Appendix

A. Additional Experiments

Advantage of token pruning in synthetic data beyond acceleration. When using inversion to generate data by optimizing Eq. (1), it primarily crafts only the label-relevant features into the synthetic data (typically the foreground regions), while background regions often remain noisy as initialization. To further illustrate this, the experiment in Tab. 6 shows that, during the inversion process, the classification loss Eq. (1) caused by the identified foreground steadily decreases, whereas the loss caused by the identified background remains nearly unchanged. As such, masking the background in synthetic data helps to reduce noise, which aligns with our experimental results in Tab. 2: pruning background tokens in the synthetic CIFAR-FS data led to a 1.34% improvement in performance.

Table 6. Loss tracking during inversion process.

Area	Classification Loss Change in Eq. (1)
Inverted Backgrounds	10.78 → 10.72
Inverted Foregrounds	10.78 → 0.12

Effect of cross-task interpolation. Tab. 7 verifies the effectiveness of the cross-task interpolation under a constrained LoRA budget of 100 on CIFAR-FS. This technique can diversify the task distribution by generating multiple interpolated tasks, which enables the meta-training to cover a broader range of tasks, thereby bolstering the generalization capabilities for unseen tasks.

Table 7. Effect of cross-task interpolation.

Ablation	5-way 1-shot	5-way 5-shot
w/o cross-task interpolation	87.97	96.81
w/ cross-task interpolation	89.69	97.05

Effect of meta-learning in LoRA Recycle. To assess the effectiveness of meta-learning, we compared it against joint supervised learning within the LoRA Recycle framework. In joint supervised learning, data from all LoRAs is aggregated to train a single LoRA through standard supervised learning. The comparative results for meta-learning and joint supervised learning are presented in Tab. 8. Experiments were conducted on the CIFAR-FS dataset, focusing on unseen few-shot tasks to evaluate generalization capabilities. As shown, meta-learning achieves significantly better performance than joint supervised learning in both 1-shot and 5-shot settings. This improvement arises because meta-learning's bi-level optimization is inherently designed to enhance generalization to unseen few-shot tasks.

Table 8. Effect of meta-learning in LoRA Recycle.

Ablation	5-way 1-shot	5-way 5-shot
joint supervised learning	78.22	94.56
meta-learning	89.69	97.05

Effect of synthetic data in LoRA Recycle. In our setting, the original training data for each LoRA is unavailable. To evaluate the effectiveness of the synthetic data, we use the results obtained from the original training data as an upper bound for comparison. As shown in Table 9, the performance on CIFAR-FS achieved with synthetic data is close to that obtained with the original training data, demonstrating the effectiveness of the synthetic data in LoRA Recycle.

Table 9. Effect of synthetic data in LoRA Recycle.

Ablation	5-way 1-shot	5-way 5-shot
original training data	91.21	98.93
synthetic data	89.69	97.05

Meta-learn what? Our framework meta-trains an extra lightweight LoRA while keeping the original VFM frozen. Based on the results shown in Tab. 10, we summarize some findings: (i) Meta-training the entire VFM is inferior to only meta-training the extra LoRA. Meta-training the entire VFM might distort the original feature space [41], leading to bias to meta-training tasks and heavy costs of computation and storage. Meta-training the extra LoRAs can preserve the knowledge of foundation models learned from large-scale pretraining while injecting task-specific knowledge into extra LoRAs. (ii) Only meta-training the last 6 LoRA layers can outperform meta-training all LoRA layers. The improvements are more obvious in 5-way 1-shot learning, suggesting that reducing learnable parameters possibly avoids overfitting with limited training data. Only meta-training the first 6 LoRA layers is less effective. This is because only updating the shallow layers is insufficient to develop effective representations compared with updating the deep layers.

Table 10. Meta-learn what?

