EXPLORING CROSS-MODAL FLOWS FOR FEW-SHOT LEARNING

Ziqi Jiang, Yanghao Wang, and Long Chen*
Department of CSE, The Hong Kong University of Science and Technology
zjiangbl@connect.ust.hk, ywangtg@connect.ust.hk, longchen@ust.hk

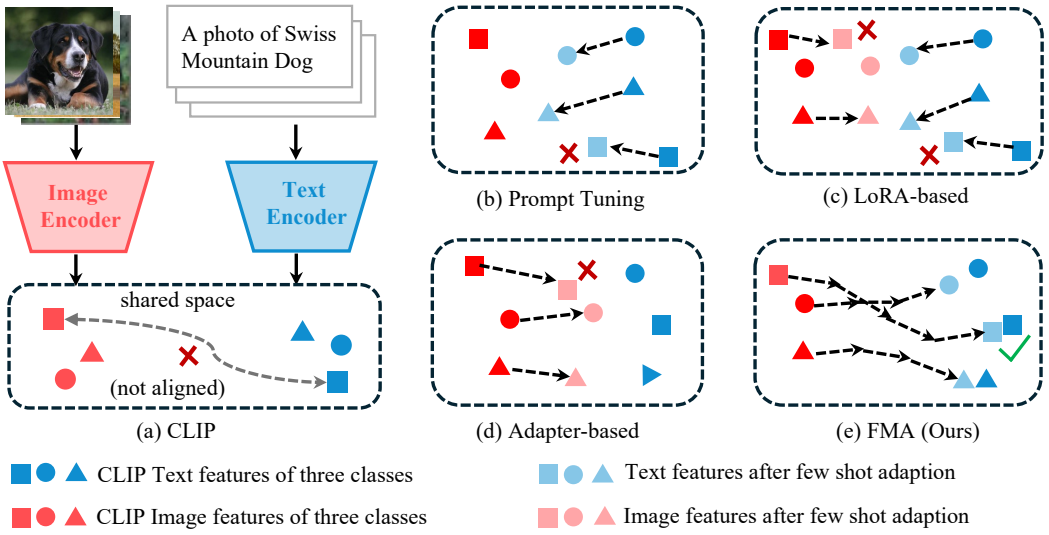

Figure 1: **Comparisons of the cross-modal alignment process of different methods**. **(a)** The overview pipeline of CLIP (Radford et al., 2021) for zero-shot cross-modal alignment and classification. Some image features and their corresponding text features are not well-aligned. **(b-d)** The alignment process of three typical types of state-of-the-art PEFT approaches, which adjust image or text features in *one single step*. The arrow shows the adjustment of corresponding features during the adaptation. For difficult classes, the image features still may be far from the corresponding text features by one step adjustment. **(e)** FMA achieves *multi-step* cross-modal alignment, which succeeds in aligning text-image features for difficult classes.

ABSTRACT

Aligning features from different modalities, is one of the most fundamental challenges for cross-modal tasks. Although pre-trained vision-language models can achieve a general alignment between image and text, they often require parameter-efficient fine-tuning (PEFT) for further adjustment. Today's PEFT methods (*e.g.*, prompt tuning, LoRA-based, or adapter-based) always selectively fine-tune a subset of parameters, which can slightly adjust either visual or textual features, and avoid overfitting. In this paper, we are the first to highlight that all existing PEFT methods perform *one-step adjustment*. It is insufficient for complex (or difficult) datasets, where features of different modalities are highly entangled. To this end, we propose the first model-agnostic *multi-step adjustment* approach by learning a cross-modal velocity field: Flow Matching Alignment (**FMA**). Specifically, to ensure the correspondence between categories during training, we first utilize a fixed coupling strategy. Then, we propose a noise augmentation strategy to alleviate the data scarcity issue. Finally, we design an early-stopping solver, which terminates the transformation process earlier, improving both efficiency and accuracy. Compared with one-step PEFT methods, FMA has the multi-step rectification ability to achieve better alignment. Extensive results demonstrate that FMA can yield significant performance gains across various benchmarks and backbones, particularly on challenging datasets. Codes: https://github.com/HKUST-LongGroup/FMA.

*Corresponding author.

1 INTRODUCTION

How to align information from different modalities (*e.g.*, text and images), is very important in almost all cross-modality tasks. Well-aligned cross-modality features play a crucial role in a variety of domains, such as achieving the impressive reasoning abilities in MLLMs (Hurst et al., 2024; Li et al., 2022; Liu et al., 2023b), and realistic generation qualities in AIGC models (Rombach et al., 2022; Esser et al., 2024). Generally, realizing good cross-modal alignment means finding a "golden" multimodal distribution where corresponding text and image features are as close as possible in the shared common space. To achieve this goal, many pretrained vision-language models (VLMs) have been proposed, such as CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021). Taking CLIP for example, as shown in Figure 1(a), it achieves cross-modal alignment by training image and text encoders jointly with a contrastive objective. By now, VLMs can achieve a general alignment between image and text, and show promising results on zero-shot tasks, like image recognition.

