Supplementary Material for "ImagineFSL: Self-Supervised Pretraining Matters on Imagined Base Set for VLM-based Few-shot Learning"

Haoyuan Yang [1,*] Xiaoou Li [2,*] Jiaming Lv [1] Xianjun Cheng [1] Qilong Wang [3] Peihua Li [1 †]

[1] Dalian University of Technology [2] Beijing University of Posts and Telecommunications [3] Tianjin University

https://peihuali.org/ImagineFSL

A. Implementation Details on Pretraining and Fine-tuning

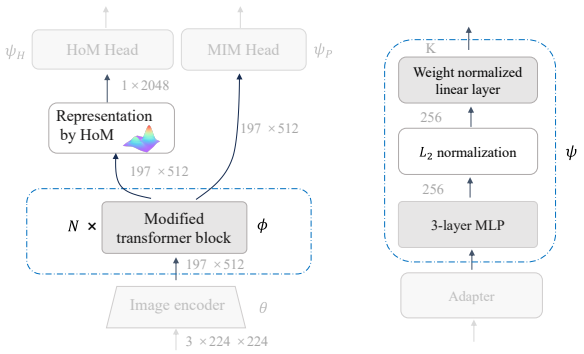
A.1. Architectures of Adapter and Heads

Figure S-1 illustrates the architectures of the adapters and projection head. The adapter is attached to CLIP's image encoder (Figure S-1a). With ViT-B/16 as the image encoder, we obtain 512-dimensional (dim) tokens. The adapter is a lightweight network comprising $N = 4$ *modified transformer blocks* (TransBlocks). The multiheaded self-attention (MSA) layer has 6 attention heads, each processing 64-dim queries, keys, and values, producing a concatenated 384-dim attention output, which is projected back to 512-dim via a linear layer. The feedforward network (FFN) following MSA has a hidden layer size of 1024, set to 2x the input dimension instead of the standard 4x increase. These modifications make our TransBlock parameter-efficient (1.8M parameters).

After the adapter, two parallel heads exist: the HoM head for higher-order moment-based image representation and the MIM head for masked image modeling. Both share the same architecture but have different parameters, as shown in Figure S-1b. Each consists of a 3-layer MLP with dimensions $input\ dim \rightarrow 1024 \rightarrow 1024 \rightarrow 256$. GELU activations are applied after the first two layers. The third layer outputs 256-dim features, which are L_2-normalized and passed through a weight-normalized linear layer. The $256 \times K$ weight matrix of this layer represents K L_2-normalized 256-dim prototypes, where $K = 4096$. The outputs are scaled by a temperature (temp) parameter and fed into the softmax function to produce probability distributions over the prototypes. The HoM head has 4.5M parameters, while the MIM head has 2.9M.

A.2. Hyperparameters of Pretraining

We mainly follow the training setup of DINOv2 [S-2], and implement our HoM-DINO based on its code repository.



(a) Adapter. (b) Projection head.

Figure S-1. Illustration of adapter (a) and projection head (b) with ViT-B/16 as the image encoder.

Hyperparameter	Value
Epoch	100
Batch size	200
Learning Rate (LR)	2e-5 → 5e-6
LR schedule	Cosine decay
Optimizer	AdamW
Weight decay	0.04 → 0.4
Stochastic depth	0.1
Gradient clipping	3.0
Teacher EMA momentum	0.996 → 1.0
Teacher temp	0.04 → 0.07
Teacher temp warmup epoch	30
Student temp	0.1
MIM *image* mask prob	0.5
MIM *patch* mask ratio	$\mathcal{U}(0.1,\ 0.5)$
HoM loss weight	1.0
MIM loss weight	1.0
Augmentation	As in DINO [S-1]

Table S-1. Hyperparameters of *pretraining*.

As in DINO, we adopt centering and sharpening techniques to prevent model collapse.

We summarize the hyperparameters in Table S-1. We train the models for 100 epochs using a batch size of 200. The learning rate starts at 2e-5 and gradually reduces to 5e-

*These authors contributed equally to this work. †Corresponding Author. The work was supported by National Natural Science Foundation of China (62471083, 62276186, 61971086).

6 using a cosine decay schedule. We use the AdamW optimizer with a weight decay following a cosine schedule from 0.04 to 0.4. We apply stochastic depth with a rate of 0.1 and gradient clipping at a threshold of 3.0. The student's temperature is set to 0.1 while we use a linear warm-up for the teacher's temperature from 0.04 to 0.07 during the first 30 epochs. The teacher is updated with an exponential moving average (EMA) of the student, with a momentum value [0.996, 1.0] following a cosine schedule. For masked image modeling (MIM), we mask the images in a batch with a probability (prob) of 0.5 and the mask ratio for the patch tokens is sampled uniformly in $[0.1, 0.5]$. The weights of HoM loss and MIM loss are both set to 1.0. We adopt the same data augmentation strategy as DINO. We do not use local crops as we find they bring no further improvement.

A.3. Hyperparameters of Fine-tuning

As in IsSynth [S-3], DISEF [S-4] and DataDream [S-5], we employ a mixed training strategy that simultaneously uses real and synthetic (synt) images within each batch. Our objective function comprises two cross-entropy (CE) losses, i.e., the loss of V-Classifier and loss of VL-Classifier, which are weighted equally. Each batch of size B contains $B/2$ real and $B/2$ synthetic images. Real and synthetic image losses are computed separately and combined with a weight.

The hyperparameters of fine-tuning for *few-shot tasks* are presented in Table S-2. For *ImagineFSL*, we use AdamW optimizer with a cosine decay of learning rate (LR), a weight decay of 1e-4 and a batch size of 128. We perform a grid search for epochs from $\{10, 20, \cdots, 80\}$ and for the adapter LR (denoted as *base LR*) from {1e-6, 1e-5, 1e-4, 1e-3} on the validation set of each task. The LR for the V-Classifier and VL-classifier is set to 10 times and 0.2 times base LR, respectively. For image augmentations, we follow implementation of DataDream. The setting of hyperparameters of *ImagineFSL$_{LoRA}$* is basically same as that of ImagineFSL, with some notable differences. The first difference is that it searches the rank and weight of LoRA among $\{16, 32, 64\}$ and $\{32, 64\}$, respectively; besides, the dropout for LoRA is set to 0.1 and VL-Classifier has the same LR with adapter. The second difference is that we fix the epoch to 10 for decreasing the cost, as *ImagineFSL$_{LoRA}$* is more expensive than ImagineFSL.

For *zero-shot recognition*, we perform fine-tuning exclusively on synthetic images, hence the loss only pertains to these images; other hyperparameters remain the same as those used in few-shot tasks. For *domain generalization*, we train with a 16-shot setting, with hyperparameters identical to those in few-shot tasks. For ImageNet-V2 and ImageNet-A, we use the corresponding synthetic images from SyntIN1K, while for ImageNet-S and ImageNet-R, we specially synthesize images for them.

Hyperparameter	**ImagineFSL**	**ImagineFSL$_{\text{LoRA}}$**
Epoch	$\{10, 20, \cdots, 80\}$	10
Batch size	128	128
Base LR	{1e-6, 1e-5, 1e-4, 1e-3}	{1e-6, 1e-5, 1e-4, 1e-3}
V-Classifier LR	Base LR $\times$ 10	Base LR $\times$ 10
VL-Classifier LR	Base LR $\times$ 0.2	Base LR $\times$ 1
LR schedule	Cosine decay	Cosine decay
Optimizer	AdamW	AdamW
Weight decay	1e-4	1e-4
Augmentation	As in DataDream [S-5]	As in DataDream [S-5]
V-Classifier loss weight	1.0	1.0
VL-Classifier loss weight	1.0	1.0
Real image loss weight	0.8	0.8
Synt image loss weight	0.2	0.2
LoRA rank	–	{16, 32, 64}
LoRA weight	–	{32, 64}
LoRA dropout	–	0.1

Table S-2. Hyperparameters of *fine-tuning*.

