Mind the Gap Between Synthetic and Real: Probing Transfer Capabilities of Stable Diffusion Images

Leonhard Hennicke[1] **Christian Medeiros Adriano**[1]

Holger Giese[1] **Jan Mathias Koehler**[2] **Lukas Schott**[2]

Abstract

Generative foundation models like Stable Diffusion comprise a diverse spectrum of knowledge in computer vision and are multiple orders of magnitude smaller in storage-size compared with large datasets. Nonetheless, they hold a potential for transfer learning, e.g., via generating data to train student models for downstream tasks, thereby presenting a form of data-free knowledge distillation based on a flexible, implicit dataset compressed into a neural network. However, the resultant student models show a significant drop in accuracy compared with models trained on real data. We investigate possible causes for this drop and focus on the role of the different layers of the student model. By training these layers using either real or synthetic data, we reveal that the drop mainly stems from the model's final layers. Further, we briefly investigate other factors, such as differences in data-normalization between synthetic and real, the impact of data augmentations, texture versus shape learning, and assuming oracle prompts. While we find that some of those factors can have an impact, they are not sufficient to close the performance gap towards real data. Building upon our insights that mainly later layers are responsible for the drop, we investigate the data-efficiency of fine-tuning a synthetically trained model with real data applied to only those last layers. Our results suggest an improved trade-off between the amount of training with real data and the model's accuracy. Our findings contribute to the understanding of the performance gap between training with synthetic and real data while indicating solutions to mitigate the scarcity of labeled real data.

1 Introduction

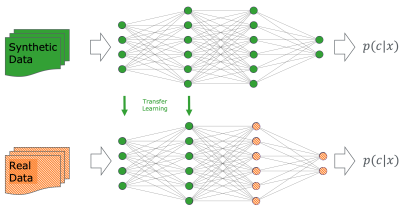
Deep learning has revolutionized computer vision tasks, but training neural networks in various settings incorporates challenges, like the need for large amounts of labeled training data [3] and the computational resources required for deployment [40, 61, 32]. While there are approaches mitigating these challenges; they often lack specificity, e.g., when using general publicly available datasets such as ImageNet [12]; or diversity, e.g., when generating data with *GANs* [18]. Foundation models [6] trained on large-scale diverse internet data offer a potential solution by reducing the need for task-specific training data. In other words, these models implicitly compress knowledge from large datasets, e.g., LAION 5B which amounts to roughly 240TB of storage into a much smaller model $\sim$2-8GB of a StableDiffusion model [48, 41]. However, these models are still large and require expensive hardware for efficient inference. Knowledge distillation [24] is an approach to transfer the knowledge from foundation models to smaller models. Given the variety of different approaches to knowledge distillation [19], we focus on the concept of data-free knowledge distillation [29].

More concretely, we leverage a generative foundation model to create task-specific training images for our student model. We base our paper on the pipeline of Fake It Till You Make It (FITYMI) [46]

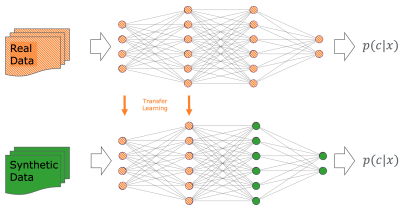
[1]Hasso Plattner Institute, University of Potsdam, `firstname.lastname@hpi.de`

[2]Center for AI, Robert Bosch, `firstname.lastname@de.bosch.com`

Workshop on Machine Learning and Compression @ NeurIPS 2024.



(a) Synthetic Data Transfer Learning.



(b) Real Data Transfer Learning.

Figure 1: Our transfer learning setup for N = 2. N indicates the number of consecutive layers that are transferred from a model that was trained on the respective first dataset, starting from the first layer.

to train models on image data synthesized by StableDiffusion [42]. Surprisingly, while synthetic images are humanly almost indistinguishable from real ones, there is a significant gap in performance (accuracy) when training neural networks only on synthetic data and evaluating on real data. We aim to gain deeper insights into our measured $\sim$23 percentage point (pp) gap in accuracy between training fully on synthetic data [46] vs. real data for ImageNet-100 and analogously 18.4pp on Pets [34].

Our contributions are threefold: (1) we find that the last synthetically trained layers are the ones mostly responsible for the drop in accuracy in the synthetic to real configuration. The implication is that, for instance, within a ResNet-50 dataset the training of all but the last two layers can be done with only synthetic data, resulting in a minor accuracy drop; (2) we show that one could pre-train all but the last two layers merely with synthetic data and fine-tune the remaining layers with a fraction of real data. This is important, because, for instance, one can utilize 1/8 of real data on a pre-trained model while suffering performance drop of only 7pp compared with 20pp (when omitting pre-training on synthesized data); (3) we show that other factors like data-augmentations, oracle prompts, differences in data normalization, and texture versus shape, while potentially helpful in reducing the gap, still do not appear to be sufficient to close the performance gap relative to training on real data.

2 Experiments

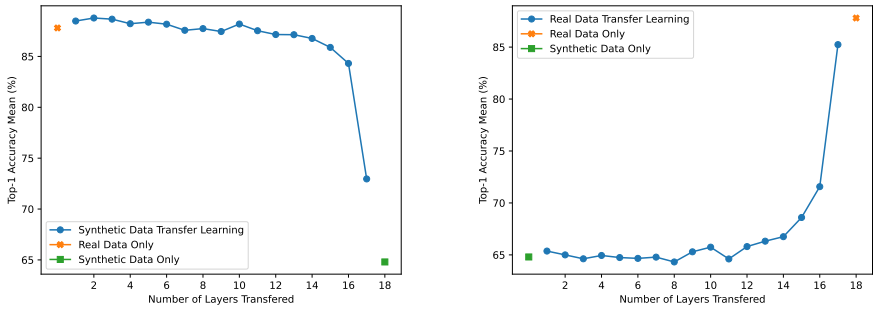
2.1 Experimental Setup

The premise of our experiments is inspired by the findings of the paper Fake It Till You Make It (FITYMI) [46] that employs a StableDiffusion model to generate purely synthetic training data for ImageNet while evaluating on the real ImageNet evaluation (holdout) dataset. For a more detailed description of their and related work, we refer to Appendices A and B. To make our results comparable with their findings, we used the same setup as in FITYMI in all of our experiments, i.e., we generate a purely synthetic ImageNet100 [55], a subset of ImageNet-1K [12], and subsequently train a randomly initialized ResNet50 network for 100 epochs using SGD [44] with a momentum of 0.9 and DINO augmentations with one global and eight local crops, unless stated otherwise. Note that the labels are given the class prompt of the StableDiffusion model. More details are in Appendix F.

For the main experiments, we additionally provide results on StableDiffusion generated images on the classes of the Oxford-IIIT Pet [34] dataset using TinyViT-5M [59], a vision transformer that also incorporates some convolutional layers. The synthetic data for Pets was generated by Stable Diffusion XL [35], using the same methods as in [36], for comparability. We train the student model using the same training pipeline as for ImageNet.

2.2 Layer Importance Experiments

It is a widely accepted intuition that *CNNs* (such as the ResNet-50 we are using) learn representations of more general features (edges, motifs of edges) in the lower layers, while progressively learning more complex combination of motifs towards the deeper layers [27]. We conducted a series of experiments using transfer learning to identify which layers are responsible for the bottleneck in performance when training with merely synthetic data. We pre-trained a CNN on either synthetic or real data, froze the first N layers, and reinitialized the remaining layers for retraining on the respective other dataset. By gradually increasing N, we expected to observe changes in the accuracy of the resulting model, reflecting the quality of features at gradually higher levels of abstraction present in the pre-trained data. These experiments were performed on ImageNet-100 and Pets datasets.



(a) Synthetic Data TL, ImageNet. (b) Real Data TL, ImageNet.

