The Hidden Costs on Distributional Shifts when Fine-tuning Joint Text-Image Encoders and Redemptions

Andrew Geng
University of Wisconsin-Madison*
Madison, WI
ageng@wisc.edu

Pin-Yu Chen
IBM Research
Yorktown Heights, NY
pin-yu.chen@ibm.com

Abstract

When considering the performance of a pre-trained model, transferred to a down-stream task, it is important to account for both the model's generalization and detection capabilities on out-of-distribution (OOD) samples. In this paper, we unveil the hidden costs of intrusive fine-tuning techniques. Specifically, we show that (1) common fine-tuning techniques can distort not only the representations necessary for domain generalization (OOD Generalization), but also the representations necessary for detecting semantic shifted OOD samples (OOD Detection). Additionally, we propose a novel reprogramming approach called `reprogrammer` *which attempts to mitigate these degradations found in common fine-tuning techniques. We show that our* `reprogrammer` *method is (2) less intrusive and can lead to better retention of pre-training representation. Subsequently, by maintaining more pre-training representation, we have found that* `reprogrammer` *performs better holistically when accounting for the in-distribution (ID), OOD Generalization, and OOD Detection performances of the down-stream model.*

1. Introduction

There are many fundamental hurdles obstructing researchers from improving OOD Generalization and OOD Detection performances in deep learning networks. These challenges can range from difficulties in encapsulating covariant (domain) shifts, to overconfidence when predicting on semantically shifted samples [24, 27, 33]. One framework of training deep learning models, that has shown impressive OOD Generalization and OOD Detection performance, is large text-image supervised pre-trained models [16, 28, 29].

However, recently it has been made more aware that common transfer learning techniques can distort the strong representations learned during pre-training, resulting in a degradation in specifically the model's OOD Generalization performance [1, 20, 38]. In this paper, we present evidence showing that common transfer learning methods, such as linear-probing (optimizing just the classification head) and full fine-tuning (optimizing all model parameters), each have their own strengths and hidden costs in terms of ID and OOD performances. More specifically, we present evidence showing that these common transfer learning techniques can degrade not only OOD Generalization but also OOD Detection performance. This subsequently beckons the question *can we build a different transfer learning technique that is less intrusive and more robust to both covariate and semantically shifted OOD samples*?

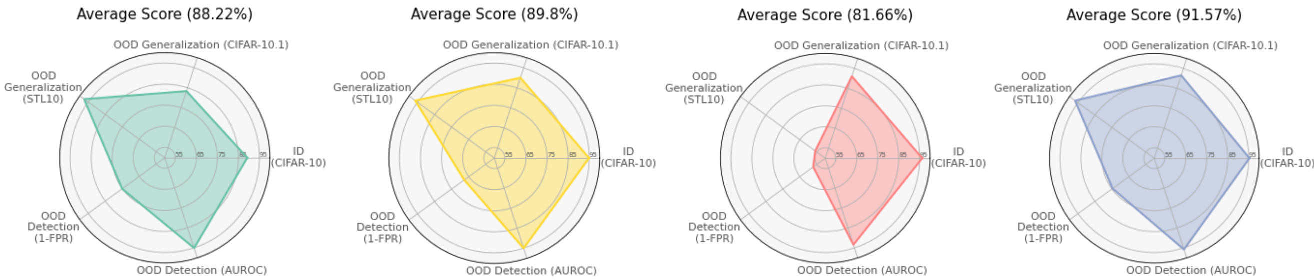
We tackle this question by exploring and altering a different paradigm of transfer learning called *model reprogramming* [3]. By leveraging and altering some key components from *model reprogramming*, we show that it is possible to reprogram a text-image pre-trained model to a down-stream ID task. We also show that, due to the less intrusive nature of reprogramming, our method is better able to maintain pre-training representation, subsequently leading to better OOD Generalization and OOD Detection performances. Formally, we propose `reprogrammer`, a novel reprogramming approach that leverages two different modalities of *model reprogramming* to reprogram both the image encoder and the text encoder simultaneously.