A.4. Implementations of Compared Methods in Pretraining

Competing Self-SL methods. For DINOv2 [S-2], we employ the same projection head as in our setup. We use centering techniques to normalize the teacher's outputs and observe stable training even without KeLeo regularization. We do not use local crops, as combination of them increases memory usage without improving performance in the few-shot setting. The hyperparameter configuration is consistent with the optimal settings used in our method.

For MAE [S-6], we design the decoder as a single TransBlock, which is followed by layer normalization plus a linear layer for pixel prediction. We strictly follow the hyperparameter setting of the original paper. Due to our limited computing resource, we use a batch size of 2048 and thus the LR is adjusted accordingly. We adopt the official code released by the original authors for implementations.

For SynCLR [S-7], we modify the 3-layer MLP projection head and the 2-layer MLP prediction head to reduce the number of parameters. Specifically, we decrease the hidden dimension of both MLPs from 4096 to 2048. Additionally, we do not use local crops during training. The projection head for MIM loss remains consistent with our method. We conduct experiment with the official code, following original hyperparameter settings, except for using a batch size of 200 with a correspondingly reduced LR.

Competing distribution modeling methods. Table S-3 summarizes image representations of different methods. *G^2DeNet* [S-8] assumes deep features follow Gaussian distribution. According to practice of the original paper, we add a linear layer after the adapter, decreasing the dimension of output tokens to 128. For all tokens including [CLS] token and patch tokens, we compute their mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Then, we obtain the Gaussian

Image Repr	Dim	Expression	Avg Acc
DINOv2	512	[CLS]	63.3/77.2
G^2DeNet [S-8]	8385	$\begin{bmatrix} \boldsymbol{\Sigma}+\boldsymbol{\mu}\boldsymbol{\mu}^T & \boldsymbol{\mu} \\ \boldsymbol{\mu}^T & 1 \end{bmatrix}^{1/2}$	64.0/77.3
MP [S-10]	2087	$[\text{[CLS]}; \text{MHC}^3(\mathbf{F})]$	63.0/76.5
HoM	2048	$[\text{[CLS]}; \mathbf{m}_1; \mathbf{m}_2; \mathbf{m}_3]$	**64.5**/**77.6**

Table S-3. Comparison of image representation (repr).

$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as the image representation, which is further embedded into the space of symmetric positive define matrices: $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \mapsto \mathbf{G} = \begin{bmatrix} \boldsymbol{\Sigma}+\boldsymbol{\mu}\boldsymbol{\mu}^T & \boldsymbol{\mu} \\ \boldsymbol{\mu}^T & 1 \end{bmatrix}^{1/2}$, where the superscript denotes the matrix square root that functions as a structural normalization. Finally, we stack the upper-triangular entries of $\mathbf{G}$ into a vector. As the image representation of G^2DeNet is large, we redesign the projection head with dimensions $8385 \rightarrow 512 \rightarrow 1024 \rightarrow 256$ to achieve a reasonable parameter count relative to the adapter. For the implementation of the matrix square root, we employ the fast iterative algorithm proposed in [S-9].

MP [S-10] introduces a module called multi-head convolutional cross-covariance (MHC^3) to capture second-order statistics. To reduce the size of MP representations, we split tokens into eight heads and compute the cross-covariance (XCov) matrix between the adjacent heads. This results in seven XCov matrices, which are stacked and passed to two consecutive 3×3 convolutions with stride 2, interleaved by GELU. The final image representation is a 2087-dim vector of the form $\left[\text{[CLS]}; \text{MHC}^3(\mathbf{F})\right]$ where $\mathbf{F}$ denotes the matrix of all patch tokens. Notably, unlike matrix square root for the covariance matrices [S-9], MP applies no structural normalization to the cross-covariance matrices. Our implementation of MP is based on the authors' code.

B. Implementation Details on Synthesizing both Captions and Images

We first describe our philosophy on design of the factors and the Prompt Template (PTe) for GPT-4 to analyze them. Then we introduce four patterns that constitute image captions and how we instruct GPT-4 to produce exemplary captions, followed by the PTe for Llama to generate extensive captions. We finally summarize all the synthetic datasets.

B.1. Analyzing Factors via GPT-4

We identify five factors that play an important role in constituting image captions, i.e., attribute, viewpoint, background (BG), lighting condition (LC) and cause of degradation of images (CD). The factors are described briefly as follows.

Attribute This factor pertains to the intrinsic characteristics of a specific concept (i.e., class). The objects, things, or scenes associated with this concept share common properties yet exhibit internal variations. These encourage generative models to focus on the characteristic traits unique to the concept while also accounting for differences, such as varied coat patterns within the same breed of cat. It assists LLMs in generating diverse captions for the given class.

Viewpoint Real-world photos are often taken from different angles (e.g., front and side) and distances (e.g., long shot and close-up). However, the T2I models may bias toward frontal, mid-distance views unless explicitly instructed. Through analysis of viewpoints via GPT-4, we have a wide range of shooting angles and distances to better replicate real-life scenes.

Background (BG) We obtain typical environments in which the specific class often appear by querying GPT-4. This makes the scenes of the generated images more heterogeneous and realistic.

Light Condition (LC) To simulate real-world photo shooting conditions, we collect via GPT-4 a wide variety of lighting conditions, spanning changes in both position of light sources and times of day and weather. This helps avoid the default single lighting (usually bright and soft) of the T2I model.

Cause of Degradation (CD) We note that real-world photos sometimes suffer from poor quality due to reasons such as blurring or poor focus. So we specifically set up a factor to simulate this situation, without which T2I models often tend to generate high-quality images.

Among the five factors, the attribute that encompasses rich visual characteristics is customized for each distinct concept. The background (BG) that spans various environments is also customized. The remaining factors, i.e., viewpoint, LC and CD, are shared across concepts. We query GPT-4 to thoroughly analyze these factors. For example, for attribute, we ask GPT-4 as follows:

PTe for GPT-4 to analyze *factors*

Generate a list of 20 distinct visual attributes that distinguish this particular {concept} from the others while accommodating variations within the {concept} in everyday photographs. Present this list as a Python array, with each element being a concise phrase that describes a unique attribute.

B.2. Generate Exemplary Captions via GPT-4

Based on these factors, we design four patterns for systematic caption generation. Attribute and viewpoint are fundamental factors included in all patterns. The Base-pattern contains only these two factors, while the other three patterns add one additional factor each. We summarize them in Table S-4. Emphasizing attribute and viewpoint in captions is crucial for mitigating the possible bias of the T2I models and maximizing image diversity. We do not specify

	Attribute	Viewpoint	Background	Lighting condition	Causes of degradation
Base-pattern	✓	✓			
BG-pattern	✓	✓	✓		
LC-pattern	✓	✓		✓	
CD-pattern	✓	✓			✓

Table S-4. Four patterns to synthesize captions.

all five features in one caption, which avoid generation of lengthy and complex captions and meanwhile allow more randomness for the generative model.

For each concept, we prompt GPT-4 to generate ten exemplary captions per pattern. We select a specific instance for each factor provided to GPT-4. Take as an example the BG-pattern that contains three factors, i.e., attribute, viewpoint, and background, the prompt to GPT-4 is as follows:

PTe for GPT-4 to generate exemplary captions

Your task is to create diverse and contextually rich captions for {concept}, which will serve as prompts for text-to-image models such as stable diffusion to generate images. To achieve this, consider the essential factors that influence visual image generation as follows: attribute that visually distinguishes the {concept}, viewpoint of the camera to capture the scenario, and background where the {concept} is photographed. These factors are crucial not only for generating images representative of various categories but also for ensuring the synthesized images reflect common photographic practices in everyday life.
For the {concept}, you will be provided with specific forms for the factors, i.e., attribute: {attribute}, viewpoint: {viewpoint} and background: {background}. By integrating them, you will generate a caption of less than 36 words. Use concise, clear and straightforward language, avoiding extravagant embellishments and vague expressions.

