Combating distribution shift in self-supervised learning with test-time adaptation

Aditya Mehrotra and Aviraj Newatia

Abstract—We explore methods to adapt self-supervised models at test time on distribution shifts. Specifically, we focus on methods which finetune the model with constraints such as low rank updates, fisher information and naive unconstrained optimization. We evaluate our method on Camelyon-17 and Cifar10/Cifar10.1.

Index Terms—Vision Transformers, Self-Supervised Learning, Representation Learning, Fine-tuning, Test Time Adaptation

1 Introduction

Deep learning architectures benefit from learning rich representations about the distribution of their training data. In most conventional supervised learning cases, we use ground truth labels or predictions about input data in order to train neural networks. However, the lack of availability of labeled data in some settings makes this challenging. Self-supervised representation learning emerged as a paradigm to train the intermediate representations of a deep learning system without ground-truth data. Once representations for the input data distribution were learned, a separate classification head (such as a linear probe) would be trained on the limited labeled data to produce a system that could interpret the learned representations.

However, these systems can suffer from distribution shift at test time. Distribution shifts occur when the distribution of the data that a deep learning system is deployed on differs from the distribution of the data that it was trained on. These can occur for a number of reasons, such as a change in the hospital that the data was collected from. Distribution shifts can cause harmful effects to learned systems, and studying methods of preventing distribution shifts and adapting trained models to observed distribution shifts is a rich field of study [1]. Existing works in test-time adaptation that target self-supervised learning methods assume access to labeled data about the test distribution. This is a significant limitation of current test-time adaptation methods in self-supervised learning settings, as in deployment it is not necessary that we have access to labeled samples from the test distribution. For instance, in the context of medical imaging, distribution shift can occur when deploying a model in a hospital who's data was not in its training set. This can occur for reasons as simple as the choice of dye in histopathology slides [2]. On deployment we wish to update the representations of the neural network to consolidate the causal features from the test distribution data to the category of the input data they belong to.

In this work we tackle this problem of performing test-time adaptation of self-supervised models without access to ground-truth labels for test data. This severely restricted case limits potential approach vectors due to the inability to train using error signals from ground-truth labels. We present an evaluation of naive, Low-Rank Adaptation [3], and Elastic Weight Consolidation [4] fine-tuning techniques as approaches to adapting trained representations to a new test distribution.

2 Related Works

2.1 Self-Supervised Learning

Self Supervised learning is a class of methods that train robust feature extractors from images in the absence of labels. These methods often involve teacher and student networks, with a knowledge distillation loss. These methods can be applied to all sorts of vision backbones such as CNNs and Vision Transformers. Once trained, these feature extractors are able to accurately classify the distribution they are trained on via simple linear probes. Self-Supervised training is often used as a method to Pre-Train feature extractors. [5] [6]

2.2 Test-time Adaptation

Test-Time adaptation is a class of methods which modify a model to adapt to incoming test distributions. There are a variety of setups and approaches to this problem. The survey paper we cited covers these in more detail [1].

In our work, we assume no access to labels at and both train and test time. Related work in this area involve updating batchnorm/layernorm statistics, or matching first order statistics of the representation learner to match between train/test distributions [7].

Our work differs in the sense that we try and fine-tune the model directly, and change the base set of weights.

• *Aditya Mehrotra and Aviraj Newatia are with the Department of Computer Science, University of Toronto, Toronto, ON, Canada and the Vector Institute, Toronto, ON, Canada.*

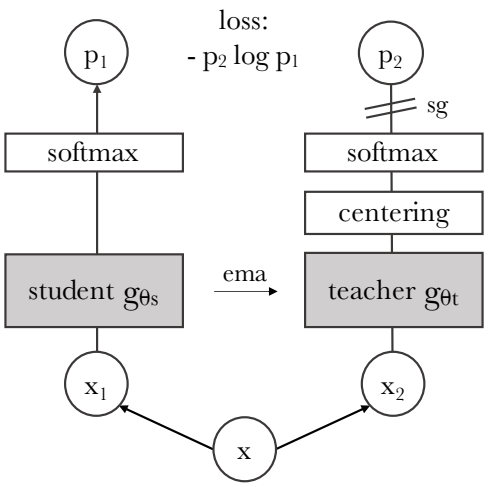

Fig. 1. DINO

3 METHODOLOGY

We propose using fine-tuning methods in order to adapt self-supervised representations to the target domain of the test distribution. Fine-tuning methods have shown promise in transfer learning for taking large pre-trained methods and adapting them for new tasks. This enables trained neural networks to change to model the distribution represented by the fine-tuning task.

