Leveraging Vision-Language Models for Resource Constrained Settings

Anonymous authors
Paper under double-blind review

Abstract

Vision-language models (VLMs) such as CLIP have emerged as extremely strong zero-shot and few-shot image classifiers. However, these models are often too expensive or cumbersome for resource constrained downstream applications. In this work, we examine how to best leverage the strength of pretrained VLMs: by extracting *task-specific* information in order to obtain a small model that can be deployed in a very specific and low-resource setting. We present the SIDCLIP method, a novel training pipeline which drastically improves the performance of small, efficient models, such as EfficientNet B0. The pipeline includes three components that are critical to obtaining strong performance: 1) augmenting the classifier with *synthetic data* generated by leveraging CLIP itself; 2) *initializing* the modeling process using a smaller CLIP model pretrained on the target architecture; and 3) incorporating *knowledge distillation* to maximally mimic the performance of the larger model. SIDCLIP improves the performance of an EfficientNet B0 model by an average of 50% on 1-shot versions of four datasets and by an average of 26% on the 8-shot versions, relative to directly trained networks, additionally approaching CLIP's linear probe performance while using a model with less than 2% of the parameters of CLIP ViT-L/14's image encoder. We hope our work can be useful as a practical guide for leveraging the power of foundation models in downstream data-scarce and budget constrained settings.

1 Introduction

Foundation models such as CLIP-based models have been shown to perform extremely well on zero-shot and few-shot image classification: via simple prompting and/or a few examples, these models can achieve classification performance on par with models trained with much more task-specific data (Radford et al., 2021). However, this performance comes at a cost: the models are extremely general and large-scale, and thus incur a high inference cost relative to smaller, more task-specific models, making them unsuitable for many edge applications. This challenge has led to a number of methods for compressing or distilling knowledge from large foundation models into smaller models. Although these techniques can preserve strong performance relative to the large foundation model, they are often not task-specific, and when they are, they often focus on preserving the model's zero-shot performance for new tasks, rather than taking advantage of limited task-specific downstream data (Popp et al., 2024; Li et al., 2023; Wu et al., 2023; Vasu et al., 2024; Sun et al., 2023).

In this work, our goal is to produce a small model that performs as close as possible to a powerful large-scale vision-language model (VLM) on a particular downstream task. We address the specific challenge of attempting to leverage the strong performance of zero- and few-shot CLIP image classification models into vastly more efficient (but task-specific) architectures. In other words, given a very limited amount of data on a desired downstream image classification task, and a very limited inference-time compute budget, we obtain the best performance on a downstream compact model by *leveraging the capabilities of larger models*. In practice, we find that three separate components are central to obtaining strong performance:

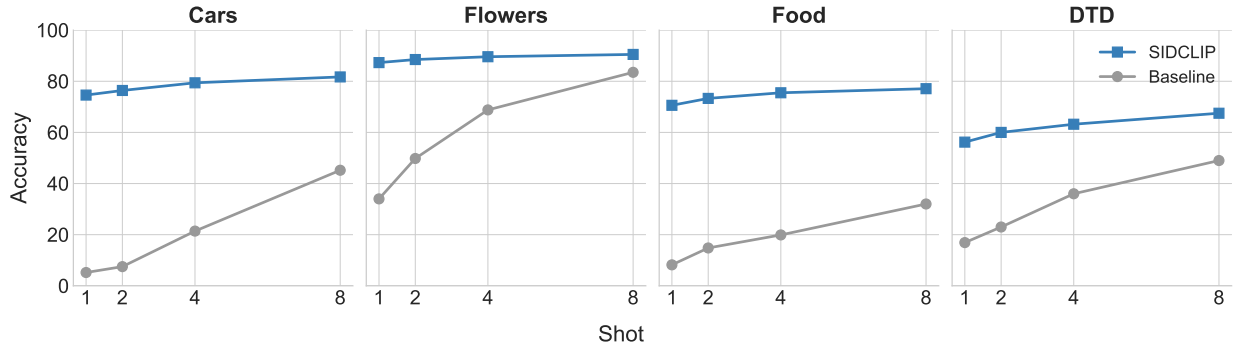

Figure 1: Using SIDCLIP to train an EfficientNet B0 model drastically outperfoms standard training. Each plot includes results from a different dataset. "Baseline" refers to an EfficientNet B0 finetuned on few-shot data under a standard training regimen. "SIDCLIP" demonstrates the notable improvement when training with our proposed method.

1. We augment the classifier with **synthetic data** generated by leveraging CLIP itself. Specifically, we use a text-to-image generative model seeded with embeddings produced from linear interpolations of the text of the class label *and* the CLIP image embeddings of the available data.

2. We **initialize** our small model as a variant based on a small CLIP model pretrained on the target architecture.

3. We incorporate **knowledge distillation** to maximally mimic the performance of the larger CLIP model.

We call our method, which incorporates the above three components, *SIDCLIP* (Synthesize-Initialize-Distill CLIP). While each of these elements alone have been the subject of exploration in the literature, we emphasize that the work here serves largely as a practical guide that demonstrates the relative value of leveraging these three capabilities, as well as ablations demonstrating the efficacy of subsets of these elements.

We evaluate our proposed approach, along with ablations and other baselines, on four common small-scale image classification benchmarks: the Stanford Cars (Krause et al., 2013), Oxford Flowers (Nilsback & Zisserman, 2008), Food 101 (Bossard et al., 2014), and Describable Textures (DTD) (Cimpoi et al., 2014) datasets. We improve the performance of small models by up to 65% across a range of models and datasets. SIDCLIP-trained models approach and even exceed the teacher's performance. A comparison of SIDCLIP training and standard training is shown in Figure 1. For a fixed student (EfficientNet B0), we show the difference in performance across datasets between standard training ("Baseline") and the best performing SIDCLIP variant.