Learnable Parts	5-way 1-shot	5-way 5-shot
Entire VFM	88.40	95.73
LoRA (all 12 layers)	89.69	**97.05**
LoRA (the first 6 layers)	85.45	95.13
LoRA (the last 6 layers)	**90.40**	96.10

Effect of constructed mask. We use the constructed mask instead of attention weights to prune tokens before the first layer in meta-training. This choice is motivated by the observation that shallow-layer attention weights are less accurate than deep-layer weights [39, 54], as shallow layers lack the fine-grained information captured by deeper layers. By leveraging final-layer attention information, our constructed mask offers more accurate guidance to identify the most informative tokens in the synthetic image. As shown in Tab. 11, under the same pruning ratio (75%), using the con-

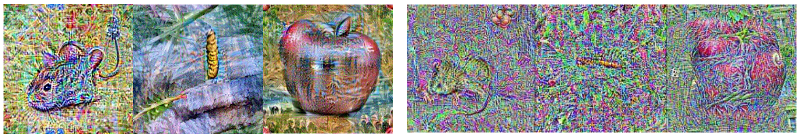


Figure 5. Visualization of synthetic images with (left) and without (right) the naturalness prior $\mathcal{R}_{\text{BN}}$.

structed mask yields significantly better performance than using first-layer attention on CIFAR-FS.

Table 11. Effect of constructed mask.

Ablation	5-way 1-shot	5-way 5-shot
first-layer attention weight	83.62	92.43
constructed mask	89.70	96.69

Experiments on more challenging dataset, Meta-Dataset [61]. We evaluate our proposed LoRA Recycle framework on the Meta-Dataset, a benchmark specifically designed to test few-shot learning models across a variety of challenging domains. This dataset provides a rigorous evaluation setting. The results, summarized in Tab. 12, demonstrate the effectiveness of LoRA Recycle compared to other baselines. Notably, LoRA Recycle achieves superior performance in both the 5-way 1-shot and 5-way 5-shot settings, while also offering the advantage of being fine-tuning-free.

Table 12. Experiments on more challenging Meta-Dataset.

Method	Fine-Tuning-Free	5-way 1-shot	5-way 5-shot
MOLE	✗	61.87	76.31
LoRAHub	✗	63.14	77.24
LoRA Recycle (ours)	✓	68.48	80.12

Experiments on more types of Vision Transformers. In this section, we evaluate the performance of our LoRA Recycle framework across multiple Vision Transformers on the CIFAR-FS dataset, further demonstrating its generalizability. We experiment with three popular Vision Transformer architectures: ViT-B (CLIP), DeiT-B [58], and LV-ViT-M [37]. Each model is compared using LoRAHub as a baseline. Tab. 13 presents the results of these experiments. The performance is evaluated in both 5-way 1-shot and 5-way 5-shot scenarios. As shown, LoRA Recycle consistently outperforms the LoRAHub baseline, while also offering the advantage of being fine-tuning-free.

Table 13. Experiments on more types of Vision Transformers on CIFAR-FS.

Model	Method	Fine-Tuning-Free	5-way 1-shot	5-way 5-shot
ViT-B (CLIP)	LoRAHub	✗	81.02	96.24
	LoRA Recycle (ours)	✓	91.03	97.05
DeiT-B [58]	LoRAHub	✗	79.52	93.32
	LoRA Recycle (ours)	✓	88.31	94.72
LV-ViT-M [37]	LoRAHub	✗	80.42	94.23
	LoRA Recycle (ours)	✓	89.52	95.35

Experiments on tasks beyond few-shot learning. In this section, we extend our evaluation to zero-shot classification tasks, demonstrating the versatility of LoRA Recycle beyond few-shot learning. To enable zero-shot classification, we recycle pre-tuned LoRAs from CLIP by replacing the classification loss used in Eq. (1) and Eq. (4) with the contrastive loss employed by CLIP. Tab. 14 presents the results on the Meta-Dataset for zero-shot classification. As shown, LoRA Recycle significantly outperforms other baseline methods, including MOLE and LoRAHub, both of which require fine-tuning. LoRA Recycle, being fine-tuning-free, achieves a higher accuracy, illustrating its effectiveness in adapting to zero-shot classification tasks.

Table 14. Experiments on zero-shot classification on Meta-Dataset.

Method	Fine-Tuning-Free	5-way 0-shot
MOLE	✗	59.36
LoRAHub	✗	60.25
LoRA Recycle (ours)	✓	64.52

Recycle LoRAs with different ranks. Tab. 15 verifies the architecture-agnostic feature of our LoRA Recycle approach. Our approach can reuse pre-tuned LoRAs with different ranks (e.g., 50% LoRAs with the rank of 4 and 50% LoRAs with the rank of 8). This is a distinctive advantage absent in existing baselines, thereby extending its practical applicability across various real-world scenarios.