However, due to the internal complexity of different modalities, pre-trained VLMs cannot achieve perfect alignment across all scenarios. Therefore, they typically require further fine-tuning steps to rectify features for better alignment. For instance, in few-shot learning, we usually need to fine-tune VLMs with a few images from base categories. Since fine-tuning the whole VLM model is computationally expensive and Linear Probing (LP, only fine-tunes the final fully-connected layer) is not effective enough, numerous parameter-efficient fine-tuning (PEFT) methods have been proposed. These PEFT methods can be broadly categorized into three groups: 1) **Prompt Tuning** (Li & Liang, 2021): Like CoOp (Zhou et al., 2022b) and CoCoOp (Zhou et al., 2022a), prompt tuning methods try to find better text representations (or features) by replacing handcrafted text prompts with learnable continuous prompts. Compared with original VLMs, they can be interpreted as applying a "movement" on all text features to better align them with the corresponding image features (*cf.*, Figure 1(b)). 2) **Adapter-based** (Houlsby et al., 2019): Such as CLIP-Adapter (Gao et al., 2024), they typically apply a learnable adapter following the VLMs' image encoder. After training, they will rectify the image features towards corresponding text features for better alignment (*cf.*, Figure 1(c)). 3) **LoRA-based** (Hu et al., 2022): These methods (*e.g.*, CLIP-LoRA (Zanella & Ben Ayed, 2024)) add trainable low-rank matrices in both text and image encoders to store the new knowledge while keeping other parameters frozen. As shown in Figure 1(d), they will shift both text and image features to be closer (*i.e.*, towards the golden distribution).

It is widely acknowledged that PEFT typically outperforms linear probing. However, we observed that their advantages compared with LP are not consistent across different datasets. To better illustrate this, we introduce the concept of dataset difficulty: A dataset with lower CLIP zero-shot performance is considered more difficult. This definition is reasonable because lower zero-shot performance means that the cross modal distribution is more complicated, thus more difficult to adjust. In this paper, we found that while PEFT methods work pretty well on relatively simple datasets, they fail to generalize to difficult ones. For example, as shown in Figure 2, the prevailing PEFT method CoOp can obtain remarkable improvements over the LP baseline on relatively simple datasets, like OxfordPets. However, the improvements on more challenging datasets, such as FGVCAircraft, are marginal. To explain this phenomenon, we propose a new perspective for all existing PEFT approaches: these methods try to adjust their general aligned multimodal distribution towards the golden distribution by *one rectification step*. This is because during the inference, the adjustments consist of a single forward pass of the trained model. Since PEFT methods involve only a small number of learnable parameters, they are usually unable to learn very complex transformations. Therefore, for difficult datasets where some multimodal features are highly entangled, these one-step methods are unable to align them, as shown in Figure 1(bcd). This leads to our core motivation: **Can we make *multi-step* rectifications to realize better cross-modal alignment?**

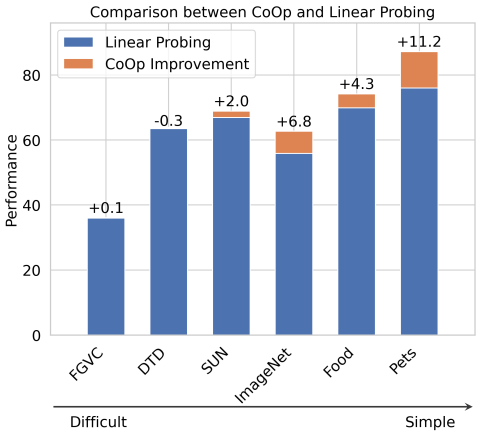

Figure 2: **Performance of CoOp and linear probing.** We experimented with the 16-shot setting and chose CLIP RN50 as the backbone.

To address this, we turn to the Flow Matching (FM) theory (Liu et al., 2022; Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022), which is developed for multi-step transformation between any two distributions. Generally, FM learns a velocity field that transports samples from a source distribution to a target distribution through an iterative process, *i.e.*, while one-step adjustment is not accurate, it is possible for the following steps to rectify prior errors. Therefore, if a velocity field can be trained to transform image features to corresponding text features, we can utilize its multi-step rectification ability for more accurate classification. Specifically, we can formulate all encoded image features as the source distribution. For the target distribution, the simplest method is to encode all prompts with category names as target features. For each velocity training step, an image and text feature will be sampled independently to compose a training pair. This velocity field will transform any given image feature (source distribution) to a corresponding text feature (target distribution), *i.e.*, classification.

Unfortunately, there are two potential issues for this straightforward approach. Firstly, although a well-trained velocity field can achieve transformation between two given distributions, it is not guaranteed to preserve local structure (*e.g.*, category correspondence). For example, as shown in Figure 3(a), it may transform image features from one class to text features of another class, resulting in wrong classification. Therefore, **the first challenge** is to figure out *how to train a velocity field, which can transfer image features from source distribution, not only to the target distribution (near text features), but also close to their right class embeddings.* Secondly, in the inference stage, the goal of flow matching for the generation task is to generate features in the target distribution, which are then decoded to real samples. But for the classification task, what we really need is to transfer image features closer to positive embeddings (*i.e.*, ground-truth class embeddings) than other negative ones. This inconsistency suggests that the previous flow matching inference strategy may not be the best practice for classification scenarios. Thus, **the second challenge** is that *how to design an appropriate inference method for classification.*

To overcome both challenges, we propose a novel framework for few-shot learning: Flow Matching Alignment (**FMA**). Specifically, we have three designs:

- **Coupling Enforcement:** Firstly, for any given image feature, we choose the one that corresponds to its class label to compose a training pair. Theoretically, this fixed coupling strategy can guarantee that the trained velocity field will act like a classifier, moving image features towards the correct direction.
- **Noise Augmentation:** However, compared with randomly pairing, this coupling strategy also reduces the number of training pairings dramatically, making it harder to train the velocity field. To solve this issue, we further add a pre-defined noise to the sampled pair during training. This augmentation makes the training data not constrained in a low-dimensional manifold, making the learning process more stable and robust.
- **Early Stopping Slover:** Besides training, vanilla flow matching inference is not suitable in classification due to the inconsistency (challenge #2). When intermediate features are distinguishable enough for classification, continuing to transfer them to obtain text features is unnecessary and inefficient. Even worse, we observed that the following inference steps are likely to move them to inaccurate text features, as shown in Figure 3(b). This inaccuracy is because even with the previous training strategies, it is still difficult to train a perfect velocity field. To tackle this, we devise an early stopping solver for few-shot learning. Concretely, instead of arriving at the target distribution. We allow the model to output intermediate features for classification. This early stopping strategy can not only reduce the inference time when the features are good enough for classification, but also reduce the risk of transferring to wrong text features.