Figure 2: Layer importance experiments for transfer learning (TL) from training on real and synthetic data for ImageNet. The evaluation is always performed on real data. We also plot the results for our baseline models trained solely on synthetic or real data. Note, these models represent the two extremes of this setup and therefore provide an approximation for a lower and upper bound.

In the context of our experiments, layers in the ResNet-50 architecture for ImageNet refer to the bottleneck blocks; for Pets, layers are defined by each convolutional block, each TinyViT block and each downsampling block, as well as the initial patch embedding block, the final classification layer and the preceding normalization layer; amounting to 18 layers total. Running this setup for N = 17 results in a setup equivalent to linear probing, while N = 18 would result in evaluating a completely pre-trained model with all parameters frozen.

The Gap between synthetic and real: The results of this experiment are presented in Figures 2 and 5 for ImageNet and Pets, respectively. We directly see the gap between synthetic and real by comparing models solely trained on real and synthetic data (orange x and green square, 23pp on ImageNet100 and 18.4pp on Pets). The intermediate points (blue line) will be discussed in the next section. Note, that our baselines on Pets are less accurate than on ImageNet-100. We suspect that this is due to the smaller number of samples (37 classes $*$ 100 samples) in the training data compared to ImageNet-100. To show the validity of our training pipeline, we also provide baselines for real and synthetic data using a TinyViT-5M pre-trained on ImageNet-1K, which is commonly done for this dataset [26, 33]. Here, we see accuracies for real data 93.1% and synthetic data 80.2% (not shown in figure), corroborating our pipeline and also confirming the gap. Our baselines without pre-training achieve accuracy levels similar to other state-of-the-art models [49] (when training from scratch).

2.2.1 Transfer Learning

Synthetic Data In our layer-wise transfer experiments, the first N layers are transferred from a model pre-trained on synthetic data with frozen parameters, while only training the remaining layers of the model on real data. We evaluate on real data. We illustrate this setup in Figure 1a for $N = 2$. If the representations learned by the different layers of the model were all to transfer equally well to real data, we would expect the accuracy to gradually drop as we replace more and more of the layers with the ones that were pre-trained on synthetic data and fewer layers are trained on real data. However, this is not the case, suggesting that the quality of the representations that the student model learns from the synthetic data varies with the level of abstraction.

As we can see for ImageNet in Figure 2a, the accuracy varies only little when pre-training layers 1 to 16 with synthetic data. Specifically, the top-1 accuracy only decreases by 3.5 pp, as the layers 1 to 16 are replaced by the ones pre-trained on synthetic data. In fact, the top-1 accuracy even increases up to 1.0 pp as some of the in-between pre-trained layers are added.

A larger decrease of 11.3 pp in top-1 accuracy is visible if the 17th layer (the second to last layer) is replaced by one that is pre-trained on synthetic data - noticeable as a steep drop in Figure 2a. Similarly for Pets in Figure 5a the drop is clearly visible in later layers (>15).

Real Data: To further investigate our findings on ImageNet, we conducted a second series of experiments similar to the previous section. In this setup, the first N layers are transferred from a model pre-trained on real data with frozen parameters, while the remaining layers are reinitialized and trained on synthetic data (illustrated in Figure 1b for N = 2). The evaluation is always performed

Table 1: **Normalization Experiments.** Exact normalization: image normalization using the actual (channel) mean / std. dev. of (R) or (S) data instead of ImageNet values during training and evaluation. Batch norm: batch normalization layers keep updating running batch mean and std. dev. during evaluation.

Experiment	Top-1 Acc. (%)	Top-5 Acc. (%)	Train Loss
Real Data Only (R)	87.8 ±0.1	97.1 ±0.1	0.742 ±0.002
+ Exact Input Normalization	88.5 ±0.1	97.5 ±0.1	0.728 ±0.003
+ Batch Norm in Train Mode	88.5 ±0.1	97.3 ±0.1	0.726 ±0.003
Synth. Data Only (S)	64.8 ±0.1	87.6 ±0.1	0.696 ±0.003
+ Exact Input Normalization	65.9 ±0.1	87.6 ±0.2	0.694 ±0.004
+ Batch Norm in Train Mode	65.2 ±0.2	87.3 ±0.2	0.700 ±0.003

on real validation data. The results of this experiment for ImageNet are presented in Figure 2b. We found that the top-1 accuracy increases by at most 2.0pp from the baseline when adding the pre-trained layers 1 to 14, and even decreases for some in-between layers. However, when adding the pre-trained layers 15 to 17, we observed progressively larger increases in top-1 accuracy of 18.4pp in total, confirming the findings presented in the previous section. This effect is not equally visible for Pets in Figure 5b, where the accuracy increases gradually and we only see a steep increase when also training the final linear classification layer on real data.

2.3 Other Potential Causes of the Accuracy Gap

Normalization We investigate if differences in first and second-order statistics (dataset mean and variance) of the synthetic data that is used during training, and the real data that is used during evaluation, cause the drop in performance. We tested two interventions during the training of the student model. 1) We normalize the input using the actual (channel) mean and standard deviation of the respective dataset (instead of the pre-computed values available for ImageNet) during both training and evaluation. 2) We set the batch normalization layers to train mode during evaluation, causing them to keep updating their running mean and std. dev. instead of using the ones learned during training. We observe a slight improvement in accuracy for these interventions on both real and synthetic data (see Table 1). However, the gap might not to be explainable by exact normalization.

Data Augmentation As shown in FITYMI [46], the type of augmentations can make a difference for models trained on synthetic data, more so than when training on real data. Based on this, we suspected that the gap could be bridged through data augmentation. We test several sophisticated augmentation pipelines, both for models trained on real and on synthetic data. However, the accuracy gap still remains and we could also not replicate the effect of these augmentations w.r.t. leading to a larger improvement when training on synthetic data (as shown in FITYMI) compared to our baseline using no data augmentations. As shown in Table 2, we still observe this effect when comparing the specific augmentations used in FITYMI, i.e., the ones from the PyTorch example [1] and the ones from DINO [7]. However, the effect is less pronounced in our results than in the ones shown in FITYMI. The difference is likely due to a higher guidance scale [25] being set for Stable Diffusion in this specific experiment in FITYMI. Since higher guidance scales lead to less sampling diversity, their data contains less natural variations that could otherwise act similarly to data augmentations.

Local Textures Images generated by Stable Diffusion, often have inconsistencies in finer structures (like human hands) or local textures [41]. This leads us to the question of whether local textures generated by Stable Diffusion are less useful to the student model when generalizing to real data. This would contribute to the accuracy gap, as it has been shown repeatedly, that convolutional neural networks, such as our student models are biased towards local textures [17]. To test this, we generated two new datasets by applying the techniques from Stylized ImageNet [17] to our real and synthetic ImageNet-100 datasets. Stylized ImageNet uses style transfer techniques in combination with paintings to remove local texture cues and in return only retain global shape information. Despite this intervention, the accuracy gap remains (see Table 3).

Prompt Optimization Based on FITYMI, we asked whether the accuracy gap may be caused by unused potential in prompt engineering. To simulate optimal prompts, we applied unCLIP [39] to generate synthetic images with Stable Diffusion based on CLIP [37] embeddings of real ImageNet images as oracle prompts instead of text embeddings as is in our other experiments, i.e., generating

Table 2: **Augmentation Experiments.** The models were trained using different augmentation techniques, as specified in the *Experiment* column. To accurately portray the impact of the different augmentations, the baseline was trained using the FITYMI setup, as in our other experiments, but without the DINO augmentations. Hence, in this table, only the runs labeled "DINO" are equivalent to the full FITYMI setup (our baseline everywhere else).