Table 15. Architecture-agnostic property of our framework. We conduct experiments on CIFAR-FS and set the rank of meta-LoRA as 4. We reuse pre-tuned LoRAs with different ranks (e.g., 50% LoRAs with the rank of 4 and 50% LoRAs with the rank of 8).

Rank of pre-tuned LoRAs	5-way 1-shot	5-way 5-shot
100%: **4**	89.69	97.05
50%: **4** + 50%: **8**	90.67	97.12

Cross validation. Tab. 16 shows our consistent superiority compared with other baselines by exchanging meta-training and meta-testing domains.

Results of ViT-B/32. Tab. 17 show the results when using ViT-B/32 with a 32 × 32 input patch size as the implementation of VFM. In the "recycle in-domain LoRAs" scenario, our LoRA Recycle consistently outperforms the best fine-tuning-based baselines by a large margin, up to 8.93% and 1.40% for 1-shot and 5-shot learning, respectively. It also exceeds the leading fine-tuning-free baselines by up to 10.39% and 2.89% for 1-shot and 5-shot learning, respec-

Table 16. Cross validation by exchanging meta-training and meta-testing domains. [meta-training domains]→[meta-testing domain]. $\mathcal{D}_1$: MiniImageNet, $\mathcal{D}_2$: CUB, $\mathcal{D}_3$: CropDiseases. 51: 5-way 1-shot. 55: 5-way 5-shot.

Method	$[D_2, D_3] \to [D_1]$		$[D_1, D_3] \to [D_2]$		$[D_1, D_2] \to [D_3]$	
	51	55	51	55	51	55
LoRAHub + NN	81.02	93.18	85.27	95.23	76.21	92.31
LoRA Recycle$_{75}$ (ours)	**86.12**	**95.03**	**90.02**	**97.12**	**80.19**	**94.02**

tively. Fig. 6 shows the visualization of synthetic images and their masked versions generated from ViT-B/32.

Visualization of masked synthetic images at varying sparsity levels. Fig. 7 illustrates synthetic images masked at varying sparsity levels. As we can see, only a subset of tokens carry meaningful semantic information and contribute to the final predictions, while the rest often represent noise, constructed as hallucinations of the VFM's misinterpretations. Our method can effectively filter out those noisy tokens and preserve the meaningful tokens, thus effectively preventing VFM from overfitting to irrelevant noise.

Comparison with SOTA model inversion approach. Fig. 8 illustrates that the quality of our model inversion approach surpasses current state-of-the-art (SOTA) methods like CMI [11], which typically produce simpler, lower-resolution images from shallow pre-trained models. Our approach excels in three key areas: (i) quality, producing higher fidelity images; (ii) resolution, capable of generating complex images with higher resolutions of 224 × 224; and (iii) efficiency, with our double-efficient mechanism significantly accelerating the model inversion process. Moreover, our work investigates the inversion from transformer-based models, whereas existing methods mainly concentrate on convolutional architectures such as ResNet.

T-SNE visualization. Fig. 9 presents the t-SNE visualizations of images generated from LoRAs pre-tuned on diverse datasets, including CIFAR-FS; MiniImageNet, VGG-Flower, and CUB. Our model inversion approach successfully inverts the essential discriminative features.

Effect of the naturalness prior. Fig. 5 shows the efficacy of the regularization term $\mathcal{R}_{\text{BN}}$ in Eq. (1) to enhance the realism of images by enriching natural color and smoothing noise. We set the coefficient α_{BN} as 0.01.

