Training with Real instead of Synthetic Generated Images Still Performs Better

Anonymous CVPR submission

Abstract

Recent advances in text-to-image models have inspired many works that seek to train models with synthetic images, capitalizing on the ability of modern generators to control the data we synthesize and thus train on. However, synthetic images ultimately originate from the upstream data pool used to train the generative model we sample from—does the intermediate generator add any gain over simply training on relevant parts of the upstream data directly? In this paper, we study this question in the setting of task adaptation by comparing training with task-targeted synthetic data generated from Stable Diffusion—a generative model trained on the LAION-2B dataset—against training with targeted real images sourced directly from LAION-2B. We show that while targeted synthetic data can aid model adaptation, it largely lags behind targeted real data. Overall, assuming we have access to the upstream data pool of the generator, we should be cautious in our use of generated synthetic data. Studying synthetic data in settings where the upstream data is not accessible—for instance, due to copyright or privacy concerns—or searching for benefits from synthetic data even when it is present are opportunities for future work.

1. Introduction

Modern machine learning systems fundamentally depend on the quantity, quality, and distribution of their training data, all of which strongly impact downstream performance. Motivated by this observation, the field is actively developing algorithms to automatically curate high-quality data at scale. In particular, sourcing synthetic data from conditional generative models is becoming increasingly commonplace, as generative models enable algorithmic control over what data to sample and train on. For example, in the neighboring field of natural language processing (NLP), advances in language models have enabled controllable generation of large-scale synthetic instruction-tuning datasets [12, 32].

Likewise, in computer vision, modern text-to-image models increasingly allow for controlled image generation, inspiring researchers to search for similar possibilities. The high-dimensional and continuous nature of images often results in lower-quality synthetic visual data relative to synthetic discrete text in NLP; nonetheless, recent attempts using synthetic visual data shows promise [2, 22, 29]. For instance, SynCLR [28] cleverly prompts Stable Diffusion for synthetic images tailored to pre-specified downstream image classification tasks; a CLIP model trained on the resulting *targeted* synthetic dataset can outperform CLIP trained on a significantly larger untargeted dataset of real images. The underlying factors driving these gains, however, warrant closer examination. In particular, prior work has often compared *task-targeted* synthetic data to *general* real data, thereby entangling the effects of training on synthetic versus real data with the effects of targeted versus general data collection. However, critically, we observe that these variables are not intrinsically conflated: any synthetic data we generate from a model is ultimately derived from the upstream dataset used to train the generator. Thus, instead of sampling targeted synthetic data, we can alternatively *retrieve targeted real* data directly from that upstream dataset (Figure 1). In doing so, we exactly isolate the contribution of the generative model. Under this framework, we ask: **what gains (if any) does the intermediate step of training a generator and sampling synthetic data for training provide?** What gains are due to targeted data collection?

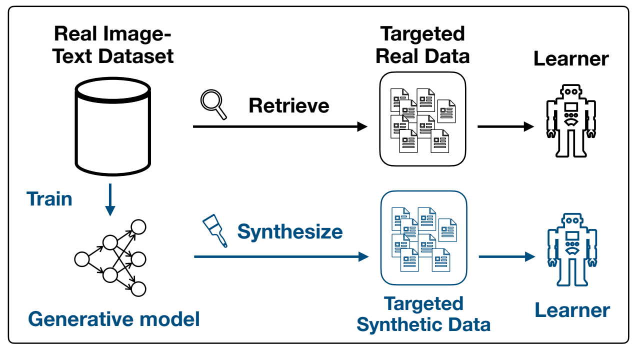


Figure 1. Given an upstream dataset of general real image-text pairs, we wish to derive a *targeted* dataset to train a learner on some target task. We can either **retrieve relevant real data** directly from the general dataset (top path), or we can first train a generative model and then **synthesize targeted synthetic data** (bottom path). Our work compares these two approaches.

In this paper, we operationalize these questions in the task adaptation setting, where high-quality targeted data is critical. We empirically compare training with *targeted synthetic images* generated from Stable Diffusion—a text-to-image model trained on the LAION-2B dataset—against training with *targeted real images* carefully sourced from LAION-2B itself. Through experiments across several data scales on three datasets where training on targeted synthetic data has shown promise [28], we find that while targeted synthetic data can be useful for model adaptation, **targeted synthetic data largely lags behind targeted real data**. Our analysis suggests that synthetic images may distort class-specific visual details that targeted real images preserve. Overall, assuming we have access to a generative model's upstream training data, our results show that synthetic data does not provide strong gains. We conclude by presenting opportunities for future work: synthetic data has exciting potential in settings where the upstream data is inaccessible or infeasible to download; or, even when the upstream data is available, synthetic data may provide gains that we have simply not yet found.