B.3. Extensive Image Caption Generation via Llama

Due to expensive cost of accessing GPT-4 API, we deploy the lightweight Llama 3 8B for generating extensive captions. To this end, we design the template prompts for Llama, where exemplary captions generated by GPT-4 are provided to facilitate in-context learning. For a given concept, a pattern is first selected, and then the specific forms of factors associated with this pattern, alongside three contextually relevant examples, are also chosen randomly.

Below we take the BG-pattern as an example, presenting the PTe in the following. The generic placeholder {concept} will be replaced by a specific target concept, and the detailed information for attributes, viewpoints, and backgrounds will be inserted into the respective placeholders: {}, {}, and {} before the arrows →. The corresponding examples will be inserted into the placeholder {} after the arrow.

	Synth Dataset	Concepts	Base-pattern	BG-pattern	LC-pattern	CD-pattern
Imagined Base Set	SyntIN1.5K	1,500	0.45	0.40	0.10	0.05
	SyntIN1K	1,000				
Task-specific Set for FSL	Caltech	100	0.45	0.40	0.10	0.05
	Aircraft	100	0.45	0.40	0.10	0.05
	Cars	196	0.45	0.40	0.10	0.05
	Food	101	0.45	0.40	0.10	0.05
	Pets	37	0.45	0.40	0.10	0.05
	Flowers	102	0.45	0.40	0.10	0.05
	DTD	47	1.00	–	–	–
	EuroSAT	10	1.00	–	–	–
	SUN	397	0.85	–	0.10	0.05
	UCF101	101	0.45	0.40	0.10	0.05
Domain Generalization	ImageNet-V2	1,000	–	–	–	–
	ImageNet-S	1,000	0.70	0.30	–	–
	ImageNet-A	200	–	–	–	–
	ImageNet-R	200	0.70	0.30	–	–

Table S-5. Summary of *synthetic datasets*. The percentage of generated images per pattern is provided.

PTe for Llama to generate captions for *BG-pattern*

Your task is to generate an image caption for a {concept}, by considering the following factors: attribute, viewpoint, and background. The caption should be suitable for use as a textual prompt for Stable Diffusion, ensuring that the generated image resembles a real-life photo. Use the three examples provided below to guide the generation of the caption:
{}, {} and {} → {}
{}, {} and {} → {}
{}, {} and {} → {}
{}, {} and {} →

B.4. Summary of Synthetic Images

In Table S-5, we summarize the synthetic datasets, providing the percentage of synthetic images per pattern. We build two imagined base sets, i.e., SyntIN1K where the concepts are the 1,000 category names of ImageNet-1K, and SyntIN1.5K where the concepts contain these 1,000 categories plus additional 500 class names randomly selected from ImageNet-21K. Besides, we synthesize task-specific datasets for 11 downstream datasets. For domain generalization, synthesizing images is a little different and is described below.

ImageNet-S. The sketch images capture the essential lines, shapes and contours of an object or scene, rather than details such as colors or textures. Therefore, we define the attribute as the type of sketches shared by all classes, which

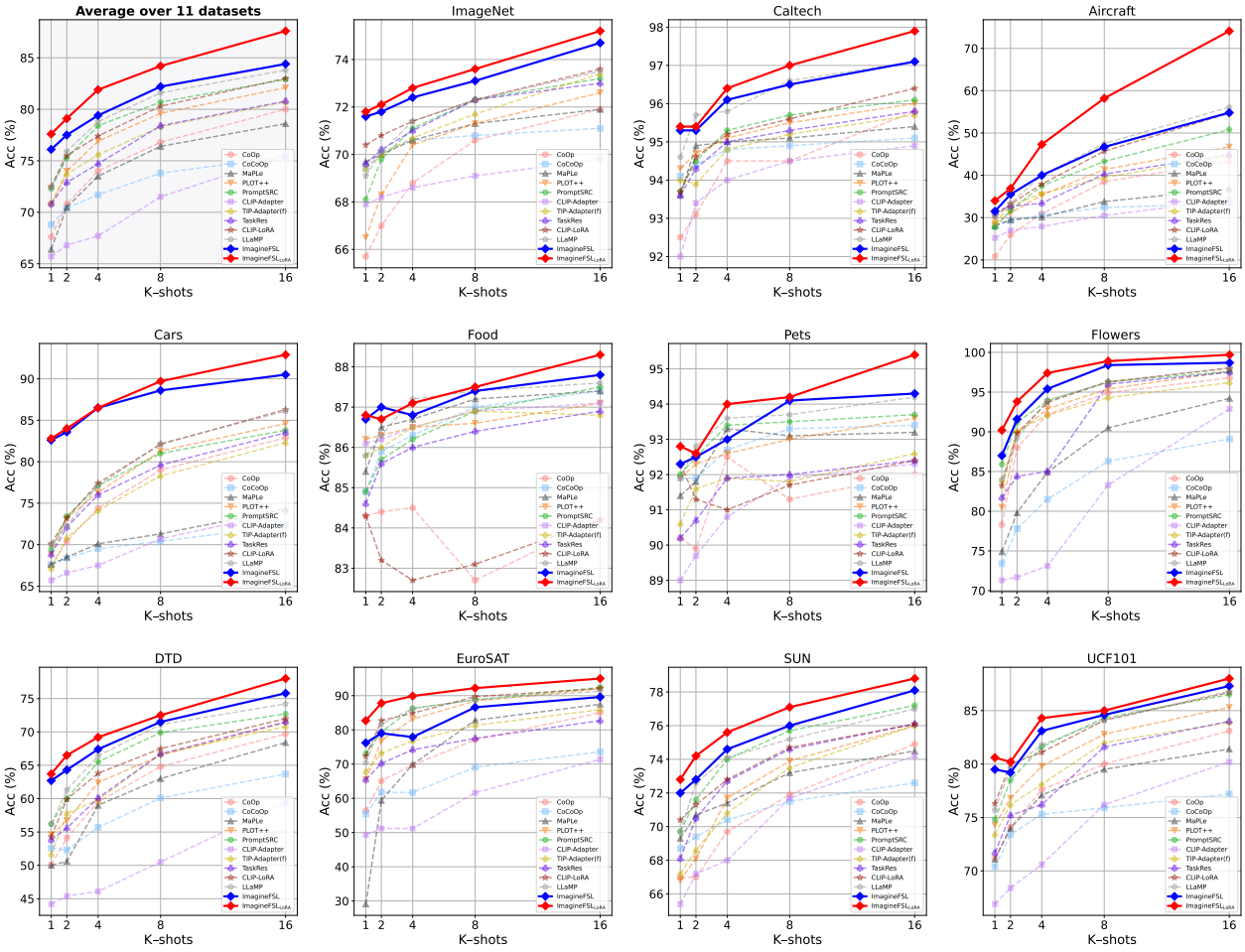


Figure S-2. Comparison to prior methods on individual 11 datasets. *This figure complements Table 2 in the main paper.*

is called style-attribute. Besides, we use simple handcrafted textual prompts and generate images for only two patterns, i.e., the Base-pattern and BG-pattern. Specifically, the textual prompt to SD3 for the Base-pattern is "Black and white {style-attribute}, a photo of a {concept}", while for the BG-pattern the prompt is "Black and white {style-attribute}, a photo of a {concept} with white background". To analyze the type of sketches, we ask GPT-4 as follows:

Query for GPT-4 to analyze *types of sketches*

Provide a list of 10 distinct styles of sketch that could appear in everyday photographs. Present this list as a Python array, with each element being a concise phrase that describes a unique type of sketches.