Experiment	Top-1 Acc. (%)	Top-5 Acc. (%)	Train Loss
Real Data Only (R)	70.1 ±0.1	88.2 ±0.2	0.007 ±0.001
+ AutoAugment	79.7 ±0.1	94.6 ±0.1	0.033 ±0.001
+ PyTorch Example [1]	86.2 ±0.1	96.5 ±0.1	0.308 ±0.003
+ AugMix [22]	86.3 ±0.1	97.2 ±0.1	0.318 ±0.002
+ PixMix [23]	86.5 ±0.1	97.2 ±0.1	0.848 ±0.007
+ DINO [7]	87.8 ±0.1	97.1 ±0.1	0.742 ±0.002
Synth. Data Only (S)	46.9 ±0.2	71.7 ±0.2	0.006 ±0.001
+ AutoAugment	56.2 ±0.2	81.4 ±0.1	0.026 ±0.100
+ PyTorch Example [1]	59.5 ±0.1	83.9 ±0.2	0.261 ±0.002
+ AugMix [22]	60.3 ±0.2	84.5 ±0.2	0.281 ±0.003
+ PixMix [23]	61.8 ±0.2	84.7 ±0.2	0.680 ±0.006
+ DINO [7]	64.8 ±0.1	87.6 ±0.1	0.696 ±0.003

Table 3: **Stylized ImageNet Experiments.** The entire training dataset was modified using the techniques from Stylized ImageNet, removing local texture cues but retaining the global shape information.

Experiment	Top-1 Acc. (%)	Top-5 Acc. (%)	Train Loss
Real Data Only (R)	87.8 ±0.1	97.1 ±0.1	0.742 ±0.002
+ Stylized ImageNet	75.6 ±0.2	92.2 ±0.1	2.099 ±0.005
Synth. Data Only (S)	64.8 ±0.1	87.6 ±0.1	0.696 ±0.003
+ Stylized ImageNet	50.5 ±0.1	77.0 ±0.2	2.053 ±0.006

one synthetic image for each ImageNet-100 image using the real image as an input for unCLIP. As CLIP creates a shared embedding space for text and images, this simulates the perfect text description for each of the real images. In our evaluation, we switch to a synthetic dataset generated by Stable Diffusion v2.1 for a fair comparison as this is the version used by unCLIP. As seen in Table 4, we do not bridge the accuracy gap with this intervention. However, compared to the lower-performing SD v2.1 baseline, we achieve a top-1 accuracy improvement of 26.3 pp, which is larger than the original accuracy gap of 23pp between real data and synthetic data generated by SD v1.4. Potentially, this could allow prompt engineering to bridge the accuracy gap, but, one should be careful drawing conclusions from this experiment. CLIP is trained on a contrastive loss based on cosine similarity between text and image embeddings while unCLIP is trained on CLIP image embeddings. Hence, prompts based on image embeddings could induce data leakage, as they provide more information than even a perfect text prompt would do, e.g., by having both the perfect angle and magnitude.

Table 4: **Prompt optimization unCLIP Experiment.** We compare student models trained on various synthetic datasets generated by two Stable Diffusion (SD) versions and unCLIP (based on Stable Diffusion v2.1, with the real ImageNet-100 images as prompts).

Experiment	Top-1 Acc. (%)	Top-5 Acc. (%)	Train Loss
unCLIP, (SD v2.1)	66.6 ±0.3	90.8 ±0.1	0.690 ±0.004
Synth. Data Only (SD v2.1)	40.3 ±0.3	66.3 ±0.4	0.511 ±0.002
Synth. Data Only (SD v1.4)	64.8 ±0.1	87.6 ±0.1	0.696 ±0.003

Closer supervision to mitigate shortcuts One potential reason why our generalization from purely synthetic data is limited might be shortcut learning [16]. Spurious correlations with our labels could lead to learning features that generalize poorly. Here, we investigate whether a closer supervision improves generalization from synthetic to real data. We leverage the CLIP model [38] for additional supervision, as it usually has good zero-shot performance. Thus, similarly to Popp et al.[36], we extend our cross entropy loss and introduce another loss to match the high dimensional OpenCLIP image model embeddings (specifically the model: *laion2b_s34b_b82k_augreg_soup*) on the StableDiffusion

Table 5: **Closer supervision with Teachers.** We compare student models trained on our synthetic ImageNet-100 dataset. We train our models to match CLIP embeddings. For the model being trained with both L2 loss and classification loss, we implement two independent heads, one learning to imitate clip embeddings and one learning classification labels, both based on the original ResNet-50 embedding layer. "Clf." stands for classification.

Experiment	Top-1 Acc. (%)	Top-5 Acc. (%)	Train Loss
Clf. Loss Only (S)	64.8 ±0.1	87.6 ±0.1	0.696 ±0.003
L2 Loss Only (S)	66.2 ±0.2	86.9 ±0.2	14.8 ±0.006
L2 Loss + Clf. Loss (S)	65.5 ±0.1	87.9 ±0.1	15.6 ±0.007

v1.4 generated data. Thus our overall loss function is

$$L = L_{CE}(f_\psi(z_\theta(x), y) + ||g_\phi(z_\theta(x)) - \textit{CLIP}(x)||_2, \tag{1}$$

z_θ is a ResNet-50 architecture as in all previous experiments but without the last layer, *CLIP* is the CLIP image encoder, and f_ψ, and g_ϕ are linear layers to project into the label space or CLIP embedding space, respectively. L_{CE} / $||.||_2$ are cross-entropy / L_2 loss. We train three variants, one with only using the classification loss, the second one with both losses and lastly, only with the L_2 loss. For the latter variant the labels are computed analogously to CLIP by computing a cosine similarity with text embeddings of class names [38]. All other trainings parameters are kept the same. The results in Table 5 show that adding closer supervision with two teachers (StableDiffusion and OpenCLIP) can marginally improve the performance but does not close the gap.

2.4 Data-Reduction Experiments

So far, we saw that there is a significant gap in performance of models trained on real data compared to synthetic data. Here, we investigate how much of the performance can be recovered by using real data for fine-tuning the last layers of an otherwise synthetically trained model. When using 1/8th of the entire real training data, we can boost our model performance from 64.8% to 80.1%, significantly closing the gap towards the model trained on the whole real training data 87.8%. Thus synthetic data extracted from a StableDiffusion models seems to be effective for representation learning as also found in [54]. Further details and experiments are shown in Appendix C.

3 Conclusion and Future Work

We investigated the accuracy gap between models trained on synthetic data versus real data. Looking into the role of data normalization, batch normalization, and data augmentations for synthetic versus real data, we identified improvements that explain part of the observed accuracy gap. Our results indicate that the final layers of the student model play a significant role in bridging this gap. When pre-training all but the last two layers using synthetic data, we observed only a minor drop in accuracy. Furthermore, we demonstrated that pre-training the majority of the layers with synthetic data and fine-tuning the remaining layers with a fraction of real data resulted in improved performance compared to models trained merely on a subset of real data. These results suggest leveraging synthetic data to mitigate a lack of labeled real training data. Our findings point to promising new avenues for future research: Firstly, a deeper exploration of the specific features learned by the final layers could shed light on their contribution to the observed accuracy gaps. Additionally, investigating various ratios of synthetic and real data during fine-tuning may reveal better trade-offs, potentially incorporating active learning to select the most informative labeled data. Lastly, an investigation into how different generative foundation models impact student model performance would provide a comprehensive understanding of the potential of synthetic data in transfer learning.

Acknowledgments and Disclosure of Funding

This work was supported in part by Bosch Center for Artificial Intelligence and the Hasso Plattner-Institute. We thank Claudia Blaiotta, Bastian Bischoff, Niclas Popp, Anna Khoreva, and Jan H. Metzen for their discussions and feedback on the manuscript. The Bosch Group is carbon neutral. Administration, manufacturing and research activities do no longer leave a carbon footprint. This also includes GPU clusters on which most experiments have been performed.