2 Related Works

Synthetic data. There has been notable evidence to indicate that synthetic data is helpful in general when training models and particularly in distillation. Azizi et al. (2023) find that augmentation of a dataset with synthetic data improves image classification performance on CNN and ViT architectures. He et al. (2023) focus on the zero- and few-shot domains and reaches a similar conclusion: that synthetic data can be used in conjunction with real data to improve performance on image classification tasks. Similarly to our work, Popp et al. (2024) generate synthetic data in order to perform distillation. This work differs from ours in two notable ways: they assume *no* access to the downstream data, rather than a small number of samples; and the aim is to transfer the general zero-shot capabilities of CLIP rather than focusing on a particular downstream task. More generally, this area of data-free distillation explores the usage of only synthetic data (and no real data) during the distillation process (Chawla et al., 2021; Fang et al., 2022).

While these prior works all incorporate synthetic data, none utilize the particular image- and text-conditioning generation method that we use in SIDCLIP. The image generation pipeline we use was introduced in Razzhigaev et al. (2023) and achieves SOTA FID scores on generated images relative to other open source models.

Compression. VLMs have remarkable few- and zero-shot performance on downstream tasks and are strong image classifiers (Radford et al., 2021; Jia et al., 2021; Li et al., 2022; Yuan et al., 2021; Zhai et al., 2023). There has been a range of work exploring the natural next step of attempting to compress these high powered models into smaller versions that require less memory and have lower inference times. Some (such as pruning, quantization, and distillation) mirror compression in non-foundation models, while others (including parameter-efficient fine-tuning such as adapter layers or prompt tuning) are unique to the VLM or LLM setting (Hinton et al., 2015; Dettmers et al., 2022; Frantar & Alistarh, 2023; Sun et al., 2024; Houlsby et al., 2019; Liu et al., 2022; Lester et al., 2021; Jia et al., 2022).

In many of the existing efforts to compress foundation models, the goal has been to preserve the *general* capabilities of the models. Rather than focusing on a model's performance on a particular task, these methods aim to broadly preserve the VLM's generalization abilities for image classification (Li et al., 2023; Wu et al., 2023; Vasu et al., 2024; Sun et al., 2023; Wu et al., 2022; Cai et al., 2025).

TinyCLIP and MobileCLIP both preserve CLIP's general purpose knowledge through distillation (Wu et al., 2023; Vasu et al., 2024). TinyViT is another method which produces a small downstream model via distillation (Wu et al., 2022). Task-specificity is not part of the distillation process for any of these methods.

Similarly to our CLIP-initialized small model, Sun et al. (2023), distill from CLIP ViT-L/14 to a smaller foundation model, and find that this distilled model outperforms a similar model trained from scratch. However, their smallest model (Swin-T) is over three times larger than our largest model and they examine only the zero-shot setting. The value of knowledge distillation for task-specific small model performance is also highlighted in Jang et al. (2025), but they do not explore few-shot settings.

Li et al. (2023) distills from a CLIP ViT-L/14 teacher to a convolutional network student such as ResNet18. They measure task-specific performance as out-of-distribution performance: they perform distillation without any of the task-specific samples and then evaluate the zero- or few-shot performance of their model on downstream tasks. While similar to our setting, this setting does not take advantage of task-specific data available during distillation and thus yields lower performance than our method.

Few-shot learning. While preserving the entirety of CLIP's performance is a worthwhile goal, it is not the correct focus for all settings. The few-shot setting, when there is limited downstream training data available, arises in situations where data collection is expensive or challenging (Wang et al., 2020). Training large-scale models from scratch is an extremely data-intensive process, so usage of few-shot data to finetune an existing model can increase accessibility and customization of the power of VLMs. While there is some work that addresses a few-shot downstream setting, these methods often preserve or augment the network architecture of large CLIP models, thus making these approaches less feasible solutions for resource-constrained users (Ma et al., 2024; Wortsman et al., 2022; Islam et al., 2021). If some downstream task-specific training data is available, these methods are not equipped to best utilize it.

3 The SIDCLIP Method

Motivation. To use CLIP as an image classifier, first an image is passed into the image encoder, and text of the possible classnames is passed into the text encoder. Then, the embedding similarity between the image and each possible classname is measured. Although this process yields high accuracy on a variety of downstream tasks, the CLIP model is unnecessarily large for many downstream applications, such as use on edge devices: one of the most commonly used CLIP models, CLIP ViT-L/14, has 307M parameters in its image encoder (Radford et al., 2021).