We demonstrate the hidden costs and trade-offs of common fine-tuning techniques and `reprogrammer` in Figure 1. Additionally, to our knowledge, we are the first to take this step in applying *model reprogramming* techniques to multi-modal text-image encoder models.

2. Methodology

In this section, we first start by introducing the image reprogrammer module before moving to the text reprogrammer module. After which we will present the full `reprogrammer` transfer learning technique. We also pro-

*Work done while A.G was working at IBM.



(a) Zero-shot (b) Linear-probing (c) Full Fine-tuning (d) Reprogrammer

Figure 1. Radar charts showcasing the trade-offs in ID, OOD Generalization, and OOD Detection performances between Zero-shot, Linear-probing, Full Fine-tuning, and `Reprogrammer` techniques. All results are averaged based on the specified CIFAR benchmarks, as described in Section 3.1, and to quantify the cost-performance trade-offs, we report the average score normalized across all metrics.

vide more `reprogrammer` details in Appendix C.

2.1. Image Reprogrammer

Consider just the CLIP image encoder $f : I \rightarrow \mathbb{R}^{b \times k}$ where b is the input image batch size and $k = 512$ is the CLIP feature size. To apply reprogramming, we leverage the commonly used adversarial program first described by Elsayed et al [10], to which we define as the reprogramming function ψ. The reprogramming function ψ is applied to the input image pre-forward pass through the CLIP image encoder f. Critically, the reprogramming function ψ is not specific to any singular input image, rather ψ will be consistently applied to all images.

Formally, we define our reprogramming function ψ as:

$$\psi(X) = \mathcal{U}(X) + \tanh(W \odot M) \quad (1)$$

where $\mathcal{U}$ denotes an image up-sampling then zero-padding function, $W \in \mathbb{R}^{d \times d \times 3}$ is the image reprogrammer parameters that is to be learned, d is the size of CLIP's input width and height, $\odot$ denotes the Hadamard product, and M is a binary masking matrix. Specifically, we define the binary masking matrix M as 0 for positions where we wish to implant the original image, and 1 for positions that we choose to reprogram.

2.2. Text Reprogrammer

Now we consider the CLIP text encoder $g : S \rightarrow \mathbb{R}^{b \times k}$ where b is the input text batch size and $k = 512$ is the CLIP feature size. Additionally, we define our text input s as a sequence of tokens $s = \{s_1, ..., s_{|s|}\}$ where s_i is the vocabulary index of the i^{th} token in the vocabulary list V_S. To apply reprogramming to a text input, we leverage and alter a version of the adversarial program as first described by Neekhara et al [26].

Formally, we define our text reprogramming function as $\Phi_{\theta,b}$ where $\Phi_{\theta,b}$ is a simple look-up embedding and bias on the tokens $\{s_i\}$ that can be parameterized by the learnable embedding tensor θ and the bias parameter b. Specifically, we define our $\theta \in \mathbb{R}^{|V_S| \times d}$ and $b \in \mathbb{R}^d$ where our default vocabulary size is $|V_S| = 49408$, which is the expected vocabulary size for the CLIP text encoder. Similarly, as with all reprogramming functions, the text reprogramming function is not specific to any singular text input, rather $\Phi_{\theta,b}$ will be consistently applied to all text inputs.

2.3. CLIP Model Reprogrammer

Finally, to train our given image and text reprogramming functions ψ and $\Phi_{\theta,b}$, we define our training objective as:

$$W^*, \theta^*, b^* = \underset{W,\theta,b}{\operatorname{argmax}} \left(\operatorname{sim}(f(\psi_W(x)),\ g(\Phi_{\theta,b}(s))) \right) \quad (2)$$

where (x, s) is an image and caption pair obtained from our training set D_{in}, f and g are the CLIP image and text encoders respectively, sim is the cosine-similarity function, and W, θ, b are the learnable parameters encapsulating our reprogramming functions $\psi_W, \Phi_{\theta,b}$. In practice, rather than directly optimizing for cosine similarity, we follow closely with the optimization schema of a symmetric cross entropy loss as was implemented in CLIP pre-training [29].