B. Preliminary of Vision Transformers (ViTs)

Preliminary of ViTs. Here, we discuss the operational mechanism behind ViTs. ViTs initially divide the input image $\boldsymbol{X}^{\text{I}}$ belonging to the space $\mathbb{R}^{H \times W \times C}$ into $n+1$ distinct, non-overlapping patches. These patches are then transformed into $n+1$ tokens, denoted as $\boldsymbol{X}^{\text{I}} = [\boldsymbol{x}_{\texttt{[CLS]}}, \boldsymbol{x}_1, ..., \boldsymbol{x}_n]$ where $\boldsymbol{x}_i \in \mathbb{R}^D$. The class token, $\boldsymbol{x}_{\texttt{[CLS]}}$, is prepended to these image tokens to facilitate the classification task. To integrate positional relationships, learnable position encodings are added to all tokens. These tokens are then processed through multiple ViT layers, which are composed of multi-head self-attention (MHSA) modules and feed-forward networks (FFN). Within each MHSA, the token set $\boldsymbol{X}^{\text{I}}$ undergoes the transformation into three distinct matrices: the query $\boldsymbol{Q}$, key $\boldsymbol{K}$, and value $\boldsymbol{V}$ matrices. The formulation of the attention mechanism is given by

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right)\boldsymbol{V}, \quad (7)$$

where d represents the dimension of the query vectors within $\boldsymbol{Q}$. We define $\boldsymbol{A}$ as the square matrix representing the attention weights across all token pairs, calculated as $\boldsymbol{A} = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right)$, with dimensions $\mathbb{R}^{(n+1)\times(n+1)}$. Specifically, $\boldsymbol{a}_i$, which is the i^{th} row of $\boldsymbol{A}$, signifies the attention weights of token $\boldsymbol{x}_i$ with respect to all tokens. Particularly, $\boldsymbol{a}_{\texttt{[CLS]}}$ refers to $\boldsymbol{a}_0$. Based on Eq. (7), the i^{th} output token can be viewed as a linear combination of all tokens' value vectors $[\boldsymbol{v}_{\texttt{[CLS]}}, \boldsymbol{v}_1, ..., \boldsymbol{v}_L]$, weighted by $\boldsymbol{a}_i$. These output tokens are subsequently forwarded to the FFN, which consists of two linear layers and an activation function. At the final ViT layer, the class token $\boldsymbol{x}_{\texttt{[CLS]}}$, summarizing the global image representation, is utilized as the classifier's input to predict the image's classification probability distribution.

Computational complexity of ViTs. Given an image split into N patches, each with an embedding dimension of D, the computational complexities of self-attention (SA) and feed-forward network (FFN) in ViTs are :

$$O(\text{SA}) = 3ND^2 + 2N^2D, \;\; O(\text{FFN}) = 8ND^2. \quad (8)$$

Since the complexities of SA and FFN scale respectively quadratically and linearly with N, our proposed double-efficient mechanism (see Sec. 4.3) significantly reduces the computational complexity by reducing the number of tokens.

C. Hyperparameter Selection and Sensitivity Analysis

In this section, we detail the selection of hyperparameters and conduct a sensitivity analysis on key hyperparameters. Generally speaking, We base our hyperparameter values on reference works and perform grid searches within the relevant ranges to identify the optimal configurations.

For the learning rate in LoRA Inversion, we refer to the settings from prior work [80], and perform a grid search over the range $[0.1, 0.25, 0.5]$. Similarly, for the learning rate in the meta-learning stage, we adopt values from the literature [56] and conduct a grid search over the range $[0.001, 0.01, 0.1]$. These ranges allow us to identify the optimal configurations.

Table 17. Recycle in-domain LoRAs. VFM is implemented with ViT-B/32. **FT** refers to fine-tuning-based baselines and **FTF** refers to fine-tuning-free baselines. **LoRA Recycle**$_x$ indicates $x\%$ tokens in synthetic data are masked (*i.e.*, different sparsity ratios). For a fair comparison between different sparsity ratios, we perform token pruning at the same layer (*i.e.*, at the last layer). Superscripts represent performance gains over the best FT baselines, while subscripts indicate gains over the best FTF baselines.