Here we ask: what if a user wants to take advantage of CLIP's strong off-the-shelf zero and few shot performance but does not need its full "general-purpose" abilities? They may only need to classify images

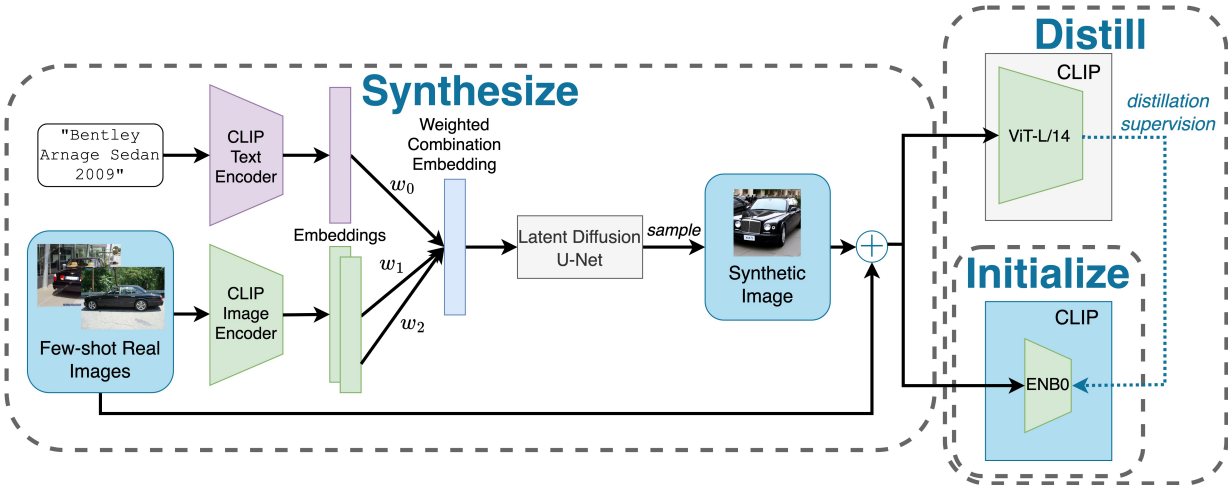

Figure 2: The three components of SIDCLIP: *synthesize* data via a weighted combination of class labels and real images; *initialize* the student as the image encoder of a small CLIP model; *distill* from a powerful teacher model.

corresponding to a specific task and cannot afford to run such a large model. In this case, it is desirable to transfer only a specific portion of CLIP's image classification capabilities to a smaller model.

With existing methods, a user would be able to produce a general-purpose small model, and potentially finetune it on the task of interest, but is left without being able to optimally take advantage of the limited training data they have. They would end up with a smaller version of CLIP, not a model tailored to their specific use case.

Problem setting. Our goal is to maximize the performance on a particular image classification task, subject to resource constraints. We have access to a small model $\mathcal{S}$ that fits our budget constraints and k labeled samples per class $c \in \mathcal{C}$, for $n = |\mathcal{C}|$ classes. We also have access to a large scale teacher VLM $\mathcal{T}$, such as a CLIP model.

Our method, SIDCLIP, consists of three essential components for leveraging CLIP's power in training a small model in a data-constrained setting. These three components are 1) synthetic data, 2) initializing the model as a small CLIP variant, and 3) distilling from CLIP to the small model. The pipeline is shown in Figure 2.

3.1 Component #1: Synthetic Data

We use synthetic data to augment the limited samples per class in a few shot setting. As described in our problem setting, we have k labeled samples per class. We use these $k \times n$ samples $\mathcal{D}_r$ and their classname labels $\mathcal{L}$ to generate additional synthetic samples that can be used for training the model. When operating in the k-shot setting, we *only* use those k samples and their classnames as input to generate additional synthetic data. The generative diffusion model we use accepts CLIP text and/or image embeddings as input and conditions its generation on these inputs. This generative process allows us to extract task-specific data from the teacher CLIP model.

More formally, when we want to generate a synthetic sample from class c, we use the label $l_c \in \mathcal{L}$ and some set of images $\{x_i\}_{i=1}^{I} \in \mathcal{D}_{r,c}$, for $I \leq k$ and where $\mathcal{D}_{r,c}$ refers to the real data available from class c. We obtain the CLIP image and text embeddings: $\texttt{img\_emb}(x_i)$ and $\texttt{text\_emb}(l_c)$ and combine them via a weighted combination:

$$\texttt{emb} = w_0 \cdot \texttt{text\_emb}(l_c) + \sum_{i=1}^{I} w_i \cdot \texttt{img\_emb}(x_i)$$

such that $\sum_{i=0}^{I} w_i = 1$. This combination is then passed into the generative model, which we sample from to obtain J synthetic samples $\mathcal{D}_{s(r)} = \{x'_j\}_{j=1}^{J} \sim \mathcal{G}(\texttt{emb})$.The subscript $s(r)$ emphasizes that the synthetic samples are generated using only the real images $\mathcal{D}_r$ and their labels.

In most cases where synthetic data is used for training, images are generated based on solely a text prompt or a text prompt and an existing image (see Section 2). In this work, we aim to maximally leverage the existing data by utilizing a data generation pipeline which can take as input linear combinations of embeddings of text and *multiple* images.

Concretely, we use the Kandinsky framework, which takes as input real images and captions (Razzhigaev et al., 2023). This pipeline obtains CLIP embeddings for each image and caption, combines them according to specified weights, and passes the joint embedding into the diffusion model to produce a synthetic sample. We chose this pipeline due to its high performance, flexibility, and off-the-shelf ease of use: it achieved strong FID scores relative to competitors and was the first text-to-image generative model that used both image priors and latent diffusion.

3.2 Component #2: Initialize as Small CLIP

We find that initializing a student model in a CLIP-style architecture allows for performance gains relative to a standalone student vision model. In this paper, we distill to three models in the EfficientNet family, a set of small convolutional networks (Tan & Le, 2020). Specifically, we use EfficientNet B0, B1, and B2, with parameter counts of 5.3M, 7.8M, and 9.2M, respectively. For our primary set of experiments, we initialize these models as small CLIP variants, that is, preserve the CLIP text encoder and replace the CLIP image encoder with the EfficientNet model. Each CLIP-EfficientNet model is pretrained on a subset of the DataComp dataset (Gadre et al., 2023).