References

[1] AI, M.: Pytorch examples. http://github.com/pytorch/examples (2013)

[2] Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., Fleet, D.J.: Synthetic data from diffusion models improves imagenet classification. arXiv:2304.08466 (2023)

[3] Bhardwaj, K., Suda, N., Marculescu, R.: Edgeal: A vision for deep learning in the iot era. IEEE Design & Test **38**(4), 37–43 (2019)

[4] Binici, K., Aggarwal, S., Pham, N.T., Leman, K., Mitra, T.: Robust and resource-efficient data-free knowledge distillation by generative pseudo replay. In: AAAI Conf. on Artificial Intelligence. vol. 36, pp. 6089–6096 (2022)

[5] Binici, K., Pham, N.T., Mitra, T., Leman, K.: Preventing catastrophic forgetting and distribution mismatch in knowledge distillation via synthetic data. In: CVF Conf. on Applications of Computer Vision. pp. 663–671 (2022)

[6] Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., et al.: On the opportunities and risks of foundation models. arXiv:2108.07258 (2021)

[7] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging Properties in Self-Supervised Vision Transformers. In: ICCV Intl. Conf. on Computer Vision (2021)

[8] Chang, X., Yang, Y., Xiang, T., Hospedales, T.M.: Disjoint label space transfer learning with common factorised space. In: AAAI Conf. on Artificial Intelligence. vol. 33, pp. 3288–3295 (2019)

[9] Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: CVPR Conf. on Computer Vision and Pattern Recognition. pp. 3558–3568 (2021)

[10] Chen, H., Wang, Y., Xu, C., Yang, Z., Liu, C., Shi, B., Xu, C., Xu, C., Tian, Q.: Data-free learning of student networks. In: CVF Intl. Conf. on Computer Vision. pp. 3514–3522 (2019)

[11] D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M.D., et al.: Underspecification presents challenges for credibility in modern machine learning. Journal of Machine Learning Research **23**(226), 1–61 (2022)

[12] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet a large-scale hierarchical image database. In: CVPR Conf. on computer vision and pattern recognition. pp. 248–255 (2009)

[13] Desai, K., Kaul, G., Aysola, Z., Johnson, J.: RedCaps: Web-curated image-text data created by the people, for the people. arXiv:2111.11431 (2021)

[14] Do, K., Le, T.H., Nguyen, D., Nguyen, D., Harikumar, H., Tran, T., Rana, S., Venkatesh, S.: Momentum adversarial distillation: Handling large distribution shifts in data-free knowledge distillation. Advances in Neural Information Processing Systems **35**, 10055–10067 (2022)

[15] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929 (2020)

[16] Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11), 665–673 (2020)

[17] Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv:1811.12231 (2018)

[18] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020), publisher: ACM New York, NY, USA

[19] Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. Intl. Journal of Computer Vision **129**(6), 1789–1819 (2021)

[20] Hammoud, H.A.A.K., Itani, H., Pizzati, F., Torr, P., Bibi, A., Ghanem, B.: Synthclip: Are we ready for a fully synthetic clip training? arXiv:2402.01832 (2024)

[21] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR Conf. on Computer Vision and Pattern Recognition. pp. 770–778 (2016)

[22] Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. ICLR Intl. Conf. on Learning Representations (2020)

[23] Hendrycks, D., Zou, A., Mazeika, M., Tang, L., Li, B., Song, D., Steinhardt, J.: PixMix: Dreamlike Pictures Comprehensively Improve Safety Measures. In: CVPR Conf. on Computer Vision and Pattern Recognition (2022)

[24] Hinton, G., Vinyals, O., Dean, J., et al.: Distilling the knowledge in a neural network. arXiv:1503.02531 (2015)

[25] Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv:2207.12598 (2022)

[26] Hong, J., Huang, K., Liang, H.N., Wang, X., Zhang, R.: Fine-grained image classification with object-part model. In: Advances in Brain Inspired Cognitive Systems: 10th International Conference, BICS 2019, Guangzhou, China, July 13–14, 2019, Proceedings 10. pp. 233–243. Springer (2020)

[27] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553) (2015)

[28] Liang, K.J., Li, C., Wang, G., Carin, L.: Generative adversarial network training is a continual learning problem. arXiv:1811.11083 (2018)

[29] Lopes, R.G., Fenu, S., Starner, T.: Data-free knowledge distillation for deep neural networks. arXiv:1710.07535 (2017)

[30] McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of learning and motivation, vol. 24, pp. 109–165. Elsevier (1989)

[31] Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)

[32] Miotto, R., Wang, F., Wang, S., Jiang, X., Dudley, J.T.: Deep learning for healthcare: review, opportunities and challenges. Briefings in bioinformatics **19**(6), 1236–1246 (2018)

[33] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)

[34] Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3498–3505. IEEE (2012)

[35] Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)

[36] Popp, N., Metzen, J.H., Hein, M.: Zero-shot distillation for image encoders: How to make effective use of synthetic data (2024)

[37] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML Intl. Conf. on Machine Learning. pp. 8748–8763 (2021)

[38] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML Intl. Conf. on Machine Learning. pp. 8748–8763 (2021)

[39] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv:2204.06125 **1**(2), 3 (2022)

[40] Ravi, D., Wong, C., Lo, B., Yang, G.Z.: A deep learning approach to on-node sensor data analytics for mobile or wearable devices. IEEE Journal of Biomedical and Health Informatics **21**(1), 56–64 (2016)

[41] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR Conf. on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)

[42] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR Conf. on Computer Vision and Pattern Recognition. pp. 10684–10695 (2022)

[43] Roth, K., Kim, J.M., Koepke, A., Vinyals, O., Schmid, C., Akata, Z.: Waffling around for performance: Visual classification with random words and broad concepts. In: CVF Intl. Conf. on Computer Vision. pp. 15746–15757 (2023)

[44] Ruder, S.: An overview of gradient descent optimization algorithms. arXiv:1609.04747 (2016)

[45] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., et al.: Photorealistic text-to-image diffusion models with deep language understanding. arXiv:2205.11487 (2022)

[46] Sarıyıldız, M.B., Alahari, K., Larlus, D., Kalantidis, Y.: Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In: CVPR Conf. on Computer Vision and Pattern Recognition. pp. 8011–8021 (2023)

[47] Sariyildiz, M.B., Kalantidis, Y., Alahari, K., Larlus, D.: No reason for no supervision: Improved generalization in supervised models. arXiv preprint arXiv:2206.15369 (2022)

[48] Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., 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)

[49] Schuler, J.P.S., Romani, S., Abdel-Nasser, M., Rashwan, H., Puig, D.: Grouped pointwise convolutions reduce parameters in convolutional neural networks. In: Mendel. vol. 28, pp. 23–31 (2022)

[50] Shadish, W.R., Cook, T.D., Campbell, D.T.: Experimental and quasi-experimental designs for generalized causal inference. Houghton, Mifflin and Company (2002)

[51] Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL Association for Computational Linguistics. pp. 2556–2565 (2018)

[52] Stöckl, A.: Evaluating a synthetic image dataset generated with stable diffusion. arXiv:2211.01777 (2022)

[53] Thanh-Tung, H., Tran, T.: Catastrophic forgetting and mode collapse in gans. In: IJCNN Intl. Joint Conf. on Neural Networks. pp. 1–10. IEEE (2020)

[54] Tian, Y., Fan, L., Isola, P., Chang, H., Krishnan, D.: Stablerep: Synthetic images from text-to-image models make strong visual representation learners. Advances in Neural Information Processing Systems **36** (2024)

[55] Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: ECCV European Computer Vision Conf. pp. 776–794. Springer (2020)

[56] Wang, L., Zhang, X., Su, H., Zhu, J.: A comprehensive survey of continual learning: Theory, method and application. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)