ImageNet-R. Hendrycks et al. [S-11] have previously identified 15 artistic renditions. However, these renditions are coarse-grained. We further ask GPT-4 to refine them, obtaining fine-grained renditions to enhance the diversity, e.g., "anime cartoon" is a refined rendition for "cartoon".

PTe for GPT-4 to analyze *rendition*

For {rendition} images, provide 3 refined artistic styles of {rendition} with the order of popularity in everyday life. Present your answer as a Python array with each element in a format like "style {rendition}" that describes a unique type of {rendition}.

For ImageNet-R, we use only two patterns for image generation, analogous to ImageNet-S. The prompt for the Base-pattern is straightforward: "{rendition}, a photo of a {concept}." For the BG-pattern, we employ the prompt "{rendition}, a photo of a {concept} with {background}," where the background is concept-specific and retrieved by GPT-4 as described in Section B.1.

C. Detailed K-shot Recognition Results on 11 Datasets

In *Table 2 of the main paper*, we compare the average accuracies across 11 datasets for each K-shot setting. Detailed results for individual datasets are presented in Figure S-2,

where the results of PromptSRC [S-12] and LLaMP [S-13] are from their respective papers, while the results of other methods are from CLIP-LoRA [S-14]. We observe that ImagineFSL consistently outperforms the competing methods across all shot settings, as evidenced by the average accuracies over the 11 datasets. Additionally, it either surpasses or matches the second-best method on individual datasets, except for EuroSAT. Notably, ImagineFSL$_{\text{LoRA}}$ exhibits significant improvements both in terms of average performance across the 11 datasets and in individual dataset performance.

D. Additional Experiments

In this section, we conduct additional experiments to evaluate the proposed methods.

D.1. Base-to-New Generalization

We adopt the evaluation protocol from CoOpOp [S-15] and compare ImagineFSL$_{\text{LoRA}}$ with previous methods on the base-to-new generalization task. Specifically, following DISEF [S-4], we fine-tune the pretrained model using 16-shot real images along with all synthetic images from the base categories. The fine-tuning hyperparameters are consistent with those described in Section A.3. The model performance is evaluated on both base and new categories, and the harmonic mean (HM) is computed as an overall performance metric. The results are presented in Table S-6.

First, our method achieves the best performance across all 11 datasets on the base categories, outperforming PromptSRC by an average margin of 2.9%. Second, our method achieves top performance on 8 out of 11 datasets on the new categories. Notably, although both methods utilize synthetic images, ImagineFSL$_{\text{LoRA}}$ consistently surpasses DISEF across all datasets on the new categories, clearly demonstrating its superior generalization capability. Finally, our method achieves best results in terms of HM on 9 out of 11 datasets, outperforming PromptSRC by an average margin of 1.0%.

Dataset		CLIP [S-16]	CoOp [S-17]	PromptSRC [S-12]	DISEF* [S-4]	ImagineFSL$_{\text{LoRA}}$*
ImageNet	Base	72.4	76.5	77.6	78.3	**79.4**
	New	68.1	67.9	70.7	71.0	**71.3**
	HM	70.2	71.9	74.0	74.5	**75.1**
Caltech	Base	96.8	98.0	98.1	98.5	**98.7**
	New	94.0	89.8	94.0	93.9	**94.1**
	HM	95.4	93.7	96.0	96.1	**96.3**
Aircraft	Base	27.2	40.4	42.7	55.9	**56.7**
	New	36.3	22.3	37.9	34.3	**38.0**
	HM	31.1	28.7	40.2	42.5	**45.5**
Cars	Base	63.4	78.1	78.3	84.1	**85.5**
	New	74.9	60.4	**75.0**	68.8	72.2
	HM	68.7	68.1	76.6	75.7	**78.3**
Food	Base	90.1	88.3	**90.7**	90.6	**90.7**
	New	91.2	82.3	91.5	91.5	**91.6**
	HM	90.6	85.2	**91.1**	91.0	**91.1**
Pets	Base	91.2	93.7	95.3	**96.4**	**96.4**
	New	97.3	95.3	97.3	97.7	**97.8**
	HM	94.1	94.5	96.3	97.0	**97.1**
Flowers	Base	72.1	97.6	98.1	98.6	**99.0**
	New	77.8	59.7	**76.5**	72.7	74.3
	HM	74.8	74.1	**86.0**	83.7	84.9
DTD	Base	53.2	79.4	83.4	**83.6**	**83.6**
	New	59.9	41.2	63.0	64.4	**64.5**
	HM	56.4	54.2	71.8	72.7	**72.8**
EuroSAT	Base	56.5	92.2	92.9	98.0	**98.6**
	New	64.1	54.7	73.9	72.9	**74.0**
	HM	60.0	68.7	82.3	83.6	**84.5**
SUN	Base	69.4	80.6	82.7	83.1	**83.6**
	New	75.4	65.9	78.5	78.2	**78.7**
	HM	72.3	72.5	80.5	80.6	**81.0**
UCF	Base	70.5	84.7	87.1	–	**87.4**
	New	77.5	56.1	**78.8**	–	76.8
	HM	73.8	76.5	**82.7**	–	81.8
Avg Acc	Base	69.3	82.7	84.3	–	**87.2**
	New	74.2	63.2	**76.1**	–	75.7
	HM	71.1	71.6	80.0	–	**81.0**

Table S-6. Comparison of *base-to-new generalization* to prior methods. **Bold**: best results; underlined: second-best results. *Using synthetic image.

D.2. Extra Ablation

Is adapter for text encoder helpful? Since we have synthesized both text and images, it is natural to consider using both modalities for CLIP adaptation. To this end, we attach a text adapter comprising three modified TransBlocks to the CLIP text encoder. This introduces an extra CLIP loss [S-16] to the Self-SL objective, which is consistent with the design of SLIP [S-24]. We perform pretraining and fine-tuning using the same settings described in Sections A.2 and A.3. Table S-7a presents the results. We observe that incorporating the text adapter leads to a performance decline, particularly on ImageNet, and the average accuracy drops by 0.8% in both 1-shot and 16-shot settings. A similar phenomenon has also been observed in previous work [S-21]. Understanding why this occurs and how to effectively leverage synthetic texts alongside synthetic images for CLIP adaptation remains a challenging problem for future research.

Effect of TransBlock count of image adapter. In Table S-7b, we assess the impact of varying the number of TransBlocks in the visual adapter. Increasing the number of TransBlocks from 2 to 4 enhances accuracy across most datasets, resulting in an approximate 0.5% increase in average accuracy for both the 1-shot and 16-shot settings. However, further increasing to 6 TransBlocks does not yield additional gains, suggesting a saturation in performance. Therefore, we opt to use 4 TransBlocks in our visual adapter throughout the experiments in this paper.

Adapter	ImageNet	Aircraft	Flowers	EuroSAT	Avg Acc
Vision	**71.6**/**74.7**	**31.5**/**54.8**	87.0/**98.7**	76.2/**89.6**	**66.6**/**79.5**
Vision+Text	68.4/72.7	30.5/54.7	**87.7**/98.3	**76.4**/89.2	65.8/78.7

(a) Effect of text adapter.

Blocks	Params	ImageNet	Aircraft	Flowers	EuroSAT	Avg Acc
2	3.67M	71.5/74.6	30.1/54.0	**87.0**/98.6	75.7/89.0	66.1/79.1
4	7.34M	**71.6**/**74.7**	**31.5**/54.8	**87.0**/98.7	**76.2**/**89.6**	**66.6**/79.5
6	11.01M	**71.6**/74.6	31.2/**55.8**	86.6/**98.8**	76.0/89.1	66.4/**79.6**

(b) Number of blocks versus performance.