2. Related Work

Learning from synthetic data. Synthetic data has been widely explored in the context of many machine learning problems [4, 6, 9, 12, 13, 18, 24, 25, 32]. In computer vision, synthetic data has traditionally been sourced from expert-crafted simulation and rendering pipelines [6, 18, 20]. Recent advances in text-to-image synthesis via diffusion models [11, 21, 26] are changing this paradigm, inspiring a new line of work that seek to train visual models on synthetic data algorithmically sampled from conditional image generation models [2, 8, 22, 29]. This shift in the source of synthetic images from programmatic simulation to a learned generator that itself derives from an upstream dataset poses a new question: does the intermediate step of training a generator and sampling synthetic data provide any gains over simply training on the upstream data directly? Our work seeks to elucidate this phenomena.

Adapting pretrained vision models. Large-scale pretrained vision models like CLIP [5, 19] offer transferable visual features that benefit a wide range of downstream tasks; it is now common to use pretrained models as a starting point for task-specific models instead of training from scratch. Our work also uses CLIP as the foundation for task adaptation. The primary methods for adapting CLIP are linear probing and finetuning, but many other methods have been proposed, focusing on parameter efficiency [3, 34], performance [7], and distributional robustness [15, 33]. Our work explores CLIP adaptation from a data-centric perspective; we compare the use of real versus synthetic data when constructing task-targeted datasets for simple finetuning.

3. Problem Setting and Method

Given a large dataset $\mathcal{D}$ of general real image-text pairs and a downstream visual classification task specified as a set of text class names $\mathcal{C}$, we wish to algorithmically construct a **targeted adaptation dataset** $\mathcal{D}_\mathcal{C}$ of images and labels to finetune and improve a pretrained vision model's performance on the downstream task. We compare two approaches for sourcing targeted data, shown in Figure 1: (1), we use $\mathcal{D}$ to first train a text-to-image generator G and subsequently query G to build a dataset $\mathcal{D}_\mathcal{C}^{(\text{synthetic})}$ of targeted synthetic images. Alternatively, (2) we source data directly from $\mathcal{D}$ by finding a relevant subset of targeted real images $\mathcal{D}_\mathcal{C}^{(\text{retrieved})} \subset \mathcal{D}$. We detail each approach below.

Sourcing data by generating synthetic images. We follow SynCLR [28], a method representative of the current state-of-the-art for curating synthetic training data from off-the-shelf text-to-image models. In brief, given the set of visual class names $\mathcal{C}$, we first synthesize a large corpus of corresponding image captions by prompting a large language model (details in Appendix A.1). We then use those captions as input for a text-to-image generator G trained on the upstream data $\mathcal{D}$, yielding a large set of synthesized images x_i. Each image x_i is assigned a one-hot class label y_i according to the class name $c \in \mathcal{C}$ used to synthesize its caption. These synthetic images and labels (x_i, y_i) form our curated dataset $\mathcal{D}_\mathcal{C}^{(\text{synthetic})}$.