3.1 Self-Supervised Pretraining with DINO

In this paper, we use the the DINO [5] training framework for pretraining and finetuning. The DINO framework is illustrated in Fig. 1.

There are two vision transformers, a "student" and "teacher". Both networks are initialized the same way, and the teacher is a direct copy of the student at the start of pretraining. Then, the teacher and student are fed augmented versions of the same image x denoted as x_1, x_2, and we get representations for both of them. We then pass them through a softmax and align their representations via a cross entropy loss. Once the loss is computed, gradient is back-propagated to the student network and the teacher parameters are updated with an EMA of the student parameters.

The intuition is that the teacher network's logits provide a moving target that the student needs to "match". Since we augment x, the representations become extremely robust to image corruptions such as blurs, solarized blurs and rotations/flips. Ultimately, we receive a strong representation learner from the teacher that can get high accuracies with simple classifiers such as KNN or linear probes.

3.2 Finetuning

Naive fine-tuning is typically performed by inserting a classification head onto the self-supervised pre-trained models, and by using error signal to update the internal representations of the model to extract the important causal features relevant to the downstream fine-tuning task. This is shown in Figure 2.

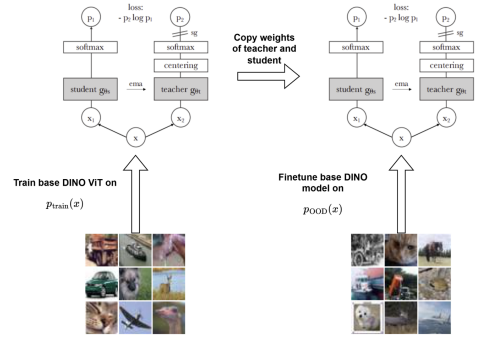

Fig. 2. Naive finetuning of a ViT on out of distribution

Considering the lack of test labels in our setting, we instead apply fine-tuning through self-supervised training. Using the original training methodology of the pre-trained representation network, we perform several epochs of training using the original self-supervised framework on data samples from the test distribution in order to enrich the representations that the network learns for them. Considering that these samples contain whichever potential distribution shift is observed, by consolidating their representations with those learned for the training distribution, the network should discard spurious correlation factors which constitute the distribution shift and identify the features that are causal and remain consistent with what was seen during training in the train distribution.

3.3 Low-Rank Adaptation

Low-Rank adaptation (LoRA) methods are a powerful and efficient method of fine-tuning deep neural networks which has shown promise in the study of large transformers and language models. Particularly in the study of vision transfomers [8], LoRA has been used to improve robustness through adversarial fine-tuning.

LoRA methods enable fine-tuning of large pre-trained methods without changing the underlying pre-trained parameters, and instead learn an adaptation matrix which is used to slightly perturb network activations towards the distribution modelled by the fine-tuning task. This adaptation matrix is learned by learning two matrices projecting the input data of a layer down to a low dimensional representation and then back up to the dimension of the output of the layer.

Performing LoRA fine-tuning on samples from the test distribution at deployment time thus naturally translates to a useful method of test-time adaptation. By training on test distribution samples, the LoRA modules learn slight perturbations to perform to the inference pipeline of test distribution samples to bring them "in-distribution" of the representations learned by the neural network.