Normalization	ImageNet	Aircraft	Flowers	EuroSAT	Avg Acc
✗	71.4/74.5	30.8/**55.3**	85.7/**98.8**	75.3/87.3	65.8/79.0
✓	**71.6**/**74.7**	**31.5**/54.8	**87.0**/98.7	**76.2**/**89.6**	**66.6**/**79.5**

(c) Impact of normalization for HoM.

Model	Method	ImageNet	Aircraft	Flowers	EuroSAT	Avg Acc
ViT-B/16	ImagineFSL	71.6/74.7	31.5/54.8	87.0/98.7	76.2/89.6	66.6/79.5
	ImagineFSL$_{\text{LoRA}}$	71.8/75.2	34.0/74.1	90.2/99.7	82.7/95.0	69.7/86.0
ViT-L/14	CoOp [S-17]	71.5/78.2	36.9/55.2	87.2/99.1	68.3/88.3	66.0/80.2
	MaPLe [S-18]	76.5/78.4	37.4/46.3	83.6/97.4	61.2/85.4	64.7/76.9
	ProGrad [S-19]	73.6/78.4	38.3/55.6	88.8/98.7	70.8/89.3	67.9/80.5
	PLOT++ [S-20]	73.7/78.6	35.2/44.1	85.0/98.8	72.4/92.2	66.6/78.4
	CLIP-Adapter [S-21]	74.6/76.4	32.9/46.4	79.5/97.3	60.1/75.8	61.8/74.0
	Tip-Adapter(f) [S-22]	76.4/79.3	38.5/55.8	90.9/98.3	67.8/86.1	68.4/79.9
	TaskRes [S-23]	76.2/78.1	39.7/55.0	87.6/97.8	70.6/84.3	68.5/78.8
	ImagineFSL	78.4/80.7	42.1/68.1	90.0/99.5	80.6/92.7	72.8/85.3
	CLIP-LoRA [S-14]	76.7/79.6	41.2/66.2	91.2/99.0	73.7/93.1	70.7/84.5
	ImagineFSL$_{\text{LoRA}}$	**78.6**/**80.9**	**44.9**/**79.0**	**94.8**/**99.7**	**85.6**/**96.0**	**76.0**/**88.9**

(d) Scaling of ImagineFSL and ImagineFSL$_{\text{LoRA}}$ model capacity.

Table S-7. *Additional* ablation and comparison in the 1-shot/16-shot settings. **Bold**: best results; underlined: second-best results.

How normalization affects HoM? In our HoM method, we apply square root and cubic root transformations to the second and third moments, respectively, to normalize them. To evaluate the impact of this normalization, we compare the performance with and without it. As shown in Table S-7c, omitting normalization results in a performance drop of 0.8% in 1-shot accuracy and 0.5% in 16-shot accuracy on average. This suggests that normalization plays a significant role in enhancing the effectiveness of our HoM approach, which is consistent with previous observation [S-9].

Further analysis on scaling model capacity. In *Table 6f of the main paper*, we study the scaling capability of ImagineFSL$_{\text{LoRA}}$ from ViT-B/16 to ViT-L/14. Here, we report additional results for ImagineFSL and compare to previous methods in Table S-7d. The results of the competing methods are from CLIP-LoRA [S-14]. We first note that scaling from ViT-B/16 to ViT-L/14 leads to significant performance improvements for our methods. Specifically, ImagineFSL achieves gains of 6.2% and 5.8% in the 1-shot and 16-shot settings, respectively, while ImagineFSL$_{\text{LoRA}}$ shows gains of 6.3% and 2.9%. These results suggest that our methods exhibit superior scaling capabilities as the capacity of CLIP models increases. Furthermore, ImagineFSL ranks as the runner-up across the board in terms of average accuracy, performing better than the third-place method on most individual datasets; ImagineFSL$_{\text{LoRA}}$ outperforms all competitors by substantial margins, in terms of both average accuracy and accuracy on individual datasets.

D.3. Complementarity to Existing Methods

To assess the generalization ability of our methodology, we apply it to several state-of-the-art CLIP adaptation methods: CoOp [S-17] (PT), CLIP-Adapter [S-21] (AT), and DISEF [S-4] (ET). We pre-train these methods on SyntIN1K and then fine-tune them using both task-specific synthetic data and few-shot real images. The results are summarized in Table S-8. By integrating our methodology, CoOp improves its performance by 5.1%/1.8% while CLIP-Adapter improves by 6.2%/6.7% in 1-shot/16-shot settings, respectively. Although DISEF already leverages synthetic images to complement its few-shot data, our approach still provides an additional 0.6%/0.5% improvement. These results underscore the broad applicability of our methodology and its potential to enhance a variety of different methods.

Method	Our Methodology	ImageNet	Aircraft	Flowers	EuroSAT	Avg Acc
CoOp	✗	65.7/71.9	26.2/43.1	78.3/96.8	56.4/85.0	56.7/73.5
CoOp	✓	70.1/72.6	26.4/46.1	83.2/96.6	67.7/86.0	61.8/75.3
CLIP-Adapter	✗	67.9/69.8	25.2/34.2	71.3/92.9	49.3/71.4	53.4/67.1
CLIP-Adapter	✓	68.9/71.4	30.2/71.0	76.4/95.4	62.7/87.3	59.6/73.8
DISEF	✗	71.0/74.0	32.0/67.9	88.0/98.9	82.3/94.3	68.3/83.8
DISEF	✓	71.2/74.1	33.5/68.9	89.6/99.3	82.7/94.7	68.9/84.3

Table S-8. Our methodology enhances state-of-the-art methods.

D.4. Further Analysis on Synthesizing Captions and Images

Are synthetic images as effective as real images? In Table S-9a, we compare ImagineFSL pretrained on synthetic versus real images. Specifically, we construct base sets containing either 0.3M or 0.45M real images from ImageNet. First, with 0.3M images, pretraining on real images yields only marginal gains (0.1% and 0.7% for 1-shot and 16-shot) over synthetic images. Despite the domain gap, models pretrained on synthetic images achieve competitive performance. Second, performance saturates for both synthetic and real images when dataset size increases to 0.45M, indicating saturation isn't related to data diversity. Since Table S-7b shows this saturation also isn't due to adapter size, we hypothesize it results from backbone capacity (see *Table 6f of the main paper*). These comparisons suggest synthetic

Base set	Size	ImageNet	Aircraft	Flowers	EuroSAT	Avg Acc
SyntIN1K	0.1M	71.1/74.4	31.0/54.8	87.0/98.8	74.8/88.1	66.0/79.0
	0.3M	71.6/74.7	31.5/54.8	87.0/98.7	**76.2**/89.6	66.6/79.5
SyntIN1.5K	0.45M	71.7/74.7	**32.9**/54.7	**87.7**/98.6	75.2/89.7	**66.9**/79.4
ImageNet	0.3M	71.6/**74.8**	32.4/**56.6**	87.5/98.9	75.4/90.5	66.7/**80.2**
	0.45M	**72.1**/74.6	32.0/56.2	87.3/**99.0**	75.0/**90.6**	66.6/80.1

(a) Comparsion of pretraining with synthetic versus real images.