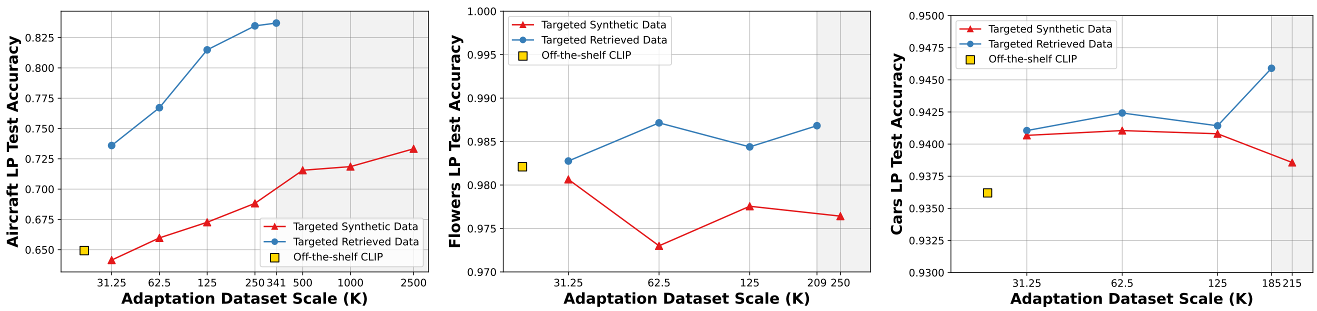
Sourcing data by retrieving real images. Alternatively, rather than querying a generator trained on an upstream dataset $\mathcal{D}$, we can directly source images from $\mathcal{D}$ itself. $\mathcal{D}$ consists of image-text pairs (x_i, t_i). To find relevant pairs, we design a simple two-step retrieve-then-filter strategy inspired by prior work on neural priming [31]. First, we gather a preliminary set S of images by coarsely **retrieving** all images x_i whose corresponding caption t_i contains at least one target class name $c \in \mathcal{C}$ as a substring:

$$S = \{(x_i, t_i) \in \mathcal{D} \colon \exists c \in \mathcal{C} \text{ such that } c \in t_i\}.$$

Each selected image-text pair $(x_i, t_i) \in S$ is further assigned a one-hot class label y_i based on the matched class name $c \in t_i$. We obtain the final targeted dataset $\mathcal{D}_\mathcal{C}^{(\text{retrieved})}$ by **filtering** the candidate set S for images whose CLIP similarity with text describing the downstream domain of interest passes some manually-defined threshold τ:

$$\mathcal{D}_\mathcal{C}^{(\text{retrieved})} = \{(x_i, t_i, y_i) \in S \colon \text{CLIP}(x_i, \text{domain text}) > \tau\}$$

For example, if our domain $\mathcal{C}$ is a set of flower names, we filter for images that have sufficiently high similarity with the text "a photo of a flower". We find $\tau = 0.2$ generally works well. See Appendix A.2 for further details.



(a) FGVC-Aircraft (b) Oxford Flowers102 (c) StanfordCars

Figure 2. We adapt a pretrained CLIP image encoder (**gold squares**) to different downstream image classification tasks, using either targeted synthetic data generated from a Stable Diffusion model trained on LAION-2B (**red triangles**) or using targeted real data directly retrieved from LAION-2B (**blue circles**). Overall, while adapting CLIP with targeted synthetic data can improve downstream linear probing accuracy over an unadapted model, synthetic data generally lags behind targeted real data. This gap persists even when we scale the sample size of the synthetic adaptation dataset beyond the maximum amount of (finite) targeted real data available (**gray shaded regions**).

4. Experiments and Results

Experimental setup. We compare the efficacy of targeted synthetic data versus retrieved (real) data for adapting a pretrained model to a downstream image classification task. We focus evaluation on three standard benchmarks where synthetic data has thus far shown promise versus similar-scale *untargeted* real data [28]: FGVC-Aircraft [16], StanfordCars [14], and Oxford Flowers102 [17].

For each downstream benchmark, we first curate an adaptation dataset $\mathcal{D}_C$ by either (1) generating synthetic images with Stable Diffusion 1.5 [21], trained on the LAION-2B dataset [23], or (2) retrieving directly from LAION-2B (Section 3). We then adapt a LAION-2B pretrained CLIP ViT-B/16 [5] image encoder by finetuning on the adaptation dataset $\mathcal{D}_C$ for a fixed 30 epochs with a cross-entropy classification loss. Finally, we report the test set linear probing (LP) accuracy, using the validation set to identify the best epoch and hyperparameters. Further training and hyperparameter details are provided in Appendix B.

4.1. Main results

Our main findings are illustrated in Figure 2.

At equal data scales, targeted synthetic data lags a retrieval approach. While finetuning with targeted synthetic data can provide gains over an unadapted CLIP model, finetuning with targeted retrieved data provides matching and often stronger performance in all settings considered. For example, on aircraft classification (Figure 2a), finetuning on 250k synthetic aircraft images improves downstream linear probing accuracy by 3.9 points (64.9% $\rightarrow$ 68.8%) over off-the-shelf CLIP, but finetuning on 250k retrieved aircraft images boosts performance by a massive 18.6 points (64.9% $\rightarrow$ 83.5%). Moreover, on Flowers102 (Figure 2b), adapting CLIP with targeted synthetic data can *hurt* performance, while targeted retrieved data improves or at least does not hurt performance on all three benchmarks. Assuming we have equal amounts of targeted retrieved and synthetic data, adapting with retrieved data is the clear winner.