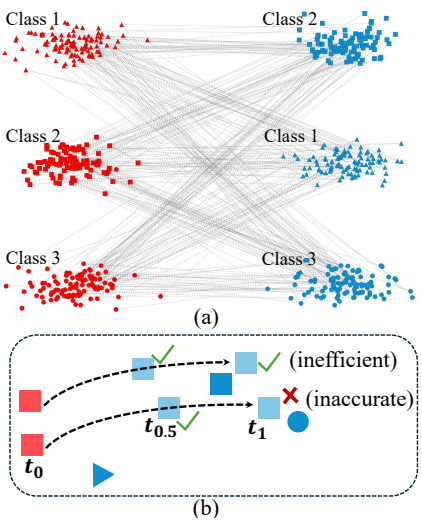

Figure 3: **Explanation of the two challenges.** (a) A flow matching example to transform image features (red) to text features (blue) distribution. Each distribution consists of features of three categories. (b) Two examples of how a trained velocity transforms an image feature into a text feature using the vanilla flow matching inference strategy.

Additionally, FMA can serve as a plug-and-play module for mult-step rectification. It only requires two sets of features in the shared space, and it is agnostic to the specific methods used for image or text feature extraction. Thus, FMA can be easily adapted to different methods, ranging from zero-shot

CLIP to various PEFT methods. To demonstrate the effectiveness and generality of FMA, we have integrated it with various backbones. Extensive results on different backbones and benchmarks have demonstrated consistent performance gains. In summary, our contributions are threefold:

- We propose a new perspective to analyze existing few-shot PEFT approaches: all these methods are one-step methods, and they usually struggle with difficult datasets.
- We are the first to explore the potential to adapt the multi-step recitification ability of flow matching for better few-shot learning performance. Based on this, we propose FMA, a novel plug-and-play multi-step rectification framework.
- Extensive results demonstrate the superiority and robustness of FMA in the few-shot classification.

2 Related Work

Few-Shot Learning in VLMs. Few-Shot Learning (FSL) is a task to learn from a few examples (Wang et al., 2020; Yue et al., 2020). Currently, the most common scenario in FSL is to fine-tune a pre-trained VLM (Radford et al., 2021; Jia et al., 2021) with limited data for classification. Because directly fine-tuning VLMs is computationally expensive, numerous PEFT methods have been proposed. Current techniques for adapting the VLMs can be broadly classified into four paradigms: Prompt Tuning, Adapter-Based and LoRA-Based. The Prompt Tuning approaches (Zhou et al., 2022a;b; Lee et al., 2023; Bulat & Tzimiropoulos, 2023) involves learning textual or visual prompts that are subsequently inserted into the input or middle layers of the pre-trained VLM encoders for a few-shot adjustment. LoRA-Based methods (Zanella & Ben Ayed, 2024; da Costa et al., 2023; Kim et al., 2024; He et al., 2022), instead, insert a small number of trainable parameters (*e.g.*, LoRA) within the encoders themselves. Adapter-based methods (Zhang et al., 2022; Gao et al., 2024), append a trainable layer like MLP to the frozen image or text encoders, and they do not require gradient backpropagation across the encoders, which is more computationally friendly.

Diffusion Model (DM) and Flow Matching (FM). Recently, diffusion model has developed into the most powerful framework in generative models (Ho et al., 2020; Song et al., 2020a). It gradually degrades the data by adding noise, then learn the reverse process. Then Score-SDE (Song et al., 2020b) points out that the learning process of DM is actually solving stochastic differential equations (SDE), which can be further interpreted as probabilistic ordinary differential equations (ODE). Consequently, FM (Liu et al., 2022; Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022) extends this by learning a velocity field to build transformations between any two distributions. Different from previous one-step generative models, such as GAN (Goodfellow et al., 2014) and VAE (Kingma & Welling, 2013), FM generates data through multi-step rectification. So far, FM has gained great success in a variety of domains, such as text-to-image generation (Esser et al., 2024; Geng et al., 2025; Albergo et al., 2023), image editing (Kim et al., 2025; Kulikov et al., 2024; Rout et al., 2024), video generation (Jin et al., 2024; Polyak et al., 2024), and so on. However, although FM can achieve transformation between two arbitrary distributions, most approaches tend to set one as the prior noise distribution. Recently, several works (He et al., 2025; Liu et al., 2025) try to transform from text distribution to image distribution directly for generation. In this paper, we explore the potential of whether FM is suitable for supervised tasks like few-shot classification.

3 Method: Flow Matching Alignment for Few-Shot Learning

Formulation. We consider the N-way K-shot classification problem: Given a training dataset consisting of N classes, each class with K examples, the goal is to train a classification model that can accurately predict the class of any test image from these N classes.

General Framework. The overall pipeline of our proposed FMA is shown in Figure 4. Specifically, FMA consists of three steps: 1) Encoding all text and images into features in the shared common space. 2) Training a velocity field to transform image features to text features for better alignment. 3) Using the trained velocity field to transform the feature of the given test image for classification.