After tuning our `reprogrammer` parameters W, θ, b we perform classification during inference time, on an input image $\hat{x}$ with m class labels $C = \{c_1, ..., c_m\}$, similar to that of zero-shot CLIP. Specifically, we make a prediction y through:

$$y = \underset{i}{\operatorname{argmax}}(\operatorname{sim}(f(\psi_{W^*}(\hat{x})), g(\Phi_{\theta^*,b^*}(s_i)) \quad (3)$$

where s_i is the class-wise captions such that s_i = "a photo of a $\{c_i\}$" and ψ_{W^*} and Φ_{θ^*,b^*} are our learned reprogramming functions parameterized by W^*, θ^*, and b^*.

3. Experiments

In this section, we first describe our experimental setup for OOD Generalization and OOD Detection in Section 3.1, before evaluating our `reprogrammer` method against other common transfer learning techniques in Section 3.2.

Table 1. **CIFAR Detection Results** OOD Detection performance comparison between zero-shot (ZS), linear-probing (LP), full fine-tuning (FFT), and `reprogrammer` (RP) methods using the `msp` [12] detector. All methods utilize the CLIP B/32 architecture fine-tuned on **CIFAR-10** as the in-distribution datasets. A description of all the semantically shifted OOD datasets is provided in Section 3.1. ↑ indicates larger values are better, while ↓ indicates smaller values are better. All values are percentages and **bold** values are the superior results.

D_{in}	Method	iSUN		LSUN Resize		Places365		Textures		Average	
		FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
No Tuning	ZS	**27.15**	95.08	**24.41**	95.61	**15.87**	97.12	32.36	92.60	**24.95**	95.10
CIFAR-10	LP	36.74	94.57	28.38	95.75	24.65	96.73	39.67	92.93	32.36	94.99
	FFT	45.47	92.78	42.95	93.41	40.92	94.06	44.85	92.30	42.89	93.40
	RP	29.58	**95.53**	25.96	**96.08**	15.94	**97.63**	**30.13**	**93.82**	25.40	**95.77**

Table 2. **CIFAR Generalization Results** OOD Generalization performance comparison between zero-shot (ZS), linear-probing (LP), full fine-tuning (FFT), and `reprogrammer` (RP) methods with **CIFAR-10** as the in-distribution dataset.

D_{in}	Method	CIFAR-10	CIFAR10.1	STL10
		Accuracy (↑)	Accuracy (↑)	Accuracy (↑)
No Tuning	ZS	89.23	83.30	**97.40**
CIFAR-10	LP	94.89	90.05	96.34
	FFT	**96.24**	91.05	55.90
	RP	95.39	**91.55**	**96.71**

Additionally, we present further ablations in Section 3.3 and Appendix F, along with more experimental results in Appendix E.

3.1. Experimental Setup

In-distribution dataset We tune our model with **CIFAR-10** [19] as the in-distribution (ID) dataset, which is a commonly used ID dataset for both OOD Generalization and OOD Detection experiments. The CIFAR-10 dataset contains labeled (32×32) resolution images covering a range of real-world objects such as horses, cats, and airplanes.

Out-of-distribution Generalization We evaluate the OOD Generalization performance on two standard covariate shifted OOD datasets. Specifically, we evaluate the generalization accuracy on the **CIFAR-10.1** [34] and **STL10** [6] datasets. Both these datasets contains images derived from semantically matching CIFAR-10 classes.

Out-of-distribution Detection For all compared downstream models, we evaluate using the `msp` detector against four commonly used OOD benchmarks. More specifically, we evaluate on the **iSUN** [40], **LSUN Resized** [42], **Places365** [45], and **Textures** [5] datasets. These OOD datasets span a wide range of objects including fine-grained images, scene images, and textural images. Additionally, these datasets are carefully chosen so that there is no semantic overlapping with respect to the CIFAR-10 dataset.

