct a series of ablation experiments, evaluating the model's classification accuracy under different module combinations.

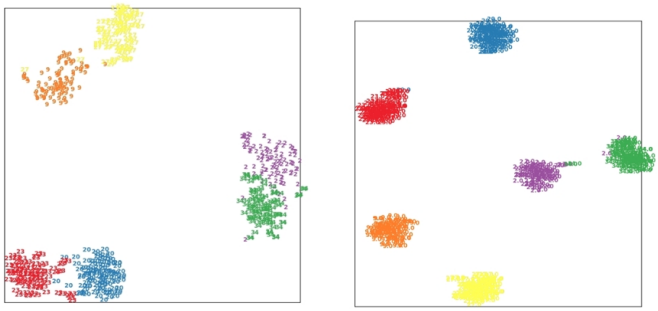
Our proposed model comprises three key modules: the CLIP classifier, the R-Cache, and the G-Cache. We systematically examine different combinations of these modules, and present the results in Table III. The experiments are conducted on the OxfordPets dataset, where the R-Cache is constructed from 16 real samples per class, and the G-Cache consists of 100 AI-generated samples per class. The results clearly demonstrate that each module provides complementary benefits, and that combining all three achieves the highest classification performance.

TABLE III
ABLATION STUDY ON EFFECTS OF THE ENHANCED CLIP, R-CACHE, AND G-CACHE

Setting			Accuracy
CLIP	R-Cache	G-Cache	
✓			87.41
	✓		66.80
		✓	60.18
	✓	✓	68.25
✓	✓		88.28
✓		✓	88.20
✓	✓	✓	88.83

Next, we evaluate the importance of the proposed projection module, which is designed to reduce the feature distribution gap between generated and real data. Table IV presents the classification results with and without applying the projection module. The results clearly demonstrate that introducing the projection module significantly improves classification accuracy across different shot settings, confirming its effectiveness.

Figure 2 provides a visualization of the feature distributions before and after applying the projection module. As shown, before projection, the generated features are more scattered and distant from the real data distribution. After projection, the generated features shift closer to the real data in feature



(a) before projection (b) after projection

Fig. 2. Visualization of generated data before and after transformation by the projection module.

space, resulting in improved within-class clustering and clearer between-class separation.

TABLE IV
ABLATION STUDY ON EFFECTS OF THE PROJECTION MODULE.

Proj.	Caltech101			OxfordPets		
	1-shot	5-shot	16-shot	1-shot	5-shot	16-shot
✗	87.42	90.06	90.47	82.77	83.24	83.16
✓	**91.52**	**93.39**	**93.35**	**87.71**	**87.74**	**88.83**

V. CONCLUSION

This paper presents a few-shot learning method that augments limited real data with AI-generated images and aligns their features using a lightweight projection module trained with circle loss. By combining real and generated data in a cache-based framework, our approach improves generalization under data-scarce conditions without relying on a real validation set. Experimental results on four benchmark datasets demonstrate that our method outperforms Tip-Adapter in most scenarios.

In future work, we plan to explore more advanced generative models and feature adaptation strategies, as well as develop efficient filtering mechanisms to select the most useful generated images, further improving performance, especially on challenging fine-grained datasets.

REFERENCES

[1] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–34, 2020.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[3] A. Ramesh, M. Pavlov, G. Goh, *et al.*, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*, 2021, pp. 8821–8831.

[4] Y. Sun, C. Cheng, Y. Zhang, *et al.*, "Circle loss: A unified perspective of pair similarity optimization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[5] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *IEEE CVPR Workshop*, 2004, pp. 178–178.

[6] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, "Cats and dogs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[7] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in *European Conference on Computer Vision*, 2014.

[8] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008.

[9] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, 2017.

[10] F. Zhuang, Z. Qi, K. Duan, *et al.*, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[12] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization," *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.

[13] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, 2021.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[15] Y. Zhou, C. Li, C. Chen, J. Gao, and J. Xu, "Lafite2: Few-shot text-to-image generation," *arXiv preprint arXiv:2210.14124*, 2022.

[16] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019.

[17] R. Zhang, W. Zhang, R. Fang, *et al.*, "Tip-adapter: Training-free adaption of clip for few-shot classification," in *European Conference on Computer Vision*, 2022, pp. 493–510.