Out-of-Distribution Robustness via Targeted Augmentations

Irena Gao, Shiori Sagawa, Pang Wei Koh, Tatsunori Hashimoto, Percy Liang
Stanford University

Abstract

Machine learning systems often must generalize across real-world domains with different data distributions. These distributions change along multiple factors: while some of these factors are spuriously correlated with the label, others are robustly predictive. For example, in wildlife conservation, animal classification models must generalize across camera deployments, with cameras' background distributions varying along both spurious factors (e.g., low-level background variations) and robustly predictive factors (e.g., the background's habitat type). In this work, we show that data augmentations offer significant out-of-distribution gains when they are carefully designed to randomize only spurious variations, while preserving the robust variations. On IWILDCAM2020-WILDS and CAMELYON17-WILDS, two domain generalization datasets, targeted augmentations outperform the previous state-of-the-art by 3.2% and 14.4% respectively. Our results suggest that data augmentations, when targeted to selectively randomize spurious cross-domain variations, can be an effective route to real-world out-of-distribution robustness.

1 Introduction

Machine learning systems are often deployed across multiple domains, including new domains that were unseen during training. Distribution shifts between domains can substantially degrade model performance [1, 2, 3, 4], especially in real-world settings, where generalizing to new domains remains persistently challenging, even for state-of-the-art domain generalization methods [1]. In this work, we show that *data augmentations* are very effective at improving out-of-distribution (OOD) robustness on two such real-world settings from the WILDS benchmark [1]. In particular, while we benchmark several data augmentations, we find that a set of augmentations, which we term *targeted augmentations*, are most effective by a large margin. Targeted augmentations incorporate application knowledge to decompose cross-domain variations into a set of *spurious factors* (i.e., uninformative across domains) versus *robustly predictive factors*. They then selectively randomize only the spurious factors, while preserving the robustly predictive factors.

We study targeted augmentations for two settings. The first setting, IWILDCAM2020-WILDS, involves classifying animals for wildlife conservation; the domains are camera traps (Figure 1, top) [5], which differ along spurious factors such as low-level variations in backgrounds (e.g., whether there is a tree on the left vs. right) and along predictive factors such as high-level variations in backgrounds that encode the habitat (e.g., jungle vs. grassland). A targeted augmentation in IWILDCAM2020-WILDS, adapted from application-specific prior work [6], copies and pastes animals onto backgrounds from other cameras to be invariant to low-level background variations. However, this augmentation only selects backgrounds from cameras that have observed the copied species in the training set, which avoids breaking correlations between label and the background's habitat type. The second setting, CAMELYON17-WILDS, involves classifying tumors for histopathology (Figure 1, bottom) [7]; the domains are hospitals, which differ along spurious factors like stain color [8] as well as predictive factors inherited from different patient populations (e.g., tumor staging and morphology) [9, 10]. In

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