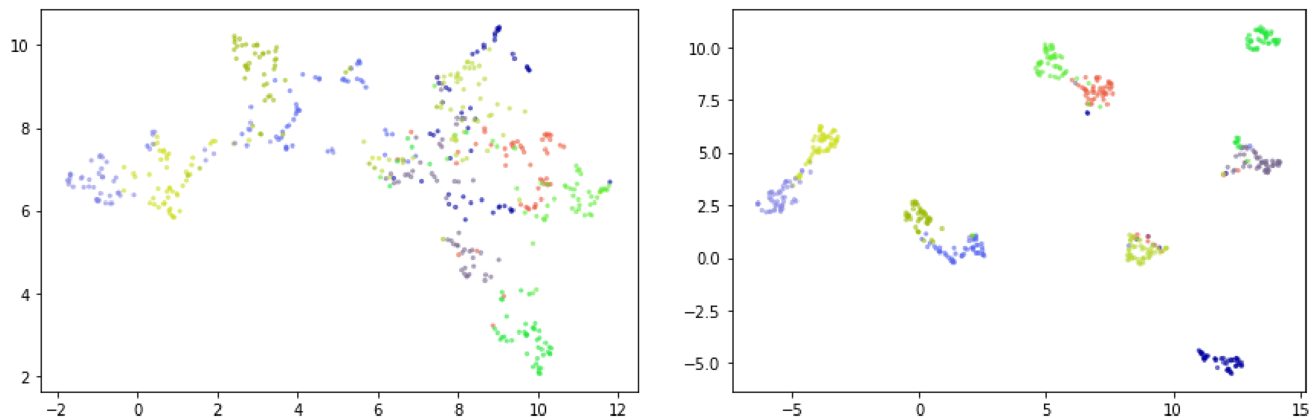
3.2. Results

Out-of-distribution Generalization We present our main results for OOD Generalization in Table 2, where we compare the OOD Generalization accuracy of our `reprogrammer` method in comparison to linear-probing and full fine-tuning.

We first observe that full fine-tuning outperforms zero-shot, linear-probing, and `reprogrammer` on the ID CIFAR-10 task, which is consistent with expectations set by prior works. However, we see that for the OOD Generalization tasks, `reprogrammer` consistently outperforms both linear-probing and full fine-tuning on each of the OOD Generalization benchmarks, with full fine-tuning in particular performing significantly worse in the STL10 OOD Generalization task. This also matches with intuition from prior work were naive fine-tuning can distort the diverse and beneficial pre-training representations necessary for OOD Generalization tasks [20]. Subsequently, given that `reprogrammer` encourages minimal alterations to the pre-trained parameters, we can observe that `reprogrammer` outperforms every other common transfer learning techniques on all of the given OOD generalization tasks.

Out-of-distribution Detection We present our main results for OOD Detection in Table 1. Specifically, we report the OOD Detection performances of our fine-tuned models across four semantically shifted OOD datasets, as well as the average across all four datasets. For a fair comparison we fix the OOD detector, leveraging the commonly used baseline `msp` detector [12], across all experiments as a way to gauge the level of overconfidence the zero-shot, linear-probed, full fine-tuned, and `reprogrammer` down-stream models have on semantically shifted OOD samples.

Firstly, we can see that both linear probing and full fine-tuning perform worse when compared with the zero-shot model. This supports our hypothesis that naive fine-tuning methods can degrade the diversely pre-trained representations needed for detecting semantically shifted OOD sam-



(a) Linear-probing Feature Space

(b) Reprogrammer Feature Space

Figure 2. UMAP visualization comparing the feature spaces between linear-probed and `reprogrammer` models using 500 randomly sampled covariate shifted (CIFAR-10.1) images.

ples. Secondly, we can also observe that `reprogrammer` outperforms every other fine-tuning technique on all of the given OOD Detection tasks. This additionally indicates to us that `reprogrammer` is better able to achieve this goal of maintaining necessary pre-training representations for a more semantically robust down-stream model.

3.3. Ablation Studies

Visualizing the Reprogrammed Feature Space In this ablation, we provide additional insights showcasing how `reprogrammer` can better align covariate shifted OOD samples. In Figure 2 we present *UMAP* visualizations comparing the feature space between the linear-probed and `reprogammer` models on covariate shifted OOD images [23]. Observing these visualization, we can see that our `reprogrammer` model is producing more separable, and more tightly bound, clusters of covariate features. This again supports our intuition that the *model reprogramming* technique is aligning the OOD samples to the already strongly tuned ID space, therefore enabling `reprogrammer` to better classify on covariate shifted OOD samples.