	Method	CIFAR-FS		MiniImageNet		Flower-VGG		CUB	
		5-way 1-shot	**5-way 5-shot**	**5-way 1-shot**	**5-way 5-shot**	**5-way 1-shot**	**5-way 5-shot**	**5-way 1-shot**	**5-way 5-shot**
FT	Full Finetuning	20.02	20.32	20.07	20.01	20.00	20.08	20.01	20.03
	Linear-probe	76.92	92.93	81.28	92.95	85.12	96.71	78.76	94.88
	Lora + Linear	76.44	94.85	79.20	93.60	83.17	96.57	76.39	95.43
	P > M > F	77.45	94.92	79.31	93.02	84.53	96.46	77.42	96.41
	Loras Avg + Linear	78.35	95.03	79.97	93.61	85.00	96.64	78.96	95.31
	MOLE	78.62	95.23	79.41	93.43	85.12	96.43	79.02	95.38
	LoraHub	79.48	95.36	80.12	93.93	85.63	96.69	79.54	95.48
FTF	NN	75.69	91.91	78.38	92.55	86.47	96.62	77.71	93.99
	Loras Avg + NN	77.05	92.56	79.63	92.60	84.21	96.35	76.32	93.61
	CAML	78.02	93.23	80.83	93.14	85.35	96.54	78.02	94.12
	LoRA Recycle	87.37	95.93	84.65	95.03	$\mathbf{91.92}^{(+6.29\%)}_{(+5.45\%)}$	97.65	$\mathbf{85.81}^{(+6.27\%)}_{(+7.79\%)}$	$\mathbf{95.95}^{(+0.47\%)}_{(+1.83\%)}$
	LoRA Recycle$_{25}$	87.91	96.09	84.93	95.06	90.49	$\mathbf{97.73}^{(+1.02\%)}_{(+1.11\%)}$	84.61	95.73
	LoRA Recycle$_{50}$	$\mathbf{88.41}^{(+8.93\%)}_{(+10.39\%)}$	$\mathbf{96.12}^{(+0.76\%)}_{(+2.89\%)}$	$\mathbf{85.61}^{(+4.33\%)}_{(+4.78\%)}$	$\mathbf{95.33}^{(+1.40\%)}_{(+2.19\%)}$	90.29	97.52	84.85	95.57
	LoRA Recycle$_{75}$	85.99	95.41	83.75	94.56	89.89	97.72	84.09	95.27

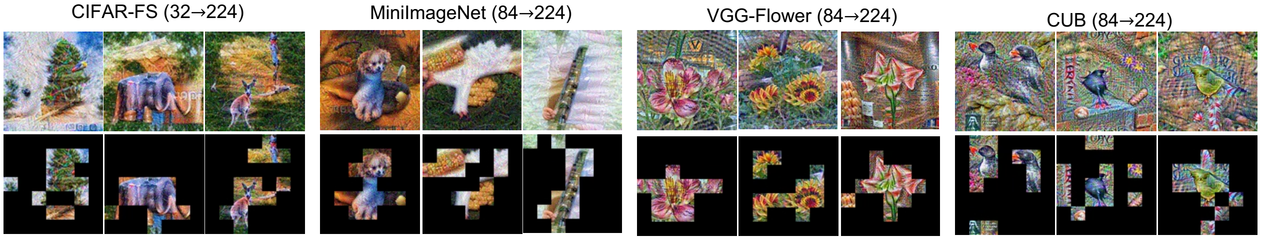


Figure 6. Visualization of synthetic images (odd line) and their 75% token-masked versions (even line) from ViT-B/32. (32 → 224) denotes the original training images' resolution is 32 × 32 while we can reconstruct images with a higher resolution of 224 × 224. Note that the size of each patch is 32 × 32, instead of 16 × 16.

We further conduct sensitivity analysis of the hyperparameter $\alpha_{\mathcal{R}}$ in Eq. (1), as it controls the balance during the inversion process. To analyze this, we conducted experiments on the CIFAR-FS dataset in both 5-way 1-shot and 5-way 5-shot settings. Tab. 18 shows the results, where we varied the value of $\alpha_{\mathcal{R}}$ to observe its effect on accuracy. Our sensitivity analysis reveals that our framework is not very sensitive to changes in $\alpha_{\mathcal{R}}$, although there are some variations among different $\alpha_{\mathcal{R}}$ values. This stability simplifies the hyperparameter tuning process, making our framework easier to apply in real-world applications.

Table 18. Sensitivity analysis of $\alpha_{\mathcal{R}}$ in Eq. (1) on CIFAR-FS.

Hyperparameter	**5-way 1-shot**	**5-way 5-shot**
0.1	89.35	96.39
0.01	89.70	96.69
0.001	88.83	95.76

D. Implementation Details of baselines

Here, we provide detailed implementation details for the baselines used in our paper..