Method	LLMs	ImageNet	Aircraft	Flowers	EuroSAT	Avg Acc
ImagineFSL	GPT-4	71.6/74.7	31.5/54.8	87.0/98.7	76.2/89.6	66.6/79.5
	+Llama3 8B	71.2/74.4	31.9/56.3	86.4/98.6	75.4/89.6	66.3/79.7
ImagineFSL$_{\text{LoRA}}$	Llama3.1 405B	71.8/**75.2**	34.0/**74.1**	90.2/**99.7**	82.7/95.0	69.7/**86.0**
	+Llama3 8B	**71.9**/75.1	**34.2**/74.0	**90.3**/99.5	**83.1**/**95.2**	**69.9**/**86.0**

(b) Closed-source GPT-4 versus open-source Llama3.1 405B.

Table S-9. Further analysis on synthesizing captions and images.

images possess high fidelity and diversity, making them a viable alternative to real images.

Replacing GPT-4 with Open-Source Llama 3.1 405B. In our synthesizing pipeline, we use GPT-4 with CoT for key factor analysis and exemplar caption generation. To assess the generalization and adaptability of our pipeline, we replace GPT-4 with the open-source Llama 3.1 405B model [S-25] to regenerate captions and corresponding images. The results, summarized in Table S-9b, show that ImagineFSL and ImagineFSL$_{\text{LoRA}}$ trained on images synthesized using Llama 3.1 405B perform comparably to those trained with GPT-4, indicating that our pipeline is not limited to GPT-4. Furthermore, we anticipate that advancements in open-source LLMs, such as Llama, Grok [S-26], and DeepSeek [S-27], along with improvements in image-to-text models like SD, will further enhance our pipeline.

D.5. Visualization

Visualization with and without pretraining. In *Table 6a of the main paper*, we quantitatively compare the performance between the non-pretraining and pretraining settings. To understand how pretraining influences the models, we visualize the self-attention maps obtained under different settings. Following DINO [S-1], we extract features from the final layer of the adapter and compute the self-attention scores between the [CLS] token and the patch tokens. Figure S-3 presents these visualization results. We observe that, as we progress from non-pretraining to SL pretraining and then to Self-SL pretraining, the model increasingly focuses on the foreground objects, with reduced attention to background distractions. This visualization supports the conclusion made in the main paper, i.e., both SL pretraining and Self-SL pretraining are superior to non-pretraining. Crucially, the Self-SL method (i.e., HoM-DINO) outperforms SL, more effectively learning discriminative features transferable from synthetic to real image domains.

Distributions of features. As described in *Section 3.1 of the main paper*, we prefer using statistical moments over assuming a prior distribution (e.g., Gaussian) for modeling feature distributions. To illustrate the rationale behind this, we visualize the feature distributions of input images. Specifically, given an input image, we extract 512-dim features from the last TransBlock and compute histogram for each feature component (Comp). These histograms are then fitted using Kernel Density Estimation (KDE) with a Gaussian kernel and a bandwidth of 1.0 to provide a smooth estimate of the underlying distribution. Figure S-4 showcases the histograms and KDE curves for two images. We observe that the distributions are complex and varied, exhibiting unimodal or multimodal peaks. This complexity suggests that assuming specifically a prior distribution may be sub-optimal. Hence, we opt to use statistical moments for distribution modeling, which is flexible in representing diverse distributions and can be learned in an end-to-end way.

Synthetic texts & images. Here, we provide quantitative analyses of synthetic texts generated by LLMs and the corresponding images generated by SD3 across different patterns. These analyses show that our generation pipeline can generate diverse, contextually rich synthetic images.

We demonstrate this through an example for each of the four patterns. Figure S-5a shows examples for the Base-pattern. As seen from the top (resp. bottom) row in the left-most panel, for the class *ragdoll cat*, the attribute of fluffy ruff (resp. soft pink nose) and the viewpoint of profile view (resp. front view) are synthesized in the texts and are correctly interpreted by SD3. For the BG-pattern, as Figure S-5b (right-most panel) shows, the background of gas station (resp. country road) are reflected for the class *1998 Eagle Talon Hatchback*. For the LC-pattern, as seen in Figure S-5c (left-most panel), the lighting condition of foggy light (resp. sunset) are clearly visible for the class *campsite*. Lastly, for the CD-pattern, we can see from Figure S-5d (right-most panel) that the cause of degradation, i.e., incorrect focus (resp. poor lighting), is highlighted for the class *ceiling fan*. Our method can flexibly create concept-specific attributes and backgrounds, provide appropriate viewpoints, adjust lighting conditions, and introduce image degradation.

In addition, we showcase the synthetic images used for *domain generalization*. Figure S-6a presents the generated images for ImageNet-S. Examples are provided for six style of sketches, including cross-hatching, ink sketch, line art, cartoon sketch, pencil sketch, and charcoal sketch. For each style, we provide two generated images and the accompanying text prompts that are stacked vertically. The example images generated for ImageNet-R are provided in Figure S-6b. Examples of 12 rendition styles are shown, such as contemporary origami, plush toy, modern sculpture, origami

art installation. Notably, supported by these synthetic images, our methods establish new state-of-the-art accuracies on both ImageNet-S and ImageNet-R.

E. Limitations and Future Research

Compared to existing works, our methods introduce non-trivial computational overhead due to training on a large-scale synthetic base set. However, since this pretraining is task-agnostic and can be completed beforehand, it does not affect the efficiency of downstream tasks. Once pretrained, our models are universally applicable to few-shot recognition, domain generalization, zero-shot recognition, and base-to-new generalization tasks.

Although our work mainly focuses on CLIP, the core ideas and techniques–pretraining on purely synthetic images, distribution-based Self-SL, and a systematic, scalable synthesizing pipeline–are model-agnostic and thus readily applicable to other VLMs, such as CoCa [S-28]. Therefore, ImagineFSL could potentially benefit from more powerful visual encoders and enhanced cross-modal interactions. Investigating our methods across diverse VLM models is a promising direction for future research.

Additionally, although our synthesizing method generates caption-image pairs, our pretraining currently utilizes only the images, as initial attempts to incorporate synthetic captions provided no benefit. As discussed in Section D.2 and [S-3], effectively leveraging synthetic texts alongside synthetic images could potentially further enhance CLIP adaptation methods and is worth future investigation.

By using statistical moments as image representations, our HoM-DINO shows clear improvement over existing Self-SL methods in few-shot scenarios. Nevertheless, statistical moments cannot fully characterize feature distributions, which are complex and varied, as illustrated in Figure S-4. To our best knowledge, modeling the probability distributions of deep features remains an open problem. More advanced distribution modeling techniques could further enhance the representational capabilities of neural network models.