3.1 Feature Extraction

Given the training dataset $\mathcal{D} = \{I^i, C^i\}_{i=1}^{N \cdot K}$, where I^i, C^i represent an image and its corresponding class label, the simplest way to encode them into the shared space is using a pre-trained VLM F_ϕ, like

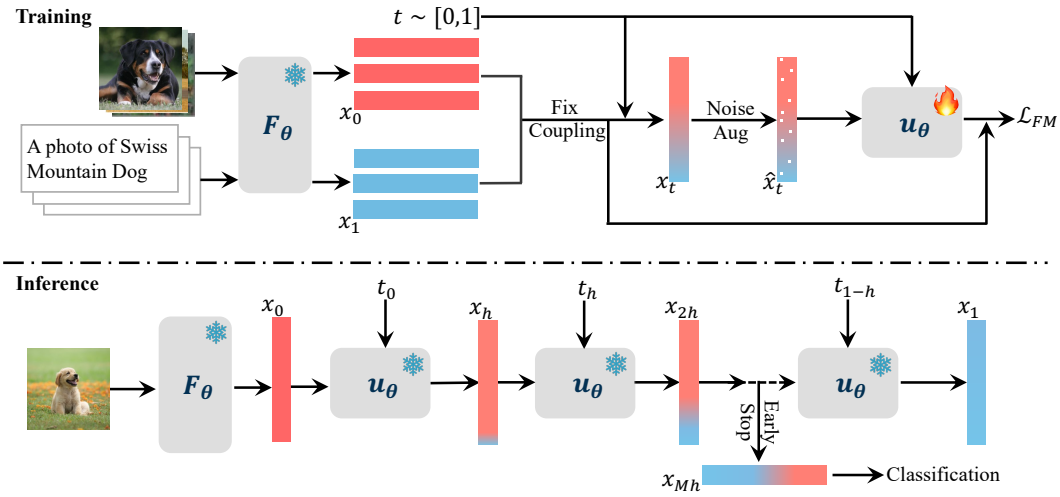

Figure 4: **Overview of Flow Matching Alignment (FMA)**. The main idea of the training stage is to learn a velocity field, which can transform image features to corresponding text features. Two designs are proposed: coupling enhancement and noise augmentation. During the inference, FMA applies an early-stopping solver that can output intermediate features for classification.

CLIP. Specifically, CLIP contains an image encoder E_I and a text encoder E_T. The image (source) distribution is obtained by encoding all images into a feature representation by: $x_0 = E_I(I) \in \mathbb{R}^d$ where d is the dimension of the feature. Here we omit the superscript for simplicity. For the text (target) distribution, CLIP first transforms all label C into text descriptions T using their class name and a handcrafted template like "`a photo of {class}`". Then the text encoder will encode these descriptions into text features by: $x_1 = E_T(T)$, with the same dimension as the image feature. Notably, our method is agnostic to the feature extraction process. While we use CLIP zero-shot as an example, other methods like CoOp can also be used to extract image and text features. In the experiment section, we discuss the influence of different feature extraction methods on FMA.

After obtaining all image and text features, FMA trains a velocity field to perform further alignment. For simplicity, we denote distributions p_0 and p_1 as the collections of image features and text features.

3.2 Training Stage

Typically, in flow matching training, we randomly sample an image feature $x_0 \sim p_0$ and a text feature $x_1 \sim p_1$ to compose a training pair. Following Rectified Flow (Liu et al., 2022), a transfer trajectory is defined as linear interpolation between x_0 and x_1 on time $t \in [0, 1]$: $x_t = tx_1 + (1 - t)x_0$. The conditional velocity field $v_t(x_t|x_1)$ is defined as the derivative of the trajectory with respect to time t: $v_t(x_t|x_1) = \frac{\mathrm{d}x_t}{\mathrm{d}t} = x_1 - x_0$. The marginal velocity field $u_t^\theta(x_t)$ can be learned by solving a simple least square regression problem between $v_t(x_t|x_1)$ and u_t^θ :

$$\mathcal{L}_{FM}(\theta) = \mathbb{E}_{x_t \sim p_t, t \sim [0,1]}[\|u_t^\theta(x_t) - (x_1 - x_0)\|^2], \tag{1}$$

where we use the distribution p_t to denote the collection of all intermediate features x_t. However, the learned velocity field learned $u_t^\theta(x_t)$ cannot guarantee class-level correspondence because it is an estimation of $v_t(x_t)$, an expectation of the $v_t(x_t|x_1)$ over all possible text features x_1:

$$v_t(x_t) = \int v_t(x_t|x_1) \frac{p_t(x_t|x_1)p_1(x_1)}{p_t(x_t)} \mathrm{d}x_1. \tag{2}$$

Consequently, the learned u_t^θ will drive an image feature towards the average of all text features, which usually results in incorrect classification.

Coupling Enforcement. To solve this issue, we propose coupling enforcement. Specifically, for a given image feature x_0, we exclusively sample its corresponding ground-truth text feature x_1. Due to the sparsity in the high-dimensional manifold, their transfer trajectories can be considered as mutually non-crossing when the dataset is small. This means for given x_t, there exists only one potential text feature x_1 in Eq. (2). Consequently, the marginal velocity is equivalent to the conditional one:

$$v_t(x_t) = v_t(x_t|x_1). \tag{3}$$

This means we are secretly using the conditional velocity field $v_t(x_t|x_1)$ to move the image feature to the direction of x_1. If x_1 corresponds to the right class label, the flow matching can then transform the image feature to the correct text feature in the target distribution.

While this coupling enforcement strategy is intuitive, it induces another new challenge: data scarcity. Since we only pair one image feature with its target text feature, the model is trained on only $\frac{1}{N}$ of the potential data compared with randomly pairing, where N is the number of classes in the dataset. As a result, large portions of the velocity field's domain remain unsampled, preventing it from providing reliable instructions to move image features.