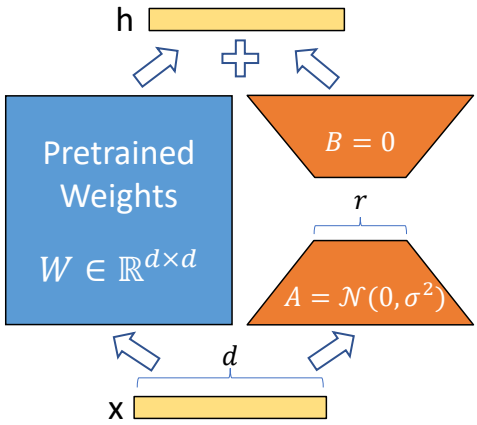

Fig. 3. LoRA

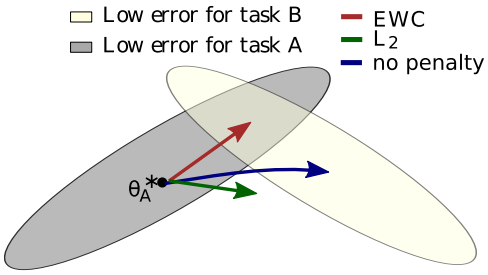

Fig. 4. Elastic Weight Consolidation (EWC)

3.4 Elastic Weight Consolidation

Elastic Weight Consolidation (EWC) is a method from the continual learning community used to mitigate the phenomena of catastrophic forgetting [4]. Catastrophic forgetting is the phenomena in continual learning settings where deep neural networks tend to "forget" training samples encountered earlier in training.

Since the goal of test-time adaptation is to expand the distribution of input data that the pre-trained network can reliably, robustly, and causally represent - it is an important consideration to ensure that the fine-tuning procedure does not overfit the network to the test distribution, and forget instances from the training distribution. To ensure this, EWC adds a regularization term to the loss function for training based on the Fisher Information Matrix [9]. This induces a per-parameter regularization restricting the movement of the parameter in weight-space to a surface on which it still represents the same information about the posterior distribution of the learned model after pre-training. This enables the behaviour of the neural network to consolidate the data points seen later in training, which in the case of test-time adaptation are the test distribution samples, without forgetting the important information about the training distribution. This process improves retention of causally important features regarding both the training and test distributions, while shifting the parameters to represent what is common between the two distributions, which intuitively should be the task relevant information.

4 Experimental Results

4.1 Setup

We evaluate Naive, LoRA, and EWC fine-tuning for test-time adaptation on pretrained vision transformers [10] trained in a self-supervised manner using the DiNO algorithm [5]. We focus on smaller vision transformers, particularly the family of TinyViTs [11] and SmallViTs.

We employ two datasets that exhibit distribution shifts in image-classification tasks for different reasons. First, we train on CIFAR10 [12] and evaluate on CIFAR10.1 [13] [14]. Second, we evaluate on the Camelyon17 [2] dataset from the WILDS [15] distribution shift benchmark. In this dataset we use the `harmful` and `not harmful` dataset splits to display distribution shift.

To evaluate our adapted models we attach two types of classifiers to the pre-trained backbone network. Firstly, we employ a linear probe, trained on the test distribution labels. The accuracy of this linear probe gauges the separability of the representation space, and is standard in self-supervised learning literature. Secondly, we use a k-Nearest-Neighbor (KNN) [16] classification head, trained on the training set, to measure the consolidation of input data samples. A high accuracy on a KNN classification head indicates that the representations of samples from both the training distribution and test distribution that belong to the same class lie close together in the representation space. This implies that the learned representations are good causal feature extractors.

4.2 Camelyon17

We pre-train a `ViT_Tiny` model on the training set of the Camelyon17 dataset from WILDS. We employ the DiNO framework, training for 500 epochs with a learning rate capped at 0.0005 and 10 warmup epochs, and the AdamW optimizer [17]. We evaluate the performance of the pre-trained network combined with a KNN classifier and a trained linear probe on the `train`, `harmful` and `not harmful` dataset splits which represent in-distribution, out-of-distribution and potentially harmful distribution shift, and out-of-distribution and likely not-harmful distribution shift in input images. Both of the out-of-distribution images are collected from hospitals other than those from which the training samples were collected.

For test-time adaptation we perform 50 epochs of Naive fine-tuning, LoRA fine-tuning, and EWC fine-tuning using the same learning rate. We evaluate 5 checkpoints for each fine-tuning method (10, 20, 30, 40, and 50 epochs). We selected LoRA hyperparameters using a grid search, settling on a rank of [2,4,8].