Synthetic data can sometimes decrease the gap with retrieved data given increasing scale, but remains behind. The amount of data we can collect via retrieval is fundamentally finite and limited based on the upstream data pool. For example, even after searching through all 2 billion LAION samples for images relevant to the Aircraft benchmark, our retrieval-based curation method found only 341k targeted samples. In contrast, it is easy to create ever-larger synthetic datasets by simply generating more images. Scaling the synthetic adaptation dataset size beyond the amount of retrieved data available (illustrated in the gray-shaded regions of Figure 2), we find that increasing the amount of targeted synthetic data does not always improve performance. On StanfordCars and Flowers102, for instance, scaling from 125k synthetic images to 210k+ synthetic images barely shifts the downstream accuracy. On Aircraft, scaling does help; there is a clear upward trend in performance as the amount of targeted synthetic data increases (*e.g.*, scaling from 250k $\rightarrow$ 500k synthetic images improves performance from 68.8% $\rightarrow$ 71.6%). However, synthetic data still lags retrieved data: matching the performance of a mere 31.25k retrieved aircraft images requires scaling the synthetic adaptation dataset to 2.5M images, reflecting an 80x difference in dataset size and required finetuning compute. Naively extrapolating this ratio outwards, matching the performance of the full 341k retrieved adaptation dataset would require nearly 30 million synthetic images. We note, however, that synthetic data is unlikely to truly scale infinitely, as synthetic data fundamentally derives from the (finite) training set of our generative model. Nonetheless, the performance of synthetic data is likely unsaturated at the 2.5M scale (*i.e.*, accuracy is still trending up); studying whether further scal-

Figure 3. We visualize synthetic (middle box) and retrieved real (right box) aircraft images, comparing to ground truth (left box). While the synthetic images are recognizable as aircraft, they often distort key details such as the wheel configuration that retrieved images preserve.

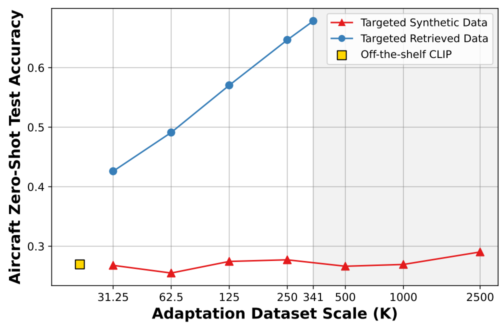


Figure 4. Zero-shot Aircraft accuracy of a **pretrained CLIP model** adapted with either **synthetic** or **retrieved** airplane images.

ing can outperform retrieved data is left for future work.

4.2. Why does synthetic data lag retrieved data?

We further analyze the Aircraft task, as it was the only task where adapting CLIP with either synthetic or retrieved data yielded linear probing (LP) accuracy gains. Figure 3 visualizes a few randomly-chosen images from our Aircraft adaptation datasets. Although the synthetic images are identifiable as airplanes, they often misrepresent class-specific details. For example, a correctly depicted Airbus A320 should feature two sets of dual wheels at its rear, yet our synthetic images often exhibit incorrect wheel configurations. In contrast, retrieved images preserve these details.

We hypothesize that this qualitative discrepancy in detail precision may partially explain why synthetic data lags retrieved data. Specifically, we hypothesize that while adapting CLIP with targeted synthetic data helps align CLIP's representation to the broad aircraft domain (as evidenced by the observed LP accuracy gains), synthetic images alone are too noisy to directly learn an effective task model (especially in a fine-grained classification setting like Aircraft). To quantitatively test this hypothesis, we evaluate the *zero-shot accuracy* of CLIP models adapted with targeted synthetic and retrieved aircraft images (Figure 4). Overall, adapting models with retrieved images yields strong zero-shot performance that improves with dataset scale, while adapting with synthetic images barely changes zero-shot accuracy from an unadapted CLIP baseline. Notably, CLIP adapted with either 31.25k retrieved images or 2.5M synthetic images both achieve a similar LP accuracy ($\sim 73\%$), yet the model adapted with synthetic data achieves a much worse zero-shot accuracy (29.0% versus 42.6%). Thus, models adapted with synthetic data have distinctly different behaviors; additional linear probing after model adaptation is crucial for the gains from synthetic data that we observed.