- **Fine-tuning baselines.** "Full Fine-Tuning" updates the entire model on the target task via gradient descent. "Linear probe" only updates the classification head. "LoRA + Linear [26]" updates the layer-wise rank decomposition matrices and the classification head. For fine-tuning, we select the best results from learning rates $[0.1, 0.01, 0.001]$. For LoRA, we set the rank to 4.
- **Multi-LoRAs composition baselines.** "LoRAs Avg" refers to averaging all given pre-tuned LoRAs into a single LoRA, which can be further fine-tuned with the classification head ("LoRAs Avg + Linear") or directly make inference via Nearest Neighbour ("LoRAs Avg + NN") without fine-tuning. "LoRAHub [33]" takes a further step which obtains a single LoRA by a weighted sum of given pre-tuned LoRAs, where the weight values are fine-tuned on the target task. "MOLE [6]" fine-tunes a learnable gating function to composing the outputs of different LoRAs. For LoRAHub, we use a gradient-free approach to fine-tune the coefficients of pre-tuned LoRAs, following the setup in the original paper. For MOLE, we use gradient descent to fine-tune the learnable gating function. We select the best fine-tuning results from learning rates $[0.1, 0.01, 0.001]$.
- **Few-shot adaptation.** The current state-of-the-art baseline, P > M > F [27], performs few-shot adap-

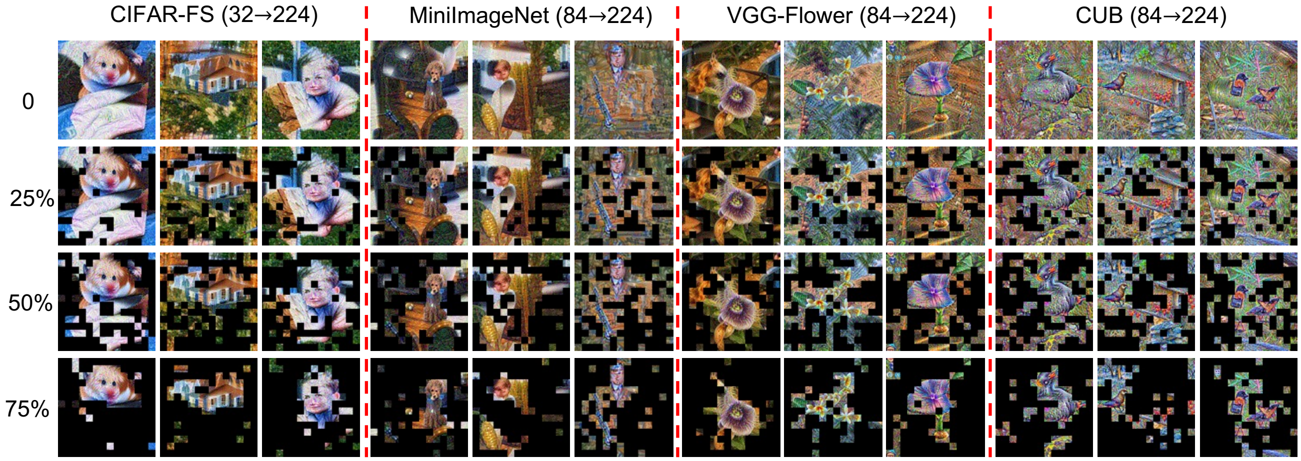


Figure 7. Visualization of masked synthetic images at varying sparsity levels. (32 → 224) denotes the original training images' resolution is 32 × 32 while we can reconstruct images with a higher resolution of 224 × 224.