3.3 Component #3: Knowledge Distillation

Knowledge distillation is a common model compression technique that uses a large, powerful teacher model to train a smaller student model by aligning the student's output probabilities to those of the teacher. There are many variants of loss functions used to align these sets of probabilities, but the most common is based on the Kullback-Leibler (KL) divergence as proposed in Hinton et al. (2015):

$$\mathcal{L}_{KL} = \alpha \cdot T^2 \cdot D_{\text{KL}}(SM(\tilde{y}), SM(\hat{y})) + (1 - \alpha) \cdot CE(\hat{y}, y)$$

where D_{KL} refers to KL divergence, CE is cross entropy, SM is softmax, $\tilde{y}$ is the teacher output probabilities, $\hat{y}$ is the student output probabilities, y is the true labels, α is a hyperparameter that trades off influence from teacher labels and true labels, and T is a temperature parameter.

We use this standard KL setting in our experiments. We have a teacher image encoder which outputs image embeddings of size d^T_{img} and a student image encoder which outputs image embeddings of size d^S_{img}. We also have a common text encoder which produces text embeddings of size d_{text}.

For each task, we append classification heads to both teacher and student models, with linear layers of shape $d^T_{img} \times c$ and $d^S_{img} \times c$, respectively. Before distillation, we finetune the teacher linear layer on the task of interest. We initialize the student layer with the text embeddings of each class: we obtain the embeddings for captions `"A photo of {classname}."` or `"A photo of {classname}, a type of {category}."` for each class and concatenate them into a tensor of shape $d_{text} \times c$ where $d^S_{img} = d_{text}$. Then, during distillation, the teacher and its appended layer are frozen, and both the student and its appended layer are updated.

We use a distillation set $\mathcal{D} = \mathcal{D}_r \cup \mathcal{D}_{s(r)}$ consisting of the real samples $\mathcal{D}_r$ and the synthetic samples $\mathcal{D}_{s(r)}$ generated by conditioning on those real images.

4 Results

We demonstrate that SIDCLIP allows us to approach the performance of CLIP ViT models while using an image encoder with as few as 2% of the parameters. Each of the three components (synthesize, initialize,

Table 1: SIDCLIP outperforms baselines and competing methods.

Dataset	Model	Method	Params (M)	Shot 1	2	4	8	Full
Cars	CLIP ViT-B/32	FT	86	43.9	48.5	58.6	65.9	80.2
	CLIP ViT-L/14	FT	307	78.1	79.0	81.5	83.3	91.1
	EfficientNet B0	Train	5.3	5.2	7.5	21.4	45.2	**87.6**
		$\text{SIDCLIP}_{\text{B/32}}$	5.3	**74.6**	**76.4**	**79.4**	**81.7**	85.5
		$\text{SIDCLIP}_{\text{L/14}}$	5.3	66.4	73.0	75.9	80.4	87.3
	EfficientNet B1	Train	7.8	4.8	7.6	18.3	42.0	**87.6**
		$\text{SIDCLIP}_{\text{B/32}}$	7.8	**72.7**	**73.3**	**76.9**	79.6	84.0
		$\text{SIDCLIP}_{\text{L/14}}$	7.8	60.6	64.2	71.7	**77.2**	85.3
	EfficientNet B2	Train	9.2	5.3	7.9	18.8	43.9	**88.3**
		$\text{SIDCLIP}_{\text{B/32}}$	9.2	**68.6**	**74.2**	**74.1**	78.1	84.0
		$\text{SIDCLIP}_{\text{L/14}}$	9.2	60.9	57.2	72.3	**78.5**	85.2
Flowers	CLIP ViT-B/32	FT	86	65.0	76.6	86.4	90.8	92.8
	CLIP ViT-L/14	FT	307	90.3	94.9	97.5	98.5	98.6
	EfficientNet B0	Train	5.3	34.0	49.8	68.8	83.5	86.9
		$\text{SIDCLIP}_{\text{B/32}}$	5.3	87.3	88.5	89.6	90.5	**91.4**
		$\text{SIDCLIP}_{\text{L/14}}$	5.3	**88.5**	**88.7**	**92.6**	**94.5**	84.6
	EfficientNet B1	Train	7.8	39.4	54.7	71.3	85.7	88.1
		$\text{SIDCLIP}_{\text{B/32}}$	7.8	**86.1**	**86.8**	87.7	89.3	89.7
		$\text{SIDCLIP}_{\text{L/14}}$	7.8	84.5	**86.8**	**90.0**	**93.1**	**93.1**
	EfficientNet B2	Train	9.2	32.3	50.9	69.7	84.3	87.7
		$\text{SIDCLIP}_{\text{B/32}}$	9.2	**86.2**	86.7	87.9	90.3	89.7
		$\text{SIDCLIP}_{\text{L/14}}$	9.2	85.4	**87.4**	**91.3**	**93.9**	**94.1**
Food	CLIP ViT-B/32	FT	86	66.5	70.7	74.5	78.3	86.7
	CLIP ViT-L/14	FT	307	92.8	92.8	93.1	93.4	95.2
	EfficientNet B0	Train	5.3	8.2	14.8	19.9	32.0	83.3
		$\text{SIDCLIP}_{\text{B/32}}$	5.3	**70.6**	**73.3**	**75.5**	**77.1**	86.6
		$\text{SIDCLIP}_{\text{L/14}}$	5.3	58.4	64.0	68.8	72.0	**88.9**
	EfficientNet B1	Train	7.8	10.0	15.3	19.6	30.9	83.9
		$\text{SIDCLIP}_{\text{B/32}}$	7.8	**66.3**	**69.7**	**72.2**	**73.7**	85.6
		$\text{SIDCLIP}_{\text{L/14}}$	7.8	53.1	59.0	64.7	68.2	**87.9**
	EfficientNet B2	Train	9.2	8.5	12.3	17.6	28.7	83.7
		$\text{SIDCLIP}_{\text{B/32}}$	9.2	**66.7**	**70.3**	**72.3**	**73.9**	85.9
		$\text{SIDCLIP}_{\text{L/14}}$	9.2	54.5	59.9	65.3	68.6	**88.2**
DTD	CLIP ViT-B/32	FT	86	47.0	54.3	58.6	64.4	73.0
	CLIP ViT-L/14	FT	307	56.7	61.7	67.9	72.5	79.4
	EfficientNet B0	Train	5.3	16.9	23.0	36.0	49.0	63.5
		$\text{SIDCLIP}_{\text{B/32}}$	5.3	**56.2**	**60.0**	**63.2**	**67.5**	**71.7**
		$\text{SIDCLIP}_{\text{L/14}}$	5.3	51.5	57.1	62.2	66.9	**71.7**
	EfficientNet B1	Train	7.8	18.3	19.2	36.4	50.3	63.8
		$\text{SIDCLIP}_{\text{B/32}}$	7.8	**56.7**	**61.7**	**63.2**	**66.7**	68.7
		$\text{SIDCLIP}_{\text{L/14}}$	7.8	48.9	57.2	62.2	66.5	**70.1**
	EfficientNet B2	Train	9.2	15.6	25.9	39.0	52.5	65.2
		$\text{SIDCLIP}_{\text{B/32}}$	9.2	**57.8**	**60.1**	**63.2**	66.7	68.9
		$\text{SIDCLIP}_{\text{L/14}}$	9.2	50.0	57.3	61.8	**67.7**	**70.3**