4. Conclusion

In this paper, we showcased that maintaining pre-training representation is critical to the robustness of the down-stream model with respect to both covariate and semantically shifted OOD samples. Additionally, we propose an alternative approach for transferring text-image encoder models called `reprogrammer` that attempts to minimize the distortion to the model's pre-training representations. Experimental results further showcases the strength of `reprogrammer` when compared to other common fine-tuning techniques. We hope that our work provides additional insights into the hidden costs of common transfer learning techniques, and inspire future works to leverage reprogramming approaches for transfer learning.

References

[1] Anders Andreassen, Yasaman Bahri, Behnam Neyshabur, and Rebecca Roelofs. The evolution of out-of-distribution robustness throughout fine-tuning. 2021. 1

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020. 7

[3] Pin-Yu Chen. Model reprogramming: Resource-efficient cross-domain machine learning. 2022. 1, 7

[4] Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020. 7

[5] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3, 8, 12

[6] Adam Coates, Andrew Ng, and Honlak Lee. An analysis of single-layer networks in unsupervised feature learning. *Fourteenth International Conference on Artificial Intelligence and Statistics*, 15:215—-223, 2011. 3, 8

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image

database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 12

[8] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. *Conference on Computer Vision and Pattern Recognition*, 2021. 7

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021. 7

[10] Gamaleldin F. Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. *International Conference on Learning Representations*, 2019. 2, 7, 9

[11] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *International Conference on Computer Vision*, 2021. 7, 8, 12

[12] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017*, 2017. 3, 7, 13

[13] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *Conference on Computer Vision and Pattern Recognition*, 2021. 7, 8, 12

[14] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960, 2020. 7

[15] Rui Huang and Yixuan Li. Towards scaling out-of-distribution detection for large semantic space. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 12

[16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *International Conference on Machine Learning*, 2021. 1

[17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *International Conference on Machine Learning*, 2021. 7

[18] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *European Conference on Computer Vision*, 2020. 7

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 3

[20] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *International Conference on Learning Representations*, 2022. 1, 3, 11

[21] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018*, 2018. 7

[22] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020. 7

[23] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018. 4

[24] John P Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-ofdistribution and in-distribution generalization. *International Conference on Machine Learning*, 2021. 1

[25] Paarth Neekhara, Shehzeen Hussain, Jinglong Du, Shlomo Dubnov, Farinaz Koushanfar, and Julian McAuley. Cross-modal adversarial reprogramming. *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2427—-2435, 2022. 7

[26] Paarth Neekhara, Shehzeen Hussain, Shlomo Dubnov, and Farinaz Koushanfar. Adversarial reprogramming of text classification neural networks. *Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, 2019. 2, 7, 10

[27] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. 1

[28] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, Mingxing Tan, and Quoc V. Le. Combined scaling for open-vocabulary image classification, 2021. 1, 11

[29] Alec Radford, Jong Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *International Conference on Machine Learning (ICML)*, 139:8748–8763, 2021. 1, 2, 7, 10

[30] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 7

[31] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *International Conference on Machine Learning*, 2019. 7, 8, 12

[32] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. *European Conference on Computer Vision*, 2020. 7

[33] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification.

Advances in Neural Information Processing Systems, 2020. 1

[34] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008. 3, 8

[35] Yun-Yun Tsai Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. *International Conference on Machine Learning*, 2020. 7, 9

[36] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 8, 12

[37] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019. 7, 8, 12

[38] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. 2021. 1, 9

[39] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 8, 12

[40] Pingmei Xu, Krista A. Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R. Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *CoRR*, abs/1504.06755, 2015. 3, 8

[41] Chao-Han Huck Yang, Yun-Yun Tsai, and Pin-Yu Chen. Voice2series: Reprogramming acoustic models for time series classification. In *International Conference on Machine Learning*, pages 11808–11819. PMLR, 2021. 7

[42] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 3, 8

[43] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. *Conference on Computer Vision and Pattern Recognition*, 2022. 7

[44] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, and Curtis P. Langlotz. Contrastive learning of medical visual representations from paired images and text. 2020. 7

[45] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 3, 8, 12