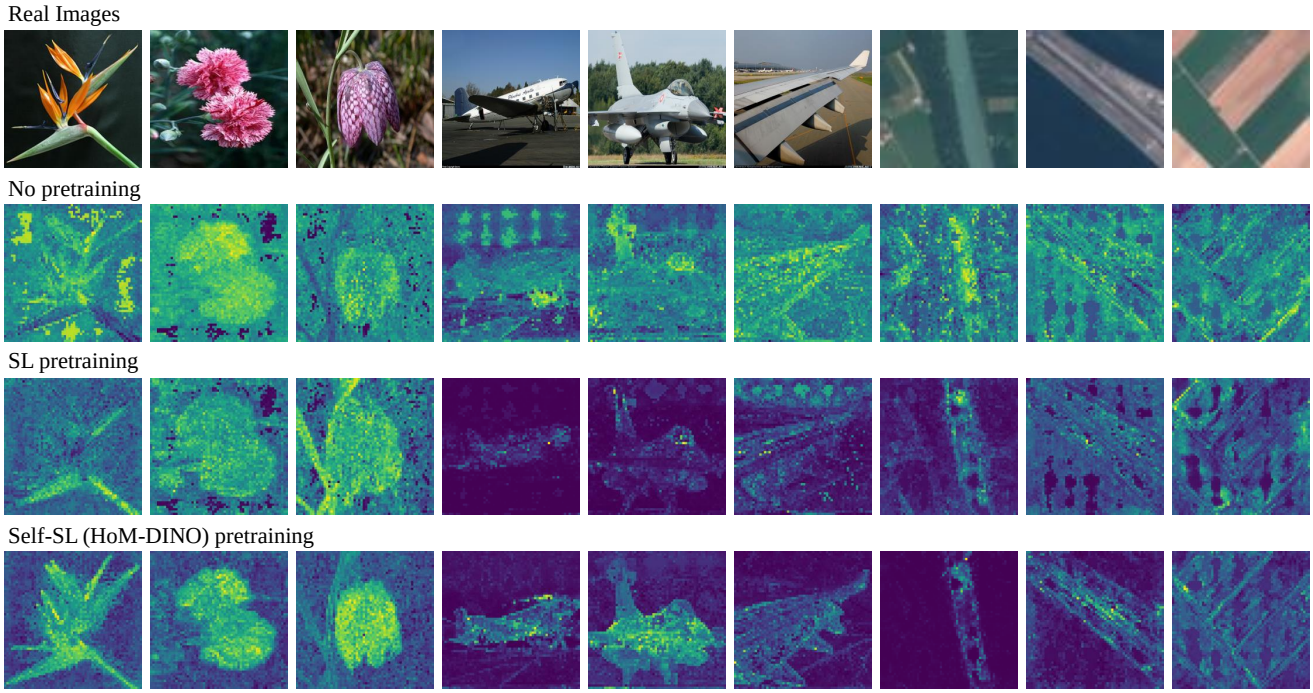


Figure S-3. Self-attention maps in the settings of no pretraining, SL pretraining and Self-SL pretraining.

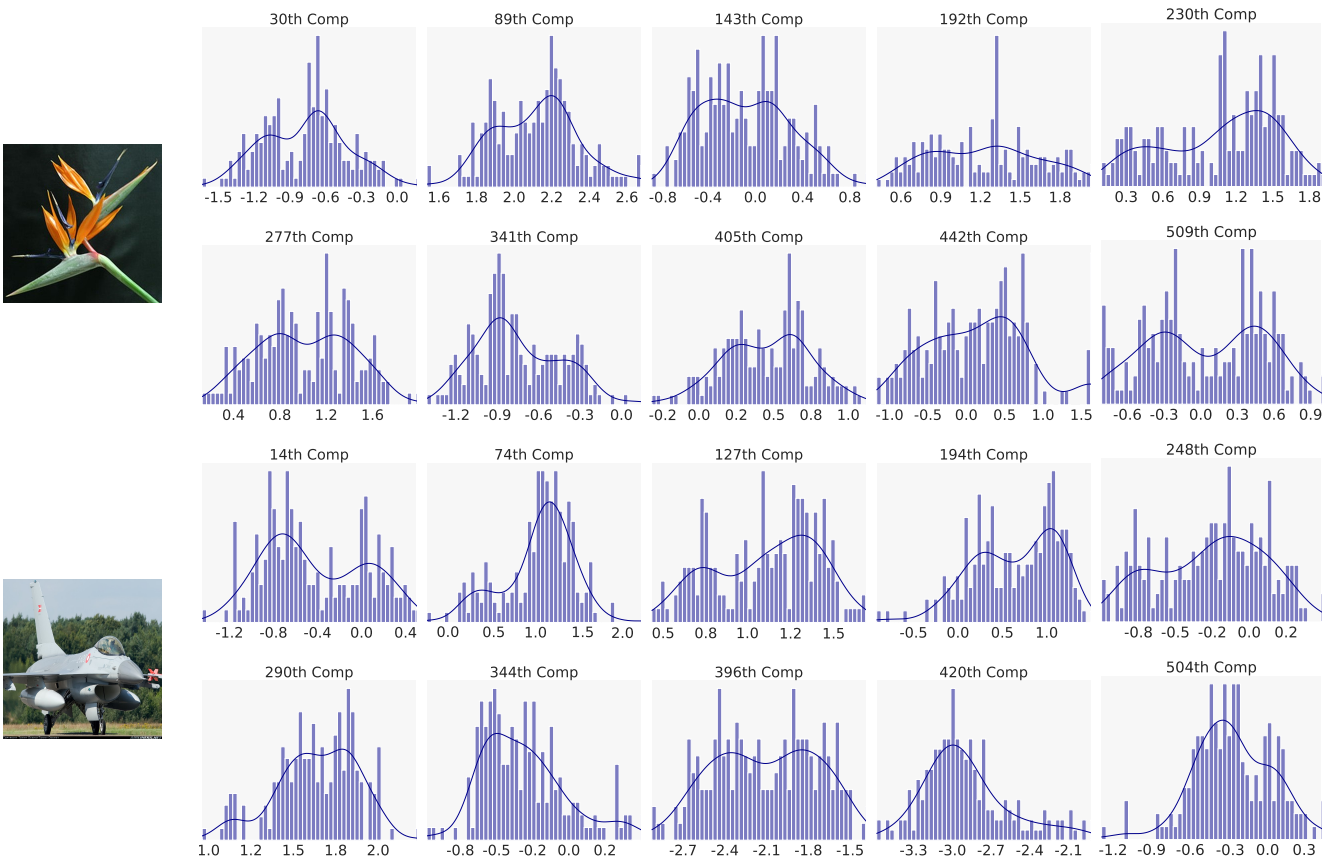
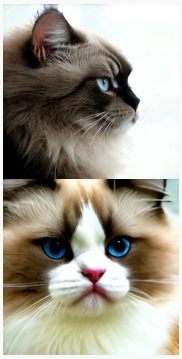


Figure S-4. For each of the two images (left), we plot histograms and KDE curves for 10 randomly selected feature **Comp**onents.

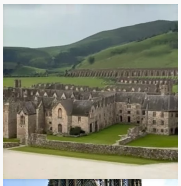


A *ragdoll cat* with a {fluffy ruff} around its neck, captured in a {profile view}, looking regal and adorable

A *ragdoll cat* with a {soft pink nose}, shot in a slightly elevated {front view}, showing its adorable face.

2012 BMW 1 Series Coupe {front view} featuring {angled headlights} with corona rings.

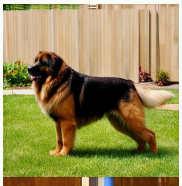
A *2012 BMW 1 Series Coupe* captured from a {rear view} angle, showcasing its {short overhangs} at rear.



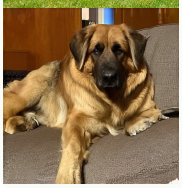
A {long shot} of an *abbey*'s {monastic cells}, with rows of simple yet elegant stone buildings, set amidst rolling hills and distant mountains.

A stunning *abbey* with its majestic transepts, captured in a {tilted view}, highlighting the intricate stone carvings and {stained glass windows}.

(a) **Base-pattern**.



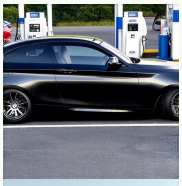
A *leonberger* with long, feathered legs stands in a {backyard}, photographed from a profile view.



Leonberger with a slightly arched neck, captured from a slightly elevated front view, resting on a {couch}.

A side view of *poutine* with dark, rich gravy, set against a {bustling street food stall}.

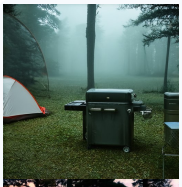
A front view of a plate of *poutine* with crispy golden fries, set on a {clean and modern kitchen counter}.



2012 BMW 1 Series Coupe with a wide, low stance, captured from the side at a busy {gas station}.

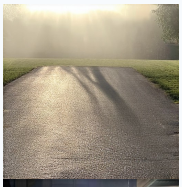
2012 BMW 1 Series Coupe with a wide, low stance, captured in a front view, driving down a winding {country road}.