[57] Wang, X., Lin, B., Liu, D., Xu, C.: Efficient transfer learning in diffusion models via adversarial noise (2023)

[58] Wang, Z., Yang, E., Shen, L., Huang, H.: A comprehensive survey of forgetting in deep learning beyond continual learning. arXiv:2307.09218 (2023)

[59] Wu, K., Zhang, J., Peng, H., Liu, M., Xiao, B., Fu, J., Yuan, L.: Tinyvit: Fast pretraining distillation for small vision transformers. In: European conference on computer vision. pp. 68–85. Springer (2022)

[60] Yamaguchi, S., Kanai, S., Kumagai, A., Chijiwa, D., Kashima, H.: Transfer learning with pre-trained conditional generative models. arXiv:2204.12833 (2022)

[61] Zhang, T., Gao, C., Ma, L., Lyu, M., Kim, M.: An empirical study of common challenges in developing deep learning applications. In: ISSRE Intl. Symp. on Software Reliability Engineering. pp. 104–115 (2019)

[62] Zoph, B., Ghiasi, G., Lin, T.Y., Cui, Y., Liu, H., Cubuk, E.D., Le, Q.: Rethinking pre-training and self-training. Advances in neural information processing systems **33**, 3833–3845 (2020)

Supplementary Material

A Preliminaries

Definition 1 - Stable Diffusion a.k.a. latent diffusion model (*LDM*) is derived from a type of generative model [42] that employs equally weighted denoising autoencoders to predict the denoised version of input data, with an objective function minimizing the difference between the added and the predicted noise.

$$\mathcal{L}_{LDM} = \mathbb{E}_{\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[||\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))||_2^2\right] \tag{2}$$

where τ_θ is the specialized encoder for the additional value guiding the generation process, which is trained jointly with ϵ_θ and y is the additional value. By shifting computations to a perceptually analogous but lower-dimensional domain through an auto-encoder, Stable Diffusion reduces the computational complexity because it can focus on the semantically significant aspects of data.

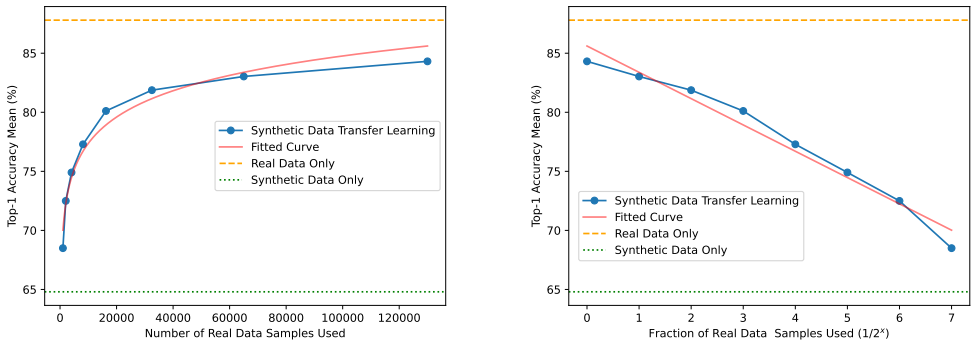
Definition 2 - Data-Free Knowledge Distillation precludes the access to the original training data for transferring knowledge from a teacher model to a student [29]. Instead, one leverages either the architecture or the learned parameters of the teacher model to distill knowledge to the student. This process often involves methods such as feature matching, activation mimicking, or utilizing generative models to generate synthetic data for distillation (our approach).

B The State of the Art of Data-Free Knowledge Distillation

Current works exhibit a significant gap on ImageNet-1k [12] top-1 accuracy if only trained on synthetic data compared to real data. For instance, StableRep [54] has 34.9% zero-shot accuracy compared to 73.7% linear probing with a ViT-B/16 [15]. As a point of reference, we note that CLIP, which has been trained on real data [38], has 68.6% zero-shot accuracy on a ViT-B/16 and 59.6% on a ResNet50 [21]. In contrast, SynthCLIP [20] only provides a zero-shot accuracy of 30.5% with a ViT-B/16 backbone. Lastly, also [46] only achieves 42.9% with a ResNet50. From our perspective, this gap in accuracy is quite puzzling, as for humans, synthetic images from models like StableDiffusion look almost indistinguishable from real ones.

We base our paper on the pipeline of Fake It Till You Make It (FITYMI) [46]. Analogously, to their approach training models on image data generated by StableDiffusion [42]. We adapt the FITYMI setup for our experiments to make our results directly comparable. We re-compute our replication of the paper's best-performing model for ImageNet-100 in all our result tables. Note that due to random seeds, we observe minor fluctuations ($< 1\%$ compared to the results in Table 1 in [46] with a prompt scheme consisting of class and definition $p_c =$"c, d_c", and guidance scale $= 2$). Furthermore, we focus on using the ImageNet-100 dataset instead of ImageNet-1K due to resource constraints, similar to most experiments in FITYMI. Most importantly, this reduced dataset still exhibits a fairly low accuracy of just 28.4% if it is trained in a naive training fashion, e.g., without augmentations. To investigate parts of this gap, the FITYMI paper already covers some ablations on ImageNet-100 like variations of prompts, augmentations, and the guidance scale that each significantly impact the performance. Especially, introducing augmentations improves the baseline performance by 14.8 pp. Similarly, tuning the guidance scale improves the performance by 20 pp. Note that both factors have been tuned in isolation and the benefits might be less than their sum if combined. While the aforementioned tricks increase the performance, a large portion of the gap remains, e.g., a vanilla training on the real ImageNet-100 data leads to 87.4% top-1 accuracy compared to the best data-free variant from [46] achieving 64.8%. Thus, we extend the ablations by testing out further factors and investigating where in the network discrepancies between real and synthetic data originate.

Similarly, to FITYMI, StableRep [54] presents a pipeline learning representations solely based on synthetic images. They use captions from the datasets CC3M [51], CC12M [9] or RedCaps [13], respectively, to generate image data with Stable Diffusion v1.5 and then train the model on the generated data in an unsupervised manner to learn the representations used for linear probing and zero-shot classification. While the synthetic ImageNet-1K only includes less than 1.3 million samples, StableRep uses 10 million and 11.6 mio. samples from CC12M and RedCaps, respectively. For CC3M only 2.7 mio. samples are used. Also, StableRep trains a ViT-B/16 (~ 86 mio. trainable params.) for 35 epochs, whereas we train a ResNet50 (~ 25 mio. trainable params.) for 100 epochs.



(a) Linear Scale (Read Right to Left)

(b) Logarithmic Scale (Read Left to Right)

Figure 3: **Data Reduction Experiments.** The first 16 layers are trained on synthetic data and frozen. The last two layers are fine-tuned with a different number of real data samples. The fitted curves are derived from the plotted top-1 accuracy on the respective scales via curve fitting, using least squares polynomial fitting on a 1st-degree polynomial ($f(x) = -2.23log(x) + 85.61$).

Lastly, Azizi et al. [2] use Imagen [45] instead of StableDiffusion to generate synthetic training data for image classification. They achieve a top-1 accuracy of 69.24% on ImageNet using a ResNet50 trained for 90 epochs; however, they achieve this by fine-tuning ImageNet data, resulting in their approach not being completely data-free.