distill) is critical in achieving this strong performance. Through a series of ablations and comparisons to SOTA distillation methods, we show that SIDCLIP is the dominant method when operating in a resource constrained setting.

4.1 Experimental Details

Datasets. We report results on four task-specific image classification datasets: StanfordCars, OxfordFlowers, Food101, and DTD (Krause et al., 2013; Nilsback & Zisserman, 2008; Bossard et al., 2014; Cimpoi et al., 2014). We chose these datasets due to the fine-grained nature of their classification tasks. Unlike more general classification datasets such as ImageNet, these datasets are restricted to a very limited domain, and are similar to very specific classification tasks that an end user may want to perform. StanfordCars has 196 classes, OxfordFlowers has 102, Food101 has 101, and DTD has 47. All numbers reported in this paper are Top 1 accuracies on the test sets.

Data-scarce setting. We are generally interested in any limited data setting. For experimental purposes, we simulate a data-scarce setting by creating few shot datasets from existing task-specific datasets. For each dataset, we randomly sample k images per class to produce a k-shot variant of the dataset, for $k = \{1, 2, 4, 8\}$.

Models. We use two CLIP models as teachers, CLIP-ViT-L/14 and CLIP ViT-B/32 (Radford et al., 2021). We also use three EfficientNet models (B0, B1, and B2) as students, each initialized in a CLIP-style model (Tan & Le, 2020; Akinwande et al., 2024). When performing distillation, the teacher is frozen and we update the parameters of both the student model's image encoder and its appended linear layer. When performing finetuning of a CLIP-style model we freeze the parameters of the student model and only update the parameters in the appended linear layer. When finetuning or distilling to a non-CLIP-style model , there is no appended linear layer and we update all model parameters.

Due to computational constraints, the CLIP-EfficientNet B1 and CLIP-EfficientNet B2 models were pretrained with less DataComp data than the CLIP-EfficientNet B0 model. Therefore, we note that while results within each EfficientNet model consistently demonstrate our findings, results *across* EfficientNet models are not necessarily comparable.

Data augmentation. We use RandAugment data augmentation (Cubuk et al., 2019). This is an augmentation strategy that applies random data augmentations to each image and is a top performing augmentation strategy. We apply six augmentations per image.

Zero-shot results. The zero-shot columns in Table 2 and Table 3 always indicates that no real data was used. For our method (the SIDCLIP rows), zero-shot distillation is performed by using 100 synthetic samples generated from only caption information. Lack of zero-shot results due to model incompatibility, unreleased results, or lack of synthetic data, is indicated by a dash (−).

4.2 Main Results

Across several teacher models, small student models, and few-shot dataset variants, SIDCLIP-trained models significantly outperform models trained via standard finetuning. Table 1 compares the results of applying SIDCLIP to those of training small models from scratch. For each dataset, we include upper bound baselines of CLIP ViT-L/14 and CLIP ViT-B/32 finetuned on the few-shot datasets. For each student architecture (EfficientNet B0, EfficientNet B1, and EfficientNet B2), we include one row indicating the results of standard finetuning ("Train") and one row with the results of training using SIDCLIP with each teacher ($\text{SIDCLIP}_{\text{B/32}}$ for CLIP ViT-B/32 and $\text{SIDCLIP}_{\text{L/14}}$ for CLIP ViT-L/14).

Our goal was to leverage the power of CLIP to produce a strong small-scale model, using only limited training data. Our results indicate that, using each of our three components (synthesize, initialize, distill), we are able to obtain notable performance increases of up to 65% higher than the starting models in the few shot setting, with performances that approach and even exceed those of the teacher CLIP models. These findings generally hold across teacher and student models, datasets, and few-shot instances.