(b) **BG-pattern**.



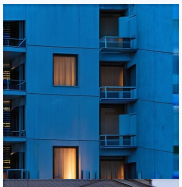
A *campsite* with outdoor cooking equipment in a three-quarter view, captured in {foggy light}.

A *campsite* with a folding table, captured in a panoramic view at {sunset}, surrounded by nature.

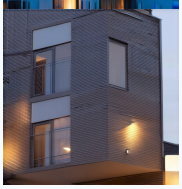


A *driveway* with no sidewalks, seen from the rear, in the {hazy sunlight}.

A *driveway* entrance, captured from an extreme long shot with {bright overhead lighting}.



A back view of an *apartment building* with multiple floors, as the {blue hour} casts a warm glow.

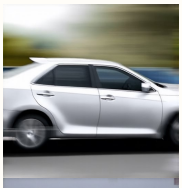


Oblique angle shot of an *apartment building* with external lighting fixtures in soft {evening light}.

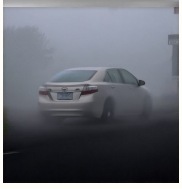
(c) **LC-pattern**.

A merry-go-round on a *playground*, captured in a three-quarter rear view, with visible {color distortion}.

Extreme long shot of a *playground* with playhouses, {low resolution} causing pixelation.



2012 Toyota Camry Sedan with chrome trim accent around window from side view with {motion blur}.



A *2012 Toyota Camry Sedan* with hatchback, shot in a three-quarter rear view, obscured by a heavy fog, showcasing the {poor weather condition}.

Close-up shot of the motor housing of a *ceiling fan*, captured with {incorrect focus}, resulting in a blurry image.

Close-up shot of a *ceiling fan*, captured in {poor lighting}, highlighting its blades and motor in a dark atmosphere.

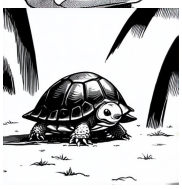
(d) **CD-pattern**.

Figure S-5. Example synthetic images for the four patterns. We highlight the *concept*, and the factors including attributes, viewpoints, backgrounds, lighting conditions and causes of degradation.

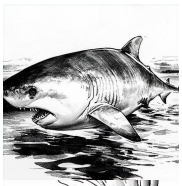
Black and white {cross-hatching}, a photo of a *brambling* with white background.

Black and white {cross-hatching}, a photo of a *hen* with white background.

Black and white {cartoon sketch}, a photo of a *backpack* with white background.



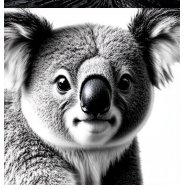
Black and white {cartoon sketch}, a photo of a *mud turtle.*



Black and white {ink sketch}, a photo of a *tiger shark.*

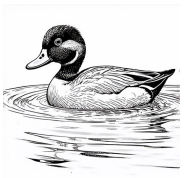
Black and white {ink sketch}, a photo of a *cockatoo* with white background.

Black and white {pencil sketch}, a photo of a *volcano* with white background.



Black and white {pencil sketch}, a photo of a *koala* with white background.

Black and white {line art}, a photo of a *Maltese* with white background.

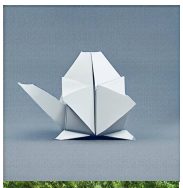


Black and white {line art}, a photo of a *duck.*

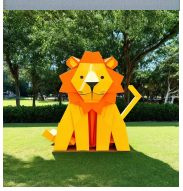
Black and white {charcoal sketch}, a photo of a small white *butterfly.*

Black and white {charcoal sketch}, a photo of a *palace.*

(a) **Generated sketch images**.



{Contemporary origami}, a photo of a *snail.*



{Origami art installation}, a photo of a *lion* with a park featuring trees.

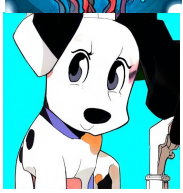
{Impressionism painting}, a photo of a *bell pepper.*

{Realism painting}, a photo of a *great white shark* with a school of fish.

{Plush toy}, a photo of of a *pineapple.*

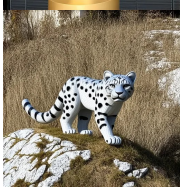
{Collectible Figurine}, a photo of a *pelican.*

{Hand-drawn cartoon}, a photo of a *jellyfish.*



{Anime cartoon}, a photo of a *dalmatian.*

{Modern sculpture}, a photo of a *harp.*



{Modernist sculpture}, a photo of a *snow leopard* with a rocky slope.

{Machine embroidery}, a photo of a *banana.*

{Morden embroidery}, a photo of a *lighthouse.*

(b) **Generated rendition images**.

Figure S-6. Example synthetic images for ImageNet-S (a) and ImageNet-R (b). The individual styles of rendition or sketch are highlighted.

References

[S-1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 1, 8

[S-2] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024:1–31, 2024. 1, 2

[S-3] 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? In *International Conference on Learning Representations*, 2023. 2, 9

[S-4] 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. 2, 6, 7

[S-5] Jae Myung Kim, Jessica Bader, Stephan Alaniz, Cordelia Schmid, and Zeynep Akata. DataDream: Few-shot guided dataset generation. In *European Conference on Computer Vision*, pages 252–268, 2024. 2

[S-6] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 159–168, 2022. 2

[S-7] Yonglong Tian, Lijie Fan, Kaifeng Chen, Dina Katabi, Dilip Krishnan, and Phillip Isola. Learning vision from models rivals learning vision from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15887–15898, 2024. 2

[S-8] Qilong Wang, Peihua Li, and Lei Zhang. G^2DeNet: Global Gaussian distribution embedding network and its application to visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6507–6516, 2017. 2, 3

[S-9] Peihua Li, Jiangtao Xie, Qilong Wang, and Zilin Gao. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recogniton*, pages 947–955, 2018. 3, 7

[S-10] Mingze Gao, Qilong Wang, Zhenyi Lin, Pengfei Zhu, Qinghua Hu, and Jingbo Zhou. Tuning pre-trained model via moment probing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11769–11779, 2023. 3

[S-11] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 5

[S-12] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15190–15200, 2023. 6

[S-13] Zhaoheng Zheng, Jingmin Wei, Xuefeng Hu, Haidong Zhu, and Ram Nevatia. Large language models are good prompt learners for low-shot image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28453–28462, 2024. 6

[S-14] 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*, pages 1593–1603, 2024. 6, 7

[S-15] 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*, pages 16816–16825, 2022. 6

[S-16] 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*, pages 8748–8763, 2021. 6

[S-17] 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, 2022. 6, 7

[S-18] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. MaPLe: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023. 7

[S-19] 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*, pages 15659–15669, 2023. 7

[S-20] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Prompt learning with optimal transport for vision-language models. In *International Conference on Learning Representations*, 2023. 7

[S-21] 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, 2023. 6, 7

[S-22] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-Adapter: Training-free adaption of CLIP for few-shot classification. In *European Conference on Computer Vision*, pages 493–510, 2022. 7

[S-23] Tao Yu, Zhihe Lu, Xin Jin, Zhibo Chen, and Xinchao Wang. Task residual for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10899–10909, 2023. 7

[S-24] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. SLIP: Self-supervision meets language-image pre-training. In *European conference on computer vision*, pages 529–544, 2022. 6

[S-25] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 8

[S-26] xAI. Grok 3 beta–the age of reasoning agents. https://x.ai/news/grok-3, 2025. Accessed: 2025-03-20. 8

[S-27] DeepSeek-AI, Wenfeng Liang, Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, and Huazuo Gao. DeepSeek-V3: A high-performance mixture-of-experts language model. *arXiv preprint arXiv:2412.19437*, 2024. 8

[S-28] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. 9