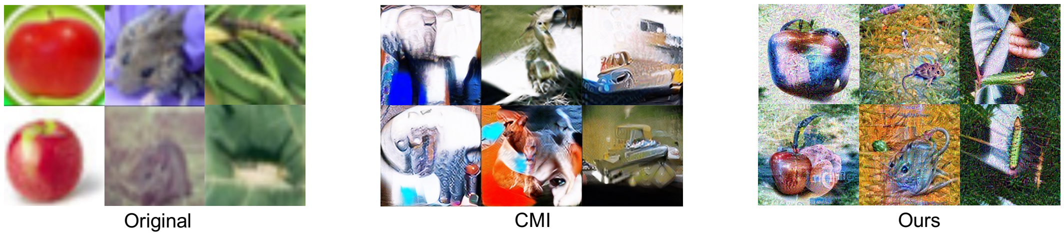
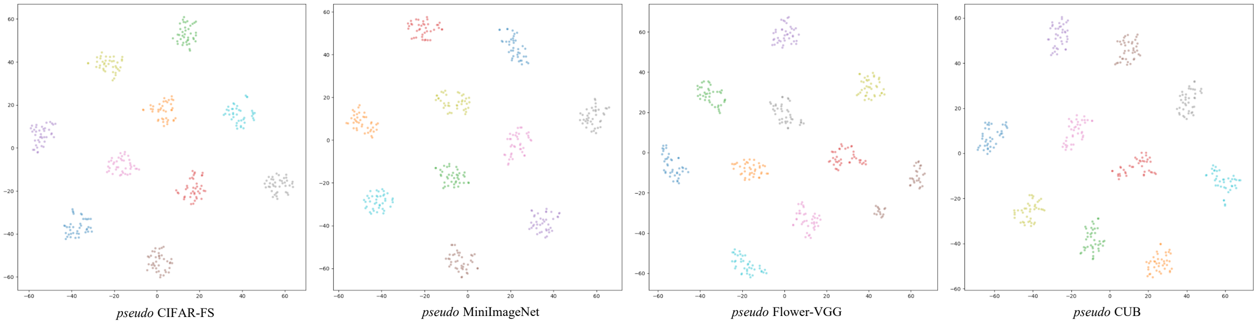


Figure 8. Comparison with SOTA model inversion approach. Our model inversion approach surpasses the current SOTA method CMI [11], delivering superior image quality with greater efficiency.


pseudo CIFAR-FS

pseudo MiniImageNet

pseudo Flower-VGG

pseudo CUB

Figure 9. T-SNE visualization of synthetic images. Our model inversion approach successfully inverts the essential discriminative features, which is beneficial to the following meta-learning.

tation by stacking three stages: pre-training, meta-training, and fine-tuning. We follow the original paper's setup and apply data augmentation to the support set of the target tasks. We select the best fine-tuning results from learning rates $[0.1, 0.01, 0.001]$.

- **Fine-tuning-free baselines.** "Nearest Neighbour (NN)" makes predictions based on the label of the closest class center. "CAML [13]" trains a sequence model to simulate the in-context learning of LLMs. Since we do not have real data to train the sequence model, we use synthetic data generated from pre-tuned LoRAs to train the sequence model. All other settings are consistent with the original paper.

E. More Discussions

Discussions on the inconsistent performance gains across various datasets. When we use LoRAs from the dataset the same as the testing dataset (in-domain setting), those LoRAs can provide domain-specific priors. This is particularly useful when the foundation model's pre-training dataset varies from the testing dataset. The main paper's Tab. 2 confirms this, showing a higher performance gain on CIFAR-FS (+10.01%) than other datasets (average +4.98%). The larger disparity between CIFAR-FS and the pre-training dataset is supported by the baseline NN in the main paper's Tab. 2, showing that directly transferring the foundation model to CIFAR-FS results in a lower accuracy (78.06%) compared to other testing datasets (average 85.31%). When we use LoRAs from datasets different from the testing dataset, performance gains across datasets are relatively stable, since these LoRAs offer limited useful domain-specific priors for all testing datasets.

Paradigms for Adaptable Foundation Models Several paradigms have been proposed to make large foundation models more adaptable. These paradigms involve combinations among Pre-training (P), Meta-learning (M), Fine-tuning (F) or PEFT, and In-context learning (I). Here, we provide a discussion over three paradigms, including P>F or P>PEFT, P>M>F and P>M>I. > indicates the sequence. Traditional P>F and P>PEFT [16, 42, 57] often fail to adapt foundation models to data-limited and real-time applications due to their need for sufficient data and explicit fine-tuning.

An emerging strategy, P>M>F, introduces a meta-learning phase before fine-tuning, preparing the pre-trained model for subsequent fine-tuning. This paradigm has shown promising results in vision [5, 27], language [1, 18, 24] and vision-language [34, 51, 79] domains.