Noise Augmentation. We design another strategy, noise augmentation, to solve this problem. It injects a time-dependent Gaussian noise into the intermediate features x_t during the training process. Adding random noise ensures the distribution x_t does not collapse to a low-dimensional manifold (Song & Ermon, 2019), thus learning more accurate velocity estimation. Concretely, the noise schedule is chosen as a time-dependent sequence. Inspired by Schrödinger bridge (Liu et al., 2023a), we sample the new noise-augmented features $\hat{x}_t$ from a Gaussian distribution:

$$\hat{x}_t \sim \mathcal{N}\left(\hat{x}_t | x_t, t \cdot (1-t) \cdot \sigma^2(x_t)\right). \tag{4}$$

After obtaining $\hat{x}_t$, the new ground truth $u_t^\theta(\hat{x}_t)$ is the direction pointing to the target text feature x_1:

$$v_t(\hat{x}_t|x_1) = \frac{x_1 - \hat{x}_t}{1-t}. \tag{5}$$

Then Eq. (1) can be applied for training $u^\theta(x_t)$. The training is summarized in Algorithm 1.

3.3 Inference Stage

After learning a velocity field $u_t^\theta(x_t)$, we use it to transform any image feature x_0 for classification. Vanilla flow matching inference process adapts an ODE solver, such as the Euler Method, to iteratively transform x_0 into a text feature x_1 by M steps:

$$x_{t+h} = x_t + h \cdot u_t^\theta(x_t), \tag{6}$$

where $h = \frac{1}{M}$ is the integration stepsize. x_1, is an approximation of a sample from p_1 and used for classification: first, calculate its cosine similarity with text features of all classes, then select the class with the highest similarity as the prediction.

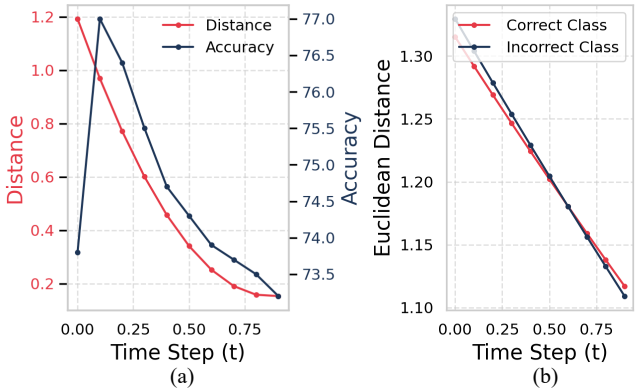

Figure 5: (a) Red line: Average distance to target figure at different timesteps. Black line: Accuracy using features at different timesteps for classification. (b) At different timesteps, the distance between the intermediate features and the correct/incorrect text feature.

However, this inference method may lead to the inconsistency issue in few-shot learning, as we mentioned before. Specifically, as shown in Figure 5(a), with t approaching 1, the distance between intermediate features x_t and their corresponding text features becomes smaller, which means they are indeed moving towards to the target distribution. However, the classification accuracy using x_t first increases, then decreases. This indicates that although the coupling enhancement strategy is adapted, some features may still move towards text features of incorrect classes. To better illustrate this, we show an example in Figure 5(b) where this inference method leads to a classification error. As the transformation continues, x_t becomes closer to the incorrect text feature than to the correct one. In this situation, the vanilla inference method, which use x_1 for classification, will lead to incorrect results.

Early Stopping Solver (ESS). To solve this problem, We propose a simple yet effective inference strategy: utilizing an early stopping solver (ESS) to terminate the transformation when the intermediate features are discriminative for classification. Specifically, instead of integrating over the full time interval $[0, 1]$, we fix the stepsize h in Eq. (6) as a constant and choose a constant number of inference steps M. $u_t^\theta(x_t)$ then transforms an initial image feature x_0 to an intermediate feature $x_{\hat{T}}$ at the final time $\hat{T} = h \cdot M$, as shown in Algorithm 2. Subsequently, $x_{\hat{T}}$ is used for classification. Notably, the optimal strategy is to find a sample-specific t for each input image feature, and we left this as a promising direction for future research.

Algorithm 1 Training

Input: paired image features and text features $\mathcal{D} = \{(x_0^i, x_1^i)\}_{i=0}^{N \times K}$
repeat
  $t \sim \mathcal{U}([0, 1])$, $(x_0, x_1) \sim \mathcal{D}$
  $x_t = (1 - t)x_0 + tx_1$
  $\hat{x}_t \sim \mathcal{N}(\hat{x}_t | x_t, t \cdot (1 - t) \cdot \sigma^2(x_t)))$
  Take gradient descent on $u_t^\theta(\hat{x}_t)$
until converges

Algorithm 2 Inference

Input: image features x_0, trained $u_t^\theta(x_t)$, steps M, stepsize h
$t = 0$
for $n = 1$ to M **do**
  $x_{t+h} = x_t + h \cdot u_t^\theta(x_t)$
  $t = t + h$
end for
return $x_{h \cdot M}$ as new image feature

4 EXPERIMENTS

Datasets. We evaluated FMA on the few-shot classification task. Specifically, We conducted experiments on 11 benchmarks, including Aircraft (Maji et al., 2013), EuroSAT (Helber et al., 2019), DTD (Cimpoi et al., 2014), SUN397 (Xiao et al., 2010), UCF101 (Soomro et al., 2012), StanfordCars (Krause et al., 2013), ImageNet (Deng et al., 2009), Flowers102 (Nilsback & Zisserman, 2008), Food101 (Bossard et al., 2014), OxfordPets (Parkhi et al., 2012), Caltech101 (Fei-Fei et al., 2004), sorted from difficult to easy. To better illustrate the effect of dataset difficulty on our method, we divided these datasets into two groups: the first five as the difficult set, and the remaining six as the easy set. For each dataset, we followed the standard protocol to split the training, validation, and test sets. For K-shot classification, we randomly sample K images from each class as the training set. Unless otherwise specified, we use pre-trained CLIP ViT-B/16 (Radford et al., 2021) as the backbone.