C Data-Reduction Experiments

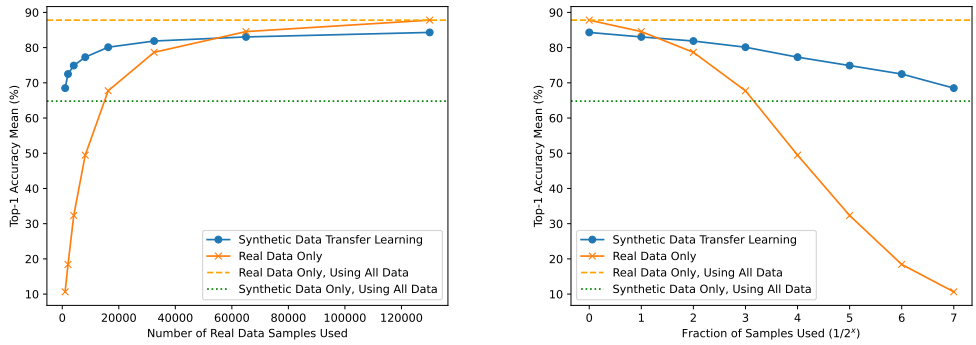
Based on the results of our layer importance experiments, we suspect that a model with only the last layers trained on real data (and the earlier layers were already pre-trained on synthetic data) would require less data to achieve a similar accuracy as a baseline model trained fully on real data. To test this hypothesis, we pre-train the first 16 layers of our model on synthetic data, i.e., $N = 16$, and then reinitialize and retrain the remaining layers on a randomly drawn, reduced number of samples from the real ImageNet-100 dataset, effectively reducing the amount of real data necessary for training. Assuming our intuition is correct, we should observe that the accuracy remains within a similar range as the baseline model accuracy, as we decrease the amount of real training data used.

The results of these experiments are presented in Figure 3. We decrease the amount of real training data by halving it for each consecutive experiment. Even though we reduce the real training exponentially, the top-1 accuracy only decreases by 4.2pp after the third iteration (when using 1/8th of the entire real training data). The fitted curve suggests that the top-1 accuracy decreases logarithmically, as the amount of real training data is reduced. Additionally, we show in Figure 4 a series of reductions in real training data for a randomly initialized model, the drop in accuracy is steeper than for the models with frozen layers pre-trained on synthetic data, suggesting that synthetic pre-training can be applied to mitigate the lowered accuracy caused by a lack of real data.

D Additional Definitions

Definition 3 - Knowledge Distillation consists of transferring knowledge from a larger teacher model to a smaller student model while maintaining performance and reducing computational costs [24]. Initially, the teacher model is trained on a large dataset to achieve high accuracy. Then, the student model learns to mimic the teacher's behavior using both the original dataset and the teacher's soft targets, which contain probabilistic distribution insights. A Kullback-Leibler divergence loss function guides the student model to align its output distribution with that of the teacher model. equation and line below can be removed to save space

$$KL(\hat{y}, y) = y \cdot \log \frac{y}{\hat{y}} \tag{3}$$



(a) Linear Scale (Read Right to Left)

(b) Logarithmic Scale (Read Left to Right)

Figure 4: **Training on a reduced real dataset.** Top-1 accuracy of reducing the amount of real training data with (blue, same as Figure 3) and without (orange) synthetic data transfer learning.

where $\hat{y}$ and y are the predicted and true labels respectively.

Definition 4 - Foundation Models are deep neural networks (*DNN*) models with very large numbers of trainable parameters, which, thus, require orders of magnitude more data than traditional *DNN* models. Besides their prediction power, foundation models [6] are more adaptable to downstream tasks.

D.1 Connection to Transfer Learning

Transfer Learning - *TL*. Reusing knowledge from pre-trained models via TL requires that data from pre-trained tasks (source) and new ones (target) have some overlap. Unfortunately, this assumption is unrealistic in domains subject to data sparsity [8]. To bridge this gap between source-target labels, one can combine similarity-guided training and adaptive adversarial noise selection [57]. Another approach is to apply self-supervision methods, for instance, Yamaguchi et al. [60] investigated a two-stage process that (1) synthesizes target samples by conditioning on source generative model and, then, (2) labels these new samples via self-supervised learning methods. Albeit *TL* has shown to outperform knowledge distillation [60], one still has to cope with the threat of "negative transfer", i.e., learning wrong concepts or forgetting correct ones [62].

Catastrophic forgetting - *CF*. Re-training models needs to cope with the fundamental phenomenon of catastrophic forgetting [30], i.e., neural networks when trained on different tasks tend to "forget" (lose accuracy) in previously learned tasks. This phenomenon is not limited to sequential learning setups (e.g., continual learning, reinforcement learning, domain adaptation). Fine-tuning of foundation and generative models are also affected [58], for instance, in generative adversarial networks - *GANs* trained continuously [56]. Some of the mitigation to *CF* involve momentum and gradient penalty [53] or changes in the discriminator component of the *GAN* [28]. Meanwhile, in *GANs* continuously trained in the context of data-free knowledge distillation (*DFKD*) [10], mitigation of *CF* by preserving memory buffers for past samples [5], remembering previous distributions for generative replay [4], and keeping an exponential moving average to minimize the impact of abrupt shifts in the distribution [14]. Our approach is complementary to these approaches, as we combine an architecture-based approach (fixing parameters, i.e., layers) with synthetic or real data for pre-training and then fine-tuning. Catastrophic forgetting could happen in various steps of the pipeline, for instance, the synthetic images cause the student model to forget features that were successfully learned by the teacher model. While freezing layers might mitigate this loss, it might still happen in the retrained layers, because one cannot guarantee that all fine-tuned knowledge is new knowledge.

E Threats to Validity

We discuss possible threats to the validity [50] of our evidences and claims. **Construct validity** threats consist of concepts misrepresenting the object of study, for instance, assuming that accuracy represents the performance of the classifier when the samples are imbalanced (i.e., unequal number of negative and positive instances) or when false negatives and false positives have distinct levels of importance (i.e., costs, risks/safety) for downstream tasks (e.g., planning). We mitigated this validity threat by working only with balanced datasets. Although sound, this assumption might be too strong for real-world settings, where number of instances, relative costs/risks, and error rates are not homogeneous or comparable across classes [52]. **Internal validity** threats relate to assumptions that can be invalidated by alternative explanations for the measured effects or lack thereof, e.g., hidden confounders or selection bias induced by data leakage. We mitigated this by controlling certain parameters (e.g., freezing vs. reinitializing layers), taking the last 5 training epochs, and evaluating the top-1/top-5 outcomes. However, as we were unable to control all the parameters, there might be vestiges of confounding, e.g., (1) how the data augmentation might have produced selection biases (certain types of images benefiting more from the augmentations) or (2) the competing effects of oracle prompts and additional information the image embeddings used to simulate them may contain. A promising avenue is to further probe with the simulated prompts, as the positive effects of prompt engineering are corroborated by similar settings like WaffleCLIP [43]. **External validity** threats correspond to configurations (e.g., hyperparameters) that hinder the generalization/reproduction of similar accuracy gains in future experiments (e.g., under different methods/domains). We mitigated this risk by performing various sensitivity analyses, for instance, varying the intake order of real vs. synthetic data in pre-training and the number of frozen layers. Albeit extensive, our evaluations were not exhaustive. Further promising work is to study if different architecture types and domains/classes can induce the phenomenon of underspecification [11], where models with equivalent performance on holdout sets still show distinct outcomes after being deployed.

F Additional Details on Hyperparameters

F.1 Generation of Synthetic Data

For most experiments, we use ImageNet-100 [55], a subset of ImageNet-1K [12]. This is the same subset of classes that is used in FITYMI. We use either the original data from ImageNet-100 or a synthetic version of this dataset with 1300 images generated per class. The synthetic images were generated using Stable Diffusion 1.4 unless stated otherwise. This was done using 50 diffusion steps and a guidance scale [25] of 2.0. The images were generated sized 512 x 384 and using the WordNet [31] based definition prompts from FITYMI (i.e., "<class name>, <WordNet definition of class name>"). Generating a full ImageNet-100 dataset with this setup takes over 73 single-GPU hours.