5. Discussion

Conclusion. Our work sought to answer a key question: given a large pool of general image-text data and a desired downstream task, what is the best way to make use of that data for adapting a pretrained model? Is it better to train a generative model on the data pool and sample task-targeted synthetic images for adaptation? Or do we prefer to leverage the general data directly, by finding a relevant subset? On three tasks where synthetic data has been shown effective, we discover that using relevant real data directly via retrieval is superior, partially because synthetic images from current text-to-image model often corrupt task-relevant visual details. Thus, training a generative model and sampling synthetic data does not provide any strong gain.

Limitations and Future Work. There are a few asterisks to the generality of our results that suggest future opportunities for synthetic visual data. First, we assume access to the generative model's upstream training set. This may not always hold—the upstream pool may be publicly unavailable due to copyright or privacy concerns; even if it is shared, it may be infeasible for end-users to utilize (*e.g.*, Stable Diffusion's weights are 2GB in size, whereas LAION-2B is over 200TB). Second, the downstream tasks we evaluated all admitted a simple substring-matching retrieval approach for sourcing targeted real data. However, there may be scenarios where retrieving targeted real data is challenging, yet training a generative model to produce such targeted data is easy. For instance, in NLP, instruction data may be difficult to extract from pretraining corpora but is easy to generate via a language model. What analogous settings can we find in vision? Third, our experiments focused on three fine-grained visual classification tasks. Can synthetic data provide gains over real data for a more broad visual task? Finally, we consider synthetic data and real data separately—would mixing them provide complementary gains? We leave these questions for future work.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint arXiv:2304.08466*, 2023. 1, 2

[3] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 2

[4] Manel Baradad Jurjo, Jonas Wulff, Tongzhou Wang, Phillip Isola, and Antonio Torralba. Learning to see by looking at noise. *Advances in Neural Information Processing Systems*, 34:2556–2569, 2021. 2

[5] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023. 2, 3, 1

[6] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 2

[7] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19338–19347, 2023. 2

[8] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022. 2

[9] Xuanli He, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. Generate, annotate, and learn: Nlp with synthetic text. *Transactions of the Association for Computational Linguistics*, 10:826–842, 2022. 2

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 1

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[12] Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. *arXiv preprint arXiv:2212.09689*, 2022. 1, 2

[13] Jaehun Jung, Peter West, Liwei Jiang, Faeze Brahman, Ximing Lu, Jillian Fisher, Taylor Sorensen, and Yejin Choi. Impossible distillation: from low-quality model to high-quality dataset & model for summarization and paraphrasing. *arXiv preprint arXiv:2305.16635*, 2023. 2

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

[15] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022. 2

[16] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 3

[17] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 3

[18] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017. 2

[19] 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. PMLR, 2021. 2

[20] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 102–118. Springer, 2016. 2

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3

[22] Mert Bülent Sarıyıldız, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8011–8021, 2023. 1, 2

[23] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 3

[24] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017. 2

[25] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the

game of go without human knowledge. *nature*, 550(7676): 354–359, 2017. 2

[26] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2

[27] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1

[28] Yonglong Tian, Lijie Fan, Kaifeng Chen, Dina Katabi, Dilip Krishnan, and Phillip Isola. Learning vision from models rivals learning vision from data. *arXiv preprint arXiv:2312.17742*, 2023. 1, 2, 3

[29] Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2

[30] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[31] Matthew Wallingford, Vivek Ramanujan, Alex Fang, Aditya Kusupati, Roozbeh Mottaghi, Aniruddha Kembhavi, Ludwig Schmidt, and Ali Farhadi. Neural priming for sample-efficient adaptation. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[32] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022. 1, 2

[33] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7959–7971, 2022. 2, 1

[34] 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. 2

Training with Real instead of Synthetic Generated Images Still Performs Better

Supplementary Material

A. Details in Methodology

A.1. Curating data by generating synthetic images

Given a set of visual class names $\mathcal{C}$ from our target task, we first synthesize a large corpus of image captions for each class name by prompting a large language model (we use LLamA-2 7B [30]). For each concept name $c \in \mathcal{C}$, we use three type of prompts to convert c into an image caption following [28]. For the sake of completeness, we detail the prompts here:

1. $c \mapsto$ caption. We prompt the language model (LM) to directly translate the class name into a caption using a prompt with 3 few-shot in-context examples.