4.1 COMPARISONS WITH STATE-OF-THE-ARTS

Settings. We compared our FMA framework with 8 state-of-the-art methods, including CoOp (Zhou et al., 2022b), CoCoOp (Zhou et al., 2022a), CLIP-Adapter (Gao et al., 2024), Tip-Adapter (Zhang et al., 2022), PLOT++ (Chen et al., 2022), KgCoOp (Yao et al., 2023), ProGrad (Zhu et al., 2023) and CLIP-LoRA (Zanella & Ben Ayed, 2024). We implemented our FMA framework on top of CLIP-LoRA. Specifically, we first used a pre-trained CLIP-LoRA model to extract image and text features. After that, a velocity network was trained on these features according to FMA framework. Notably, the dataset for training CLIP-LoRA and the velocity field are kept the same, which means no extra knowledge is introduced. The AdamW (Loshchilov & Hutter, 2017) optimizer was used in FMA training with a learning rate of 0.0002, and we used a cosine annealing learning rate scheduler. For FMA inference, we set the default stepsize $h = 0.1$. To find the optimal number of inference steps M, we first attempted various values in the validation dataset, then chose the one with the highest performance as the final number of inference steps for test dataset.

Results. We reported the classification accuracy of all methods on 11 datasets in Table 1. For each dataset, we reported the results of 1-shot, 4-shot, and 16-shot classification. The results show that FMA achieves the best performance on most datasets. Notably, compared with simple datasets, FMA shows more significant effectiveness on difficult datasets. This validates our previous conclusion: Multi-step rectification is necessary when the cross-modal distribution is complicated to align.

4.2 GENERALIZATION ABILITY

Settings. While many PEFT approaches like CoOp can improve the performance of few-shot classification, they usually result in the degradation of the generalization ability.

Therefore, we explored whether FMA will cause this problem by evaluating the cross-dataset transferability. Specifically, we chose Imagenet to train the velocity field based on CoOp, but tested it on difficult and easy datasets respectively. To conduct a more comprehensive evaluation, we not only reported the generalization (cross-dataset transfer) results, but also provided the adaptation (few-shot learning) performance and its harmonic mean.

Table 2: **Evaluation of the generalization ability of FMA.** "D" and "E" mean average performance on difficult and easy datasets, respectively.

Method	Adaptation		Generalization		Harmonic	
	D	E	D	E	D	E
CLIP	49.1	77.8	47.6	80.0	48.3	78.9
CoOp	71.4	87.1	44.0	**76.3**	54.4	81.3
+FMA	**73.8**	**88.0**	**44.3**	75.9	**55.4**	**81.5**

Table 1: **Comparison with other state-of-the-art approaches.** Based on CLIP-LoRA, we add FMA to further improve the performance. The highest value of each dataset is bolded.

Shots	Method	Difficult						Easy						
		Aircraft	SAT	DTD	SUN	UCF	Avg	Cars	Net	Flowers	Food	Pets	Caltech	Avg
0	CLIP (2021)	24.8	47.8	43.8	62.5	66.7	47.6	65.5	66.7	67.4	85.3	89.1	92.9	77.7
1	CoOp (2022b)	20.8	56.4	50.1	67.0	71.2	53.1	67.5	65.7	78.3	84.3	90.2	92.5	79.8
	CoCoOp (2022a)	28.1	55.4	52.6	68.7	70.4	55.0	67.6	69.4	73.4	84.9	91.9	94.1	80.2
	TIP-Adapter (2022)	28.8	67.8	51.6	67.2	73.4	57.8	67.1	69.4	83.8	85.8	90.6	94.0	81.8
	CLIP-Adapter (2024)	25.2	49.3	44.2	65.4	66.9	50.2	65.7	67.9	71.3	86.1	89.0	92.0	78.7
	PLOT++ (2022)	28.6	65.4	54.6	66.8	74.3	58.0	68.8	66.5	80.5	86.2	91.9	94.3	81.4
	KgCoOp (2023)	26.8	61.9	52.7	68.4	72.8	56.5	66.7	68.9	74.7	**86.4**	92.1	94.2	80.5
	ProGrad (2023)	**28.9**	57.0	52.8	67.0	73.3	55.8	68.2	67.0	80.9	84.9	91.4	93.5	81.0
	CLIP-LoRA (2024)	28.0	71.9	54.1	**70.3**	75.4	60.6	69.4	70.3	81.4	85.1	91.9	93.8	82.0
	+FMA (Ours)	28.3	**73.0**	**55.1**	**70.6**	**75.9**	**61.5**+0.9	**69.8**	70.2	**84.9**	85.2	**92.1**	**94.5**	**82.8**+0.8
4	CoOp (2022b)	30.9	69.7	59.5	69.7	77.6	58.7	74.4	68.8	92.2	84.5	92.5	94.5	84.5
	CoCoOp (2022a)	30.6	61.7	55.7	70.4	75.3	64.2	69.5	70.6	81.5	86.3	92.7	94.8	82.6
	TIP-Adapter (2022)	35.7	76.8	59.8	70.8	78.1	52.8	74.1	70.7	92.1	86.5	91.9	94.8	85.0
	CLIP-Adapter (2024)	27.9	51.2	46.1	68.0	70.6	66.5	67.5	68.6	73.1	86.5	90.8	94.0	80.1
	PLOT++ (2022)	35.3	83.2	62.4	71.7	79.8	62.4	76.3	70.4	92.9	86.5	**92.7**	95.1	85.6
	KgCoOp (2023)	32.2	71.8	58.7	71.5	77.6	62.6	69.5	69.9	87.0	86.9	92.6	95.0	83.5
	ProGrad (2024)	34.1	69.6	59.7	71.7	77.9	68.0	75.0	70.2	91.1	85.4	92.1	94.4	84.7
	CLIP-LoRA (2024)	38.8	83.5	64.0	72.8	81.1	69.7	77.4	71.4	92.9	82.6	90.6	95.0	85.0
	+FMA (Ours)	**40.3**	**85.0**	**67.0**	**73.7**	**82.4**	**71.5**+1.8	**78.9**	**72.0**	**95.0**	**83.2**	90.8	**95.8**	**86.0**+1.0
16	CoOp (2022b)	43.3	86.0	70.0	74.9	83.1	64.8	83.1	71.4	97.2	84.4	91.1	95.5	87.1
	CoCoOp (2022a)	33.8	75.5	65.8	72.8	76.0	72.2	72.4	71.1	87.1	87.4	93.2	95.2	84.4
	TIP-Adapter (2022)	44.6	85.9	70.8	76.0	83.9	63.9	82.3	73.4	96.2	86.8	92.6	95.7	87.8
	CLIP-Adapter (2024)	34.2	71.4	59.4	74.2	80.2	72.3	74.0	69.8	92.9	87.1	92.3	94.9	85.2
	PLOT++ (2022)	46.7	**92.0**	71.4	76.0	85.3	63.9	84.6	72.6	97.6	87.1	**93.6**	96.0	88.6
	KgCoOp (2023)	36.5	76.2	68.7	73.3	81.7	74.3	74.8	70.4	93.4	**87.2**	93.2	95.2	85.7
	ProGrad (2024)	43.0	83.6	68.8	75.1	82.7	70.6	82.9	72.1	96.6	85.8	92.8	95.9	87.7
	CLIP-LoRA (2024)	54.7	90.7	73.0	76.0	86.2	76.1	86.0	73.4	97.9	84.2	91.6	96.1	88.2
	+FMA (Ours)	**57.8**	91.0	**75.4**	**77.2**	**87.1**	**77.7**+1.6	**87.7**	**73.5**	**99.1**	85.1	91.6	**96.5**	**88.9**+0.7