On the Cars, Flowers, and DTD datasets, SIDCLIP consistently achieves within around 10-30% of its teacher's performance in the few shot setting and occasionally outperforms the teacher. On Food, SIDCLIP

remains farther from the teacher model, particularly when distilling from CLIP ViT-L/14. We hypothesize that this may be due to more instances of food in the pretraining datasets for both teacher and student. In this case, additional food examples do not add much information to the model.

SIDCLIP with distillation from CLIP ViT-B/32 often outperforms distillation from CLIP ViT-L/14. This may be due to the smaller capacity gap between teacher and student, which has been shown to be beneficial for distillation performance.

4.3 Additional Comparisons

In Table 2 we include comparisons to other similar methods. Rows that include the performance of our proposed method are highlighted. TinyCLIP and TinyViT use distillation to train a downstream image classification model (Wu et al., 2022; 2023). Unlike our method, which allows for specialization on a specific task, these methods focus on maintaining CLIP's overall performance. Since few-shot results were not reported in these papers, we perform an evaluation of some of these methods. For TinyCLIP, we ran few-shot linear probe experiments on the smallest available model (8M parameter image encoder). For TinyViT, we ran few-shot finetuning experiments on the smallest available model (5.4M parameters).

Our method outperforms competitors by large margins in the few-shot setting. Although SIDCLIP performs worse than competitors on zero-shot, we note that the other models here are up to two times larger, and our strong few-shot results highlight the value of our pipeline in the intended setting.

Table 2: SIDCLIP outperforms similar methods in the few shot setting.

Model	Params (M)	Zero shot	Few shot (k)	Full shot
Cars				
EfficientNet B0 (SIDCLIP)	5.3	65.3	**79.4** (4)	85.5
TinyViT-5M Wu et al. (2022)	5.4	-	13.9 (4)	87.7
EfficientNet B1 (SIDCLIP)	7.8	60.4	76.9 (4)	84.0
TinyCLIP Wu et al. (2023)	8	7.8	17.1 (4)	31.1
EfficientNet B2 (SIDCLIP)	9.2	61.6	72.3 (4)	85.2
TinyViT Popp et al. (2024)	11	**81.9**	–	**90.7**
ResNet18 Li et al. (2023)	11	20.4	39.7 (5)	–
Flowers				
EfficientNet B0 (SIDCLIP)	5.3	10.1	**92.6** (4)	**94.6**
TinyViT-5M Wu et al. (2022)	5.4	-	74.9 (4)	92.3
EfficientNet B1 (SIDCLIP)	7.8	4.8	90.0 (4)	93.1
TinyCLIP Wu et al. (2023)	8	56.5	86.8 (4)	82.4
EfficientNet B2 (SIDCLIP)	9.2	6.4	91.3 (4)	94.1
TinyViT Popp et al. (2024)	11	**68.3**	–	90.6
ResNet18 Li et al. (2023)	11	18.2	54.3 (5)	–
Food				
EfficientNet B0 (SIDCLIP)	5.3	61.9	**75.5** (4)	**86.6**
TinyViT-5M Wu et al. (2022)	5.4	-	21.0 (4)	84.7
EfficientNet B1 (SIDCLIP)	7.8	55.7	72.2 (4)	85.6
TinyCLIP Wu et al. (2023)	8	55.1	58.4 (4)	72.7
EfficientNet B2 (SIDCLIP)	9.2	56.6	72.3 (4)	86.0
TinyViT Popp et al. (2024)	11	**71.9**	–	83.0
ResNet18 Li et al. (2023)	11	35.7	44.0 (5)	–

4.4 Ablations

Table 3 demonstrates the additional value of each SIDCLIP component and set of components. The columns "Synthesize," "Initialize," and "Distill" indicate the presence or absence of each component in each row. We select only the Flowers dataset and CLIP ViT-L/14 teacher supervision for our set of ablations for computational reasons.

We note that each of the components alone leads to an improvement over the baseline. However, not all combinations lead to an improvement. For instance, distilling to a CLIP-initialized model (Initialize ✓and Distill ✓) often underperforms distilling to the standalone model (Distill ✓). Additionally, we note that the usage of synthetic data during training (Synthesize ✓) consistently underperforms the baseline, possibly because there is not enough structure and supervision for the small model to reliably learn from the synthetic images.

Across shots and student models, we note that all three elements of SIDCLIP are necessary to consistently obtain the best performance.

Table 3: Ablations of every subset of SIDCLIP components on the Flowers dataset. Distillation supervision is from the CLIP ViT-L/14 teacher. All three components are required to obtain the best performance.