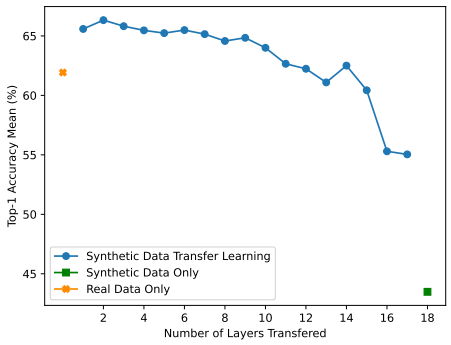
F.2 Model Training and Evaluation

For training our models, we linearly increased the learning rate up to 0.1 for the first 10 epochs and then decayed it on a cosine schedule. In addition, temperature scaling with L2-normalized weights was used on the final classification layer. Our training pipeline is based on the implementation of [47], which is publicly available and adapted for our classification task. For further details like exact cropping parameters are provided in the supplemental material.

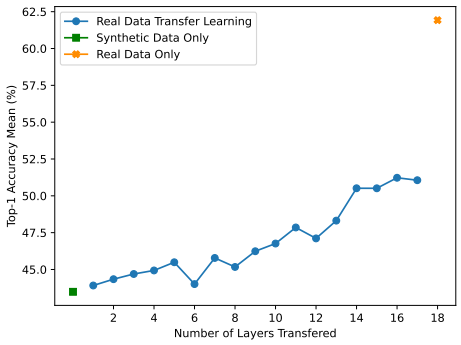
The exact cosine schedule used for learning rate decay, given as a formula to calculate the learning rate for each each consecutive batch while the learning rate is decayed using this schedule, taken from [47]:

$$LR(i) = 0.000001 + 0.5 * (LR_{base} - 0.000001) * (1 + cos(\pi * i/i_{max})) \tag{4}$$

where i_{max} is the number of iterations the learning rate is decayed, i.e., the number of batches per epoch times the number of epochs the schedule is applied to, i is the current iteration and LR_{base} is the base learning rate of 0.1 multiplied by the batch size divided by 256. We use a batch size of 64 and 32 for ResNet-50 and TinyViT respectively.



(a) Synthetic Data TL, Pets.



(b) Real Data TL, Pets.

Figure 5: Results of the layer importance experiments using transfer learning (TL) from training on real and synthetic data for Pets. The evaluation is always performed on real data. We also plot the results for our baseline models trained on synthetic and real data only, as these present the two extremes of this experiment setup and therefore provide an approximate lower and upper bound.

Table 6: **Synthetic Data Transfer Learning on ImageNet-100.** This table provides the concrete results for Figure 2a. Using the first N layers of the model pre-trained on synthetic ImageNet-100 data with frozen weights, the remaining layers were initialized randomly and trained on real ImageNet-100 data. We report the mean and standard deviation on real ImageNet-100 validation data of the last 5 epochs for top-1 and top-5 accuracy.

Frozen Layers	Top-1 Acc. (%)	Top-5 Acc. (%)	Train Loss
None - Synth. Data Only (S)	64.8 ±0.1	87.6 ±0.1	0.696 ±0.003
None - Real Data Only (R)	87.8 ±0.1	97.1 ±0.1	0.742 ±0.002
+ L. 0-1 from (S)	88.5 ±0.2	97.4 ±0.1	0.713 ±0.003
+ L. 0-2 from (S)	88.8 ±0.1	98.0 ±0.1	0.747 ±0.003
+ L. 0-3 from (S)	88.7 ±0.1	97.4 ±0.1	0.753 ±0.003
+ L. 0-4 from (S)	88.2 ±0.1	97.5 ±0.1	0.752 ±0.002
+ L. 0-5 from (S)	88.4 ±0.2	97.7 ±0.1	0.762 ±0.004
+ L. 0-6 from (S)	88.2 ±0.1	97.6 ±0.1	0.762 ±0.004
+ L. 0-7 from (S)	87.6 ±0.1	97.5 ±0.1	0.763 ±0.003
+ L. 0-8 from (S)	87.7 ±0.1	97.8 ±0.1	0.769 ±0.004
+ L. 0-9 from (S)	87.4 ±0.1	97.3 ±0.1	0.786 ±0.001
+ L. 0-10 from (S)	88.2 ±0.1	97.4 ±0.1	0.787 ±0.001
+ L. 0-11 from (S)	87.5 ±0.1	97.6 ±0.1	0.797 ±0.003
+ L. 0-12 from (S)	87.2 ±0.1	97.0 ±0.1	0.808 ±0.003
+ L. 0-13 from (S)	87.1 ±0.2	97.1 ±0.1	0.824 ±0.003
+ L. 0-14 from (S)	86.8 ±0.1	96.7 ±0.1	0.822 ±0.002
+ L. 0-15 from (S)	85.9 ±0.2	96.8 ±0.1	1.010 ±0.002
+ L. 0-16 from (S)	84.3 ±0.2	96.0 ±0.1	1.277 ±0.002
+ L. 0-17 from (S)	73.0 ±0.2	92.1 ±0.1	2.104 ±0.004

For multicropping, the relative crop sizes are randomly sampled from the following intervals: $[0.25, 1.0]$ for global crops and $[0.05, 0.25]$ for local crops. For our experiments, we report the mean and standard deviation on real ImageNet-100 validation data of the last 5 epochs for top-1 and top-5 accuracy.

G Supplemental Tables for Sections 2.2 and C

Table 7: **Real Data Transfer Learning on ImageNet-100.** This table provides the concrete results for Figure 2b. Using the first N layers of the model pre-trained on real ImageNet-100 data with frozen weights, the remaining layers were initialized randomly and trained on synthetic ImageNet-100 data. We report the mean and standard deviation on real ImageNet-100 validation data of the last 5 epochs for top-1 and top-5 accuracy.

Frozen Layers	Top-1 Acc. (%)	Top-5 Acc. (%)	Train Loss
None - Real Data Only (R)	87.8 ±0.1	97.1 ±0.1	0.742 ±0.002
None - Synth. Data Only (S)	64.8 ±0.1	87.6 ±0.1	0.696 ±0.003
+ L. 0-1 from (R)	65.4 ±0.3	87.4 ±0.2	0.713 ±0.003
+ L. 0-2 from (R)	65.0 ±0.1	87.4 ±0.1	0.723 ±0.003
+ L. 0-3 from (R)	64.6 ±0.1	87.2 ±0.1	0.724 ±0.002
+ L. 0-4 from (R)	64.9 ±0.2	87.8 ±0.2	0.721 ±0.004
+ L. 0-5 from (R)	64.7 ±0.2	86.4 ±0.1	0.727 ±0.004
+ L. 0-6 from (R)	64.7 ±0.2	86.6 ±0.2	0.732 ±0.002
+ L. 0-7 from (R)	64.8 ±0.1	87.6 ±0.2	0.738 ±0.003
+ L. 0-8 from (R)	64.3 ±0.2	86.6 ±0.2	0.740 ±0.003
+ L. 0-9 from (R)	65.3 ±0.1	86.7 ±0.2	0.746 ±0.004
+ L. 0-10 from (R)	65.7 ±0.2	88.3 ±0.2	0.748 ±0.002
+ L. 0-11 from (R)	64.6 ±0.3	87.0 ±0.2	0.698 ±0.004
+ L. 0-12 from (R)	65.8 ±0.4	87.1 ±0.1	0.703 ±0.002
+ L. 0-13 from (R)	66.3 ±0.4	87.6 ±0.2	0.712 ±0.001
+ L. 0-14 from (R)	66.8 ±0.4	87.7 ±0.1	0.727 ±0.002
+ L. 0-15 from (R)	68.6 ±0.3	89.1 ±0.2	0.889 ±0.002
+ L. 0-16 from (R)	71.6 ±0.2	90.5 ±0.2	1.069 ±0.002
+ L. 0-17 from (R)	85.2 ±0.3	95.6 ±0.2	1.925 ±0.003

Table 8: **Synthetic Data Transfer Learning on Oxford-IIIT Pet.** This table provides the concrete results for Figure 2c. Using the first N layers of the model pre-trained on synthetic Oxford-IIIT Pet data with frozen weights, the remaining layers were initialized randomly and trained on real Oxford-IIIT Pet data. We report the mean and standard deviation on real Oxford-IIIT Pet validation data of the last 5 epochs for top-1 and top-5 accuracy.