Using a KNN classification head, we report the classification accuracy using both 10 and 20 nearest neighbours. The best results comparing Naive fine-tuning, EWC, and LoRA are presented in Table 1.

We observe that during the fine-tuning procedure, the neural network exhibits a degradation in performance on

TABLE 1
Camelyon17 classification results with 10NN.
We show Top1 accuracy.

Method	Train	Val	Harmful	Not-Harmful
Base	**99.98**	**99.74**	73.19	78.90
Naive	99.83	99.55	73.28	79.27
EWC	73.31	79.26	73.31	79.26
LoRA	98.68	98.13	**93.64**	**84.44**

TABLE 2
Camelyon17 classification results with 10NN across LoRA Ranks.
We show Top1 accuracy.

LoRA Rank	Train	Val	Harmful	Not-Harmful
2	98.56	97.73	88.08	83.90
4	**98.68**	**98.13**	**93.64**	**84.44**
8	98.05	97.05	89.03	79.39

the training set, as shown in Figure 5. Fine-tuning with EWC also leads to small decreases in performance of the model on in-distribution dataset splits. This was an interesting outcome, as the increases in accuracy observed on the out-of-distribution data was marginal. Considering that Naive fine-tuning did not offer much in the way of performance gains on the out-of-distribution data, this behavior is not surprising, as it promotes retaining accuracy on the data it has already been trained on.

As visible in Table 2, the LoRA adaptation also produced a decrease in accuracy on the in-distribution data, such as the `Train` and `Val` dataset splits. However, LoRA adapters were able to increase performance on the out-of-distribution `Harmful` and `Not Harmful` dataset splits. We observe that the performance increase observed on the `Harmful` split is significantly greater than the increase observed on the `Not Harmful` split. This is interesting because the base model performed worse on the `Harmful` split to begin with. We notice that performance gains on the `Harmful` split increase as LoRA rank increases to a significant point whereas performance on the `Not Harmful` split increases marginally. This is particularly interesting as the `Not Harmful` split was used for the test-time adaptation procedure. We observe that LoRA rank 4 yields the best performance on test-time fine-tuning. Increasing the rank to 8 degrades performance on the both out-of-distribution and in-distribution splits of the dataset.

We also observe in Table 3 that performance on out-of-distribution data peaks at 20 epochs of LoRA fine-tuning with some performance degradation on in-distribution data. However, we do note that after this point, the changes in performance on both in and out of distribution dataset splits are marginal implying that the test and train distributions are somewhat consolidated in representation space after few epochs of LoRA fine-tuning. This suggests that this method of test-time adaptation is efficient, likely due to the flexibility of freshly initialised LoRA adapter weights. This would allow the first few epochs of fine-tuning on test-distribution data to heavily bias the LoRA weights, thoroughly adapting the network to the new distribution.

TABLE 3
LoRA Rank 4 Fine-tuning on Camelyon17

# Epochs	Train	Val	Harmful	Not Harmful
10 Epochs	98.40	97.71	93.00	85.55
20 Epochs	98.48	97.8	**94.06**	**84.91**
30 Epochs	98.60	98.10	91.32	84.10
40 Epochs	98.67	98.08	93.44	84.39
50 Epochs	**98.68**	**98.13**	93.64	84.44

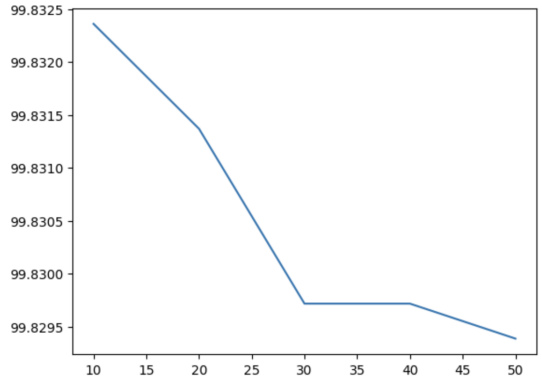

Fig. 5. KNN Classifier Accuracy on Training Set in Naive Fine-Tuning on Camelyon17

4.3 CIFAR-10

CIFAR10.1 is a set of 2000 images belonging to the CIFAR10 classes that have been adversarially sampled from TinyImagenet [18] to induce the distribution shift. In this setting, we pre-train a SmallViT on CIFAR10 for 500 epochs with a learning rate capped at 0.0005 and 10 warmup epochs and evaluate its performance on CIFAR10.1 using a KNN and Linear probing. We refer to CIFAR 10.1 as "test".