Table 3: **Model agnostic results**. We implemented FMA on several PEFT approaches using 16-shot settings.

Method	Difficult						Easy						
	Aircraft	SAT	DTD	SUN	UCF	Avg	Cars	Net	Flowers	Food	Pets	Caltech	Avg
CLIP (2021)	24.8	47.8	43.8	62.5	66.7	49.1	65.5	66.7	67.4	85.3	89.1	92.9	77.8
+FMA	46.4	87.9	72.0	73.3	83.1	72.5+23.4	82.9	71.0	98.5	87.0	92.7	95.8	88.0 +10.2
CoOp (2022b)	43.2	86.0	70.0	74.9	83.1	71.4	83.1	71.4	97.2	84.4	91.1	95.5	87.1
+FMA	47.6	88.1	73.1	75.9	84.4	73.8+2.4	85.4	72.5	98.2	85.0	91.4	95.7	88.0 +0.9
CoCoOp (2022a)	33.8	75.5	65.8	72.8	76.0	64.8	72.4	71.1	87.1	87.4	93.2	95.2	84.4
+FMA	36.9	86.9	71.9	73.4	80.3	69.9+5.1	73.5	71.9	94.5	87.8	93.4	95.6	86.1+1.7
CLIP-Adapter (2024)	33.8	70.4	59.3	74.3	80.1	63.6	74.2	71.6	93.6	87.1	92.4	94.9	85.6
+FMA	35.8	85.6	69.2	74.4	81.5	69.3+5.7	74.7	71.3	95.6	87.2	92.9	96.0	86.3+0.7
CLIP-LoRA (2022)	54.7	90.7	73.0	76.0	86.2	76.1	86.0	73.4	97.9	84.2	91.6	96.1	88.2
+FMA	57.8	91.0	75.4	77.2	87.1	77.7+1.6	87.7	73.5	99.1	85.1	91.6	96.5	88.9+0.7

Results. The results are shown in Table 2. While FMA improves the few-shot learning performance (first two columns), it doesn't result in further degradation of the generalization ability (middle two columns). Meanwhile, the improvement of the harmonic mean indicates FMA achieves a better trade-off between the adaptation and generalization ability.

4.3 Ablation Study

Architecture Agnostic. In this section We evaluated whether our FMA is effective on other PEFT approaches. Specifically, we implemented FMA on pre-trained CLIP and four different PEFT methods: CoOp, CoCoOp, CLIP-Adapter, and CLIP-LoRA. For each PEFT approach, we first fine-tuned the pre-trained ViT-B/16 CLIP accordingly. After that, we extracted image and text features using the pre-trained or fine-tuned CLIP. Finally, we trained a velocity network on these features using the FMA framework. We reported the performance on difficult datasets.

Results. As shown in Table 3, FMA achieves consistent performance improvements on all approaches, which demonstrates the architecture-agnostic ability of our framework. This means FMA can be easily adapted to different PEFT methods without significant performance degradation. Meanwhile,

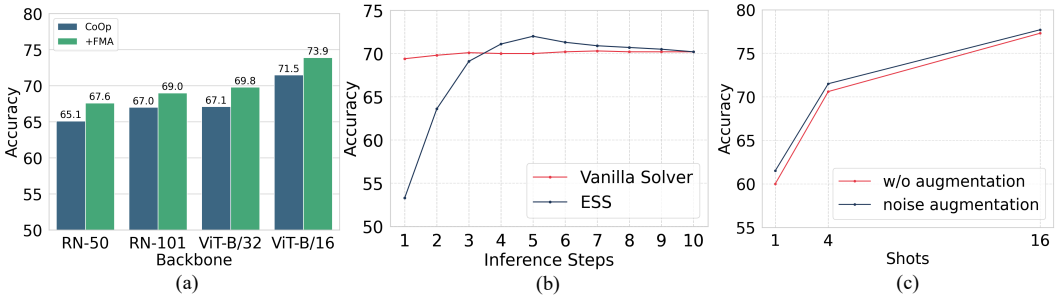

Figure 6: (a) Performance of different CLIP backbones on difficult datasets. (b) Ablation on different inference strategies. (c) Ablation on noise augmentation strategy. The average performance on 11 datasets is reported.

across all approaches, improvements on difficult datasets consistently surpass those on easy datasets. This further supports our conclusion that for difficult datasets with complex cross-modal distribution, multi-step rectification is more effective.