Frozen Layers	Top-1 Acc. (%)	Top-5 Acc. (%)	Train Loss
None - Synth. Data Only (S)	43.5 ±0.5	76.2 ±0.4	0.006 ±0.006
None - Real Data Only (R)	61.9 ±0.1	89.1 ±0.1	0.009 ±0.009
+ L. 0-1 from (S)	65.6 ±0.1	90.6 ±0.1	0.006 ±0.006
+ L. 0-2 from (S)	66.3 ±0.1	90.9 ±0.1	0.005 ±0.005
+ L. 0-3 from (S)	65.8 ±0.2	90.8 ±0.1	0.005 ±0.005
+ L. 0-4 from (S)	65.5 ±0.1	90.8 ±0.1	0.009 ±0.009
+ L. 0-5 from (S)	65.2 ±0.1	90.7 ±0.1	0.006 ±0.006
+ L. 0-6 from (S)	65.5 ±0.2	90.0 ±0.1	0.004 ±0.004
+ L. 0-7 from (S)	65.2 ±0.1	90.6 ±0.1	0.003 ±0.003
+ L. 0-8 from (S)	64.6 ±0.1	89.6 ±0.1	0.004 ±0.004
+ L. 0-9 from (S)	64.8 ±0.1	90.0 ±0.1	0.004 ±0.004
+ L. 0-10 from (S)	64.0 ±0.0	89.6 ±0.1	0.003 ±0.003
+ L. 0-11 from (S)	62.7 ±0.1	89.8 ±0.1	0.002 ±0.002
+ L. 0-12 from (S)	62.2 ±0.1	88.9 ±0.1	0.003 ±0.003
+ L. 0-13 from (S)	61.1 ±0.1	88.4 ±0.1	0.004 ±0.004
+ L. 0-14 from (S)	62.5 ±0.1	89.4 ±0.1	0.005 ±0.005
+ L. 0-15 from (S)	60.4 ±0.2	88.1 ±0.1	0.005 ±0.005
+ L. 0-16 from (S)	55.3 ±0.2	84.7 ±0.1	0.008 ±0.008
+ L. 0-17 from (S)	55.0 ±0.1	84.8 ±0.1	0.008 ±0.008

Table 9: **Real Data Transfer Learning on Oxford-IIIT Pet.** This table provides the concrete results for Figure 2d. Using the first N layers of the model pre-trained on real Oxford-IIIT Pet data with frozen weights, the remaining layers were initialized randomly and trained on synthetic IOxford-IIIT Pet data. We report the mean and standard deviation on real Oxford-IIIT Pet validation data of the last 5 epochs for top-1 and top-5 accuracy.

Frozen Layers	Top-1 Acc. (%)	Top-5 Acc. (%)	Train Loss
None - Real Data Only (R)	61.9 ±0.1	89.1 ±0.1	0.009 ±0.009
None - Synth. Data Only (S)	43.5 ±0.5	76.2 ±0.4	0.006 ±0.006
+ L. 0-1 from (S)	43.9 ±0.5	77.2 ±0.3	0.004 ±0.004
+ L. 0-2 from (S)	44.3 ±0.4	77.0 ±0.3	0.004 ±0.004
+ L. 0-3 from (S)	44.7 ±0.4	77.0 ±0.2	0.005 ±0.005
+ L. 0-4 from (S)	44.9 ±0.4	78.4 ±0.4	0.005 ±0.005
+ L. 0-5 from (S)	45.5 ±0.4	78.2 ±0.4	0.005 ±0.005
+ L. 0-6 from (S)	44.0 ±0.5	78.0 ±0.4	0.004 ±0.004
+ L. 0-7 from (S)	45.8 ±0.4	80.2 ±0.2	0.005 ±0.005
+ L. 0-8 from (S)	45.2 ±0.4	79.1 ±0.3	0.007 ±0.007
+ L. 0-9 from (S)	46.2 ±0.4	80.7 ±0.2	0.007 ±0.007
+ L. 0-10 from (S)	46.8 ±0.4	80.0 ±0.4	0.007 ±0.007
+ L. 0-11 from (S)	47.9 ±0.2	80.9 ±0.3	0.006 ±0.006
+ L. 0-12 from (S)	47.1 ±0.4	80.6 ±0.3	0.007 ±0.007
+ L. 0-13 from (S)	48.3 ±0.4	81.1 ±0.3	0.006 ±0.006
+ L. 0-14 from (S)	50.5 ±0.4	83.6 ±0.2	0.007 ±0.007
+ L. 0-15 from (S)	50.5 ±0.3	83.0 ±0.2	0.005 ±0.005
+ L. 0-16 from (S)	51.2 ±0.1	84.4 ±0.2	0.004 ±0.004
+ L. 0-17 from (S)	51.1 ±0.2	84.4 ±0.1	0.004 ±0.004

Table 10: **Data Reduction Experiments.** This table provides the concrete data points for Figure 4. The models in Table 10a (except for the baselines) were trained with the first 16 layers of the model pre-trained on synthetic ImageNet-100 data with frozen weights and the remaining layers being initialized randomly and trained on real ImageNet-100 data. The real training data used to train the last two layers was reduced to the fraction specified in the table using random sampling. In Table 10b we show the results of the same data reduction applied to models that are trained on real data only. We report the mean and standard deviation on real ImageNet-100 validation data of the last 5 epochs for top-1 and top-5 accuracy.

(a) Synthetic Data Transfer Learning

Experiment	Top-1 Acc. (%)	Top-5 Acc. (%)	Train Loss
Real Data Only (R)	87.8 ±0.1	97.1 ±0.1	0.742 ±0.002
Synth. Data Only (S)	64.8 ±0.1	87.6 ±0.1	0.696 ±0.003
+ 1/1 of real data used	84.3 ±0.2	96.0 ±0.1	1.277 ±0.002
+ 1/2 of real data used	83.0 ±0.1	96.2 ±0.1	1.313 ±0.004
+ 1/4 of real data used	81.9 ±0.2	95.2 ±0.1	1.398 ±0.005
+ 1/8 of real data used	80.1 ±0.2	94.7 ±0.1	1.477 ±0.006
+ 1/16 of real data used	77.3 ±0.3	93.8 ±0.1	1.533 ±0.007
+ 1/32 of real data used	74.9 ±0.3	93.0 ±0.1	1.558 ±0.006
+ 1/64 of real data used	72.5 ±0.2	91.6 ±0.1	1.616 ±0.021
+ 1/128 of real data used	68.5 ±0.2	90.8 ±0.2	1.516 ±0.039

(b) Real Data Only

Experiment	Top-1 Acc. (%)	Top-5 Acc. (%)	Train Loss
1/1 of real data used	87.8 ±0.1	97.1 ±0.1	0.742 ±0.002
1/2 of real data used	84.6 ±0.1	99.0 ±0.1	0.822 ±0.002
1/4 of real data used	78.7 ±0.1	94.1 ±0.1	1.124 ±0.002
1/8 of real data used	67.7 ±0.1	89.0 ±0.2	1.529 ±0.002
1/16 of real data used	49.5 ±0.2	78.2 ±0.2	2.179 ±0.004
1/32 of real data used	32.3 ±0.1	61.9 ±0.1	2.808 ±0.005
1/64 of real data used	18.4 ±0.1	43.8 ±0.2	3.254 ±0.009
1/128 of real data used	10.7 ±0.1	28.6 ±0.2	3.454 ±0.005