In Table 4, we see numerical evidence in distribution shift in the SSL representations via KNN classifiers. The KNN fit to the train set performs well on the validation set and performs much worse on the test set.

TABLE 4
CIFAR10 classification results.
Val and Test show Top1 accuracy for 10NN and 20NN classifiers.

Method	Val		Test	
	10NN	20NN	10NN	20NN
Base	87.16	87.19	76.70	76.75

In Table 5, we try simply fine-tuning our base DINO network pretrained on CIFAR 10, on our test set. We find that performance degrades both on the train and test sets when using linear probes on checkpoints.

When we use LoRA, we find that the model performance degrades on both sets a lot slower compared to naive finetuning. This is shown in Table 6, for linear probing once again.

TABLE 5
Iterative Fine-tuning Results on CIFAR (each iteration is 10 epochs)

Method	Val	Test
Base	87.16	75.2
First iteration of finetuning	87.12	75.0
Second iteration of finetuning	86.54	74.8
Third iteration of finetuning	85.96	74.63
Fourth iteration of finetuning	85.40	74.13

TABLE 6
CIFAR10 Classification Results with LoRA

Method	Val	Test
Base	87.1	75.2
Rank-4 LoRA	86.5	75.0
Rank-8 LoRA	85.8	74.7

Finally, with EWC, we get larger changes in loss compared to LoRA, but the same trend of degradation on both sets. These results are illustrated in Table 7. We specifically see that the higher rank of LoRA, the more performance degrades on both distributions. This is intuitive, as a higher rank allows for more degrees of freedom to destroy the original model.

Ultimately, we see that each finetuning method results in worse performance on both distributions.

5 Discussion

This work evaluates how effective fine-tuning methods are at mitigating the impact of distribution shifts in deep learning systems at test time. We find that naive fine-tuning, fine-tuning with EWC, and fine-tuning with LoRA all lead to a reduction in model performance on data from the training distribution. Simultaneously, we observe that in most cases, these methods do not have a measurable positive impact on performance on the test distribution. The exception to this case was LoRA fine-tuning on the Camelyon17 histopathology datasets, in which we observed up to a 20% increase in classification accuracy on harmful distribution shifts.

We note that this result was not observed when LoRA was used to fine-tune models for CIFAR10. We consider this to be an artifact of the number of classes and the complexity of these datasets, as well as the fact that the CIFAR10.1 distribution shift dataset was adversarially created - and thus may include compound distribution shifts whereas Camelyon17 does not have these traits. We also believe that the lack of performance gains given by EWC and Naive fine-tuning are caused by the small size of the test fine-tuning dataset and that CIFAR has many more classes compared to camelyon, which is binary. This is exacerbated by the fact that transformers require large amounts of data to train, and thus are likely less flexible to changing their representative distribution.

However, we note that this work provides a baseline in the study of test-time adaptation in the strong setting of solely having unlabeled samples from the test distribution. This is a practical problem, and one that is not studied extensively in existing literature.

TABLE 7
Iterative EWC iterations on CIFAR (each iteration is 10 epochs)

Method	Val	Test
Base	87.16	75.2
First iteration of finetuning	87.16	75.1
Second iteration of finetuning	87.0	74.9
Third iteration of finetuning	86.9	74.8
Fourth iteration of finetuning	86.7	74.6

TABLE 8
EWC Camelyon17 classification results.

# Epochs	Harmful		Not-Harmful	
	10NN	20NN	10NN	20NN
10 Epochs	73.21	74.26	79.02	79.36
20 Epochs	73.27	74.28	79.19	79.44
30 Epochs	73.30	74.32	79.23	79.51
40 Epochs	73.24	74.28	**79.27**	**79.56**
50 Epochs	**73.31**	**74.33**	79.26	**79.56**

Possible extensions to this work are a combination of EWC and LoRA fine-tuning methods, as this may allow the flexibilty of LoRA and the robustness of EWC to yield a well-adapted model. Additionally, we encourage exploration into a negative weighting of the EWC regularization term, which may potentially encourage forgetting of part of the training distribution to allow for adaptation of the test distribution. Further work may include explorations into sparsity and modifications to the self-supervised learning process.