More recently, the P>M>I paradigm has been proposed in language domains, aiming to acquire more advanced in-context learning ability of LLMs. For example, LLMs are equipped with the instruction-following ability by meta-training on a broad range of tasks accompanied by instructions [8, 35]. MetaICL [50] and ICT [7] explicitly meta-train LLMs to learn to learn in context. However, paradigms for tuning-free adaptation in VFMs are less explored, hindered by their inherent in-context learning limitations.

Difference between domain generalization and our setting. Our setting is fundamentally different from domain generalization [65] in several key aspects: Domain generalization aims to learn across multiple domains to generalize to an unseen domain. It requires that both known and unseen domains share the same label space. For example, training domain 1 may include real images of cats and dogs, and training domain 2 may include animated images of cats and dogs. Then, the test domain would include paintings of cats and dogs. Our setting is more challenging, as both the labels and domains for training and testing tasks differ.

Moreover, the labels in the test tasks are unseen during training. For example, in our setting, Task 1 might involve real images of cats and dogs, Task 2 might involve animated images of tigers and lions, and the test task could involve paintings of chairs and tables.

Additionally, unlike domain generalization, our setting emphasizes a few-shot scenario in test tasks and does not require original data in training tasks.

Discussions on data-free knowledge distillation (DFKD). Data-Free Knowledge Distillation (DFKD) [12, 28, 47, 55, 59, 60, 66, 76, 82] facilitates the transfer of knowledge from a large pre-trained teacher model to a smaller, more efficient student model without requiring access to the original training data. This methodology is particularly significant in scenarios where privacy or ethical concerns limit data accessibility. DFKD approaches such as DeepInversion [80] and CMI [11] synthesize images by utilizing teacher model statistics and classification objectives, and the synthesized images are used to perform knowledge transfer. Recently, ABD [23] investigates potential security vulnerabilities in DFKD, with a focus on backdoor threats. DFKD techniques have also been applied to areas such as federated learning [49] and model quantization [43, 44].

Unlike DFKD, which primarily employs inverted data to distill knowledge from a single teacher model, our study introduces a meta-learning framework that harnesses inverted data across multiple teacher models. Moreover, instead of transferring task-specific knowledge, our framework aims to learn generalizable prior *meta-knowledge* [14], which can be rapidly adapted to new tasks. Lastly, we propose a double-efficient mechanism that accelerates both the data inversion and meta-training processes. This contribution not only speed up our framework but also holds potential for improving the efficiency in standard DFKD methods.

F. More algorithms

Meta-training stage: LoRA Recycle. We summarize our proposed LoRA Recycle in Alg. 1.

Meta-testing stage: Tuning-free adaptation with meta-LoRA (Alg. 2). After meta-training, we obtain meta-trained meta-LoRA δW^*. The testing task $\mathcal{T} = (\mathcal{D}_\mathrm{s}, \mathcal{D}_\mathrm{q})$ consists of one support set and one query set. The support set is used as "context examples" to adapt the VFM f to the specific task, while the query set is what we actually predict. To predict the label of each query example $\mathbf{X}_\mathrm{q} \in \mathcal{D}_\mathrm{q}$, we first equip the VFM f with the meta-trained δW^* to obtain the enhanced VFM $f_{\delta W^*}$. Then we feed forward the support set $\mathcal{D}_\mathrm{s}$ and the query example $\mathbf{X}_\mathrm{q}$ into $f_{\delta W^*}$. We directly output $p(\mathbf{Y}_\mathrm{pred} = i | \mathbf{X}_\mathrm{q}, \mathcal{D}_\mathrm{s})$, the probability of $\mathbf{X}_\mathrm{q}$ being classified to label i via Eq. (3b) without fine-tuning. We assign the label with the max probability as the prediction result.

Algorithm 2: Tuning-Free Adaptation with Meta-LoRA

```
INPUT The VFM $f$. The meta-trained meta-LoRA $\delta W^*$.
    The meta-testing task with one support set $\mathcal{D}_\mathrm{s}$ and one
    unlabelled query set $\mathcal{D}_\mathrm{q}$.
OUTPUT The prediction results on the query set $\mathcal{D}_\mathrm{q}$.
Equip $f$ with the meta-lora $\delta W^*$
Make predictions on $\mathcal{D}_\mathrm{q}$ based on $\mathcal{D}_\mathrm{s}$ (Eq. (3b))
```