2. $c, background \mapsto$ caption. We prompt the LM with an additional background attribute that is randomly sampled from a set that is predetermined based on the domain of $\mathcal{C}$. For example, if $\mathcal{C}$ contains a list of flower names, then possible background attributes might include "garden," "meadow," or "forest." These background attributes are automatically generated by prompting a strong instruction-tuned language model such as GPT-4 [1] with the class names $\mathcal{C}$. We provide the LM with 3 in-context examples of $c, background \mapsto$ caption mappings.

3. $c, relation \mapsto$ caption. We prompt with an additional spatial relationship attribute that is sampled from a domain-invariant set of relationships, such as "next to," "below," "besides," etc. We provide 3 in-context examples of $c, relation \mapsto$ caption mappings.

Each of these captions are directly used as text input to Stable Diffusion 1.5 to produce our targeted synthetic dataset $\mathcal{D}_{\mathcal{C}}^{(\text{synthetic})}$. When sampling from Stable Diffusion, we denoise for 50 DDIM [27] steps starting from Gaussian noise, using a classifier-free guidance [10] scale of 2.5.

A.2. Curating data by retrieving real images

After obtaining a candidate set of image-text pairs $S = \{(x_i, t_i) \in \mathcal{D} \colon \exists c \in \mathcal{C} \text{ such that } c \in t_i\}$, we wish to filter S to minimize false-positive image-text pairs where the text t_i contains a class name $c \in \mathcal{C}$, but is unrelated to the desired domain. For example, in the Aircraft task, one of the class names is "Tornado" (a type of military aircraft), but naively searching based on this class name returns many candidate images of a tornado weather event. Thus, we filter S to only keep images that are actually relevant to the desired domain via CLIP cosine similarity score. Recall from Section 3:

$$\mathcal{D}_{\mathcal{C}}^{(\text{retrieved})} = \{(x_i, t_i, y_i) \in S \colon \text{CLIP}(x_i, \text{domain text}) > \tau\}$$

The domain text (*e.g.*, if the desired classes $\mathcal{C}$ are aircraft names then the domain text might be "a photo of an airplane") here can be manually-written, or it can be automatically generated by prompting a language model with the set of class names $\mathcal{C}$.

To find a reasonable filtering threshold τ for a desired task domain $\mathcal{C}$, we simply try a sweep $\tau \in \{0.19, 0.2, 0.21\}$ and select the optimal threshold based on downstream validation set performance. The set $\{0.19, 0.2, 0.21\}$ was selected by qualitatively visualizing the CLIP similarity of a few images from the downstream benchmark training set with the desired domain text. This hyperparmeter is not finely-tuned in our paper; we leave more a more systematic ablation study to future work.

B. Details in Experimental Setup

B.1. Finetuning details

To finetune CLIP for a specific downstream image classification task, we first initialize a linear readout head W using the weights from the text-based zero-shot CLIP model [5]. Concretely, we initialize W using the CLIP text embeddings of the class names for the desired downstream task. We then append the classification head W on top of CLIP's vision encoder, and train end-to-end using a standard cross entropy classification loss against one-hot labels.

We could alternatively choose to finetune CLIP with a contrastive objective, where each positive pair is a synthetic or retrieved image alongside its corresponding caption. However, we find that cross entropy finetuning performs better across the board, so we use cross entropy finetuning for all experiments in our paper.

B.2. Hyperparameter details

We start with relatively standard hyperparameters from prior work [33], and tune them in our setting by finetuning CLIP on a small-scale dataset of retrieved or synthetic images and grid-sweeping learning rate and batch size. From the hyperparmeters we tried at this scale, we find the following work best for both synthetic and retrieved images:

- Batch size: 512
- Learning rate: 1e-5
- Warmup steps: 500
- LR schedule: Cosine decay
- L2 weight decay: 0.1

These hyperparameters are used for all our finetuning experiments. We train with an AdamW optimizer, using $\beta_1 = 0.9, \beta_2 = 0.95$.