Acknowledgments

The authors would like to thank Professor Aviad Levis for his supervision, and Professor David Lindell. We would like to acknowledge Professor Rahul Krishnan and the Department of Computer Science for the compute resources used to run experiments.

References

[1] J. Liang, R. He, and T. Tan, "A comprehensive survey on test-time adaptation under distribution shifts," 2023. [Online]. Available: https://arxiv.org/abs/2303.15361

[2] P. Bándi, O. Geessink, Q. Manson, M. Van Dijk, M. Balkenhol, M. Hermsen, B. Ehteshami Bejnordi, B. Lee, K. Paeng, A. Zhong, Q. Li, F. G. Zanjani, S. Zinger, K. Fukuta, D. Komura, V. Ovtcharov, S. Cheng, S. Zeng, J. Thagaard, A. B. Dahl, H. Lin, H. Chen, L. Jacobsson, M. Hedlund, M. Çetin, E. Halıcı, H. Jackson, R. Chen, F. Both, J. Franke, H. Küsters-Vandevelde, W. Vreuls, P. Bult, B. van Ginneken, J. van der Laak, and G. Litjens, "From detection of individual metastases to classification of lymph node status at the patient level: The camelyon17 challenge," *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 550–560, 2019.

[3] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

TABLE 9
Naive FT Camelyon17 classification results.

# Epochs	Harmful		Not-Harmful	
	10NN	20NN	10NN	20NN
10 Epochs	73.21	74.24	79.03	79.36
20 Epochs	73.23	74.26	79.18	79.46
30 Epochs	**73.25**	74.27	79.25	79.53
40 Epochs	73.24	74.28	**79.27**	79.56
50 Epochs	73.24	**74.29**	**79.27**	**79.57**

[4] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, p. 3521–3526, Mar. 2017. [Online]. Available: http://dx.doi.org/10.1073/pnas.1611835114

[5] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021. [Online]. Available: https://arxiv.org/abs/2104.14294

[6] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2020. [Online]. Available: https://arxiv.org/abs/1911.05722

[7] Y. Liu, P. Kothari, B. van Delft, B. Bellot-Gurlet, T. Mordan, and A. Alahi, "Ttt++: When does self-supervised test-time training fail or thrive?" in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 21 808–21 820. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/b618c3210e934362ac261db280128c22-Paper.pdf

[8] Z. Yuan, J. Zhang, and S. Shan, "Fulllora-at: Efficiently boosting the robustness of pretrained vision transformers," 2024. [Online]. Available: https://arxiv.org/abs/2401.01752

[9] A. Ly, M. Marsman, J. Verhagen, R. Grasman, and E.-J. Wagenmakers, "A tutorial on fisher information," 2017. [Online]. Available: https://arxiv.org/abs/1705.01064

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[11] K. Wu, J. Zhang, H. Peng, M. Liu, B. Xiao, J. Fu, and L. Yuan, "Tinyvit: Fast pretraining distillation for small vision transformers," 2022. [Online]. Available: https://arxiv.org/abs/2207.10666

[12] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:18268744

[13] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do cifar-10 classifiers generalize to cifar-10?" 2018, https://arxiv.org/abs/1806.00451.

[14] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[15] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. A. Earnshaw, I. S. Haque, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang, "Wilds: A benchmark of in-the-wild distribution shifts," 2021. [Online]. Available: https://arxiv.org/abs/2012.07421

[16] P. A. Devijver and J. Kittler, "Pattern recognition : a statistical approach," in *Pattern recognition : a statistical approach*, 1982. [Online]. Available: https://api.semanticscholar.org/CorpusID:61074523

[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019. [Online]. Available: https://arxiv.org/abs/1711.05101

[18] Y. Le and X. S. Yang, "Tiny imagenet visual recognition challenge," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:16664790