Different CLIP Backbones. This experiment was to evaluate whether FMA is effective on different CLIP backbones. We implemented FMA on top of CoOp and chose four different CLIP backbones: ResNet50, ResNet101, ViT-B/32, and ViT-B/16. Specifically, we first used a trained CoOp model to extract image and text features. After that, we trained a velocity network on these features according to our FMA framework. We reported the average performance on difficult datasets.

Results. The accuracy result is shown in Figure 6(a). Across all backbones, FMA achieves better performance than CoOp, demonstrating the effectiveness of our method. This also shows that FMA can be easily adapted to different backbones and further boost their performance.

Different Inference Strategies. To evaluate the influence of inference strategies on FMA, we conducted experiments with different numbers of inference steps using two kinds of strategies: Vanilla flow matching solver (simulate Eq. (6) over the entire time interval $[0, 1]$) and our early stopping solver. Specifically, we used the default stepsize $h = 0.1$ and enumerated inference steps $M \in [0, 10]$. We conducted the experiments with a setting of 16 shots on the DTD dataset.

Results. As shown in Figure 6(b), as the number of inference steps increases, the performance of ESS will decrease after a certain step number (8 in this case). This is because training a perfect velocity field is extremely difficult, so more inference steps mean more accumulated errors. This suggests that choosing a proper number of inference steps is crucial for achieving good performance. Additionally, compared with the vanilla solver that is not sensitive to inference steps, ESS achieves better results when an appropriate inference step is chosen.

Noise Augmentation. To show the influence of noise augmentation in FMA, we trained two velocity networks on top of CLIP-LoRA, one with noise augmentation and another without any augmentation technique. We kept all other parts in FMA the same, such as the feature extraction and inference. In the inference, we use ESS to transform image features for classification.

Results. Performance on difficult datasets is shown in Figure 6(c). We can see that noise augmentation can improve the performance of FMA compared with no augmentation. And this improvement is consistent across different sizes of the training dataset, which proves the effectiveness of this strategy in the velocity field training process.

5 Conclusion

In this paper, we study how to achieve better cross-modality alignment based on pre-trained VLMs. Although many PEFT methods have been proposed, we find that these methods usually struggle on difficult datasets. In this paper, we conclude that these approaches are "one-step", which are not enough to decouple and align features in difficult datasets. To solve this, we propose our FMA framework to utilize the multi-step rectification ability of flow matching. In the training of FMA, we design two methods, coupling enforcement and noise augmentation, to learn a velocity field that can maintain class correspondence. Additionally, we propose an early-stopping solver for better inference performance. FMA is a plug-and-play module and we have conducted extensive experiments to demonstrate its effectiveness over a variety of benchmarks. We hope our research can explore a new potential opportunity, *i.e.*, multi-step rectification in the few-shot learning area.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Adrian Bulat and Georgios Tzimiropoulos. Lasp: Text-to-text optimization for language-aware soft prompting of vision & language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 23232–23241, 2023.

Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Plot: Prompt learning with optimal transport for vision-language models. *arXiv preprint arXiv:2210.01253*, 2022.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Victor G Turrisi da Costa, Nicola Dall'Asen, Yiming Wang, Nicu Sebe, and Elisa Ricci. Diversified in-domain synthesis with efficient fine-tuning for few-shot classification. *arXiv preprint arXiv:2312.03046*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178. IEEE, 2004.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Ju He, Qihang Yu, Qihao Liu, and Liang-Chieh Chen. Flowtok: Flowing seamlessly across text and image tokens. *arXiv preprint arXiv:2503.10772*, 2025.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

Jae Myung Kim, Jessica Bader, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. Datadream: Few-shot guided dataset generation. In *European Conference on Computer Vision*, pp. 252–268. Springer, 2024.

Jeongsol Kim, Yeobin Hong, and Jong Chul Ye. Flowalign: Trajectory-regularized, inversion-free flow-based image editing. *arXiv preprint arXiv:2505.23145*, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. *arXiv preprint arXiv:2412.08629*, 2024.

Dongjun Lee, Seokwon Song, Jihee Suh, Joonmyeong Choi, Sanghyeok Lee, and Hyunwoo J Kim. Read-only prompt optimization for vision-language few-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1401–1411, 2023.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. Image-to-image schrödinger bridge. *arXiv preprint arXiv:2302.05872*, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023b.

Qihao Liu, Xi Yin, Alan Yuille, Andrew Brown, and Mannat Singh. Flowing from words to pixels: A noise-free framework for cross-modality evolution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2755–2765, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. *arXiv preprint arXiv:2410.10792*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6757–6767, 2023.

Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Interventional few-shot learning. *Advances in neural information processing systems*, 33:2734–2746, 2020.

Maxime Zanella and Ismail Ben Ayed. Low-rank few-shot adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1593–1603, 2024.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaptation of clip for few-shot classification. In *European conference on computer vision*, pp. 493–510. Springer, 2022.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16816–16825, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022b.

Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15659–15669, 2023.