				Shot					
Model	**Synthesize**	**Initialize**	**Distill**	**0**	**1**	**2**	**4**	**8**	**Full**
CLIP ViT-L/14				76.1	78.1	79.0	81.4	83.3	91.1
EfficientNet B0	✗	✗	✗	-	34.0	49.8	68.8	83.5	86.9
	✓	✗	✗	1.14	8.4	11.2	12.3	14.9	70.7
	✗	✓	✗	-	50.1	64.0	78.8	89.7	91.4
	✗	✗	✓	-	55.4	71.6	86.6	91.9	93.1
	✓	✓	✗	-	11.8	14.4	14.8	16.1	85.2
	✓	✗	✓	3.0	57.3	64.5	70.1	78.4	82.1
	✗	✓	✓	-	65.4	77.0	88.2	93.3	94.3
	✓	✓	✓	**10.1**	**88.5**	**88.7**	**92.6**	**94.5**	**94.6**
EfficientNet B1	✗	✗	✗	-	39.4	54.7	71.3	85.7	88.1
	✓	✗	✗	1.3	8.6	10.7	12.6	15.3	73.0
	✗	✓	✗	-	31.5	42.4	56.6	77.0	82.4
	✗	✗	✓	-	57.5	72.7	88.0	92.9	93.9
	✓	✓	✗	-	10.6	12.4	13.3	15.1	68.4
	✓	✗	✓	2.1	54.3	65.5	75.1	78.5	80.7
	✗	✓	✓	-	45.8	58.8	78.1	88.8	90.7
	✓	✓	✓	**4.8**	**84.5**	**86.8**	**90.0**	**93.1**	**93.1**
EfficientNet B2	✗	✗	✗	-	32.3	50.9	69.7	84.3	87.7
	✓	✗	✗	1.3	8.9	10.5	13.1	15.1	69.8
	✗	✓	✗	-	31.8	42.4	55.2	75.0	80.1
	✗	✗	✓	-	55.4	70.4	86.4	92.4	93.6
	✓	✓	✗	-	10.4	14.4	13.6	15.1	69.2
	✓	✗	✓	2.6	57.4	58.7	73.7	80.2	80.6
	✗	✓	✓	-	44.5	45.6	61.0	88.5	90.9
	✓	✓	✓	**6.5**	**85.4**	**87.4**	**91.3**	**93.9**	**94.1**

4.5 Additional Distillation Methods

We use KL distillation for our experiments due to its simplicity and strong performance. However, we expect SIDCLIP to be beneficial when used with a range of other distillation methods. In Table 4 we show a limited

Table 4: SIDCLIP works with several distillation methods.

Method	Shot 2	4	8
Baseline	49.8	68.8	83.5
KL Distillation	88.7	92.6	94.5
DKD Distillation	90.0	92.8	94.2
Intermediate Distillation	87.2	91.5	93.6

set of results indicating that SIDCLIP works synergistically with other distillation methods to obtain strong results. We show a small set of results on the Flowers dataset, using an EfficientNet B0 student and a CLIP ViT-L/14 teacher for distillation supervision. We compare the baseline, or standard finetuning, with three SIDCLIP variants: KL distillation, DKD distillation, and intermediate distillation. KL distillation is the setting we used throughout the rest of the paper. Decoupled Knowledge Distillation (DKD) (Zhao et al., 2022) is a recent SOTA distillation method which separates the standard distillation loss into two terms: a target class term and a non-target class term. We use a simple version of intermediate distillation, similar to what is proposed in (Wu et al., 2021). We generate pseudopredictions based on the features produced by the last layers of the teacher and student networks and use a KL loss between the pseudopredictions during distillation. We show that SIDCLIP leads to improved model performance, relative to the baseline, across the three distillation methods.

4.6 Computational Overhead

Synthetic data generation is expensive, but can be worthwhile in the creation of a smaller, more efficient, downstream model. We provide wall clock time references for the data generation process and for distillation using the additional synthetic images. Using one Nvidia A6000 GPU, the generation of 100 images takes 840.6 seconds (approximately 8 seconds per image). The difference in time for one epoch when using only real data as opposed to real and synthetic data can primarily be ascribed to the increased amount of data being processed. For the 8 real shot setting (8 images per class), our CLIP-initialized EfficientNet B0 model takes around 24 seconds for one epoch of distillation, and the 8 real + 300 syn shot setting (308 total images per class) takes around 1021 seconds for one epoch of distillation.

5 Conclusion

We present the SIDCLIP (Synthesize-Initialize-Distill CLIP) method, which consists of 1) augmenting the limited training data with task-specific *synthetic data* generated by using linear combinations of the CLIP image and text embeddings of existing real data; 2) *initializing* the small model as a CLIP-style model; and 3) using *knowledge distillation* to transfer more fine-grained classification information from a powerful teacher. SIDCLIP achieves the best few-shot performance on several task-specific datasets relative to existing methods, improving model performance by up to 65%, approaching and even occasionally exceeding the teacher's performance. These results hold across four datasets, two teacher architectures, three student architectures, and several few-shot settings. Our method achieves this performance by efficiently utilizing existing data to extract task-specific information from a large scale VLM such as CLIP in a resource constrained setting.

References

Victor Akinwande, Mohammad Sadegh Norouzzadeh, Devin Willmott, Anna Bair, Madan Ravi Ganesh, and J. Zico Kolter. Hyperclip: Adapting vision-language models with hypernetworks, 2024. URL https://arxiv.org/abs/2412.16777.

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J. Fleet. Synthetic data from diffusion models improves imagenet classification, 2023. URL https://arxiv.org/abs/2304.08466.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Yuxuan Cai, Jiangning Zhang, Haoyang He, Xinwei He, Ao Tong, Zhenye Gan, Chengjie Wang, Zhucun Xue, Yong Liu, and Xiang Bai. Llava-kd: A framework of distilling multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 239–249, October 2025.

Akshay Chawla, Hongxu Yin, Pavlo Molchanov, and Jose Alvarez. Data-free knowledge distillation for object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3289–3298, January 2021.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019. URL https://arxiv.org/abs/1909.13719.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022. URL https://arxiv.org/abs/2208.07339.

Gongfan Fang, Kanya Mo, Xinchao Wang, Jie Song, Shitao Bei, Haofei Zhang, and Mingli Song. Up to 100x faster data-free knowledge distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6597–6604, Jun. 2022. doi: 10.1609/aaai.v36i6.20613. URL https://ojs.aaai.org/index.php/AAAI/article/view/20613.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot, 2023. URL https://arxiv.org/abs/2301.00774.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets, 2023. URL https://arxiv.org/abs/2304.14108.

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?, 2023. URL https://arxiv.org/abs/2210.07574.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019. URL https://arxiv.org/abs/1902.00751.

Ashraful Islam, Chun-Fu Richard Chen, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, and Richard J Radke. Dynamic distillation network for cross-domain few-shot recognition with unlabeled data. *Advances in Neural Information Processing Systems*, 34:3584–3595, 2021.

Jinseong Jang, Chunfei Ma, and Byeongwon Lee. Vl2lite: Task-specific knowledge distillation from large vision-language models to lightweight networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30073–30083, June 2025.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision, 2021. URL https://arxiv.org/abs/2102.05918.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 709–727, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19827-4.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021. URL https://arxiv.org/abs/2104.08691.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL https://arxiv.org/abs/2201.12086.

Xuanlin Li, Yunhao Fang, Minghua Liu, Zhan Ling, Zhuowen Tu, and Hao Su. Distilling large vision-language model with out-of-distribution generalizability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2492–2503, 2023.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 1950–1965. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/0cde695b83bd186c1fd456302888454c-Paper-Conference.pdf.

Mengyuan Ma, Lin Qian, and Hujun Yin. Kdnet: Leveraging vision-language knowledge distillation for few-shot object detection. In Michael Wand, Kristína Malinovská, Jürgen Schmidhuber, and Igor V. Tetko (eds.), *Artificial Neural Networks and Machine Learning – ICANN 2024*, pp. 153–167, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-72335-3.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Niclas Popp, Jan Hendrik Metzen, and Matthias Hein. Zero-shot distillation for image encoders: How to make effective use of synthetic data, 2024. URL https://arxiv.org/abs/2404.16637.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Anton Razzhigaev, Arseniy Shakhmatov, Anastasia Maltseva, Vladimir Arkhipkin, Igor Pavlov, Ilya Ryabov, Angelina Kuts, Alexander Panchenko, Andrey Kuznetsov, and Denis Dimitrov. Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion, 2023. URL https://arxiv.org/abs/2310.03502.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models, 2024. URL https://arxiv.org/abs/2306.11695.

Ximeng Sun, Pengchuan Zhang, Peizhao Zhang, Hardik Shah, Kate Saenko, and Xide Xia. Dime-fm: Distilling multimodal and efficient foundation models, 2023. URL https://arxiv.org/abs/2303.18232.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. URL https://arxiv.org/abs/1905.11946.

Pavan Kumar Anasosalu Vasu, Hadi Pouransari, Fartash Faghri, Raviteja Vemulapalli, and Oncel Tuzel. Mobileclip: Fast image-text models through multi-modal reinforced training, 2024. URL https://arxiv.org/abs/2311.17049.

Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3), June 2020. ISSN 0360-0300. doi: 10.1145/3386252. URL https://doi.org/10.1145/3386252.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models, 2022. URL https://arxiv.org/abs/2109.01903.

Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Tinyvit: Fast pretraining distillation for small vision transformers, 2022. URL https://arxiv.org/abs/2207.10666.

Kan Wu, Houwen Peng, Zhenghong Zhou, Bin Xiao, Mengchen Liu, Lu Yuan, Hong Xuan, Michael Valenzuela, Xi, Chen, Xinggang Wang, Hongyang Chao, and Han Hu. Tinyclip: Clip distillation via affinity mimicking and weight inheritance, 2023. URL https://arxiv.org/abs/2309.12314.

Yimeng Wu, Mehdi Rezagholizadeh, Abbas Ghaddar, Md Akmal Haidar, and Ali Ghodsi. Universal-kd: Attention-based output-grounded intermediate layer knowledge distillation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7649–7661, 2021.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision, 2021. URL https://arxiv.org/abs/2111.11432.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. URL https://arxiv.org/abs/2303.15343.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation, 2022. URL https://arxiv.org/abs/2203.08679.

A Appendix

A.1 Qualitative analysis of synthetic images

Figure 3 shows examples of synthetic data used in the SIDCLIP pipeline. When conditioned on one or two real images as shown in the last two columns, we can see that the synthetic images directly mirror features in the real images more than when generation is only conditioned on the caption. For instance, note the colors of the Volkswagen Beetle and the butter on the waffles.

We also note in particular that the Flowers dataset tends to yield relatively poor zero-shot performance. We can observe how much the caption-only "red ginger" image differs from both the real images and the real-image-conditioned synthetic images. Additionally, the caption-only "yellow iris" includes less background foliage. This dataset-specific discrepancy may be a contributor to the impacted zero-shot performance.

Caption	Real Image 1	Real Image 2	Synthetic image generation conditioned on:		
			caption only	caption + real image 1	caption + both real images
'Rolls-Royce Phantom Sedan 2012'					
'Volkswagen Beetle Hatchback 2012'					
'red ginger'					
'yellow iris'					
'waffles'					
'spaghetti bolognese'					

Figure 3: Synthetic images mirror the real images more closely when conditioned on real images and captions, rather than captions only.

A.2 Details of synthetic image generation

The caption we used to produce the text embedding is always the classname. For zero shot, we use only the caption to prompt the diffusion model and provide no real image samples. For 1 shot, we use the single image in each class as the only real image sample. For 2, 4, and 8 shot, we sample two images from each class of our few shot dataset. In the 1 shot case, we use weights of 0.4 for text and 0.6 for image, and for larger shots, we use weights of 0.2 for the text and 0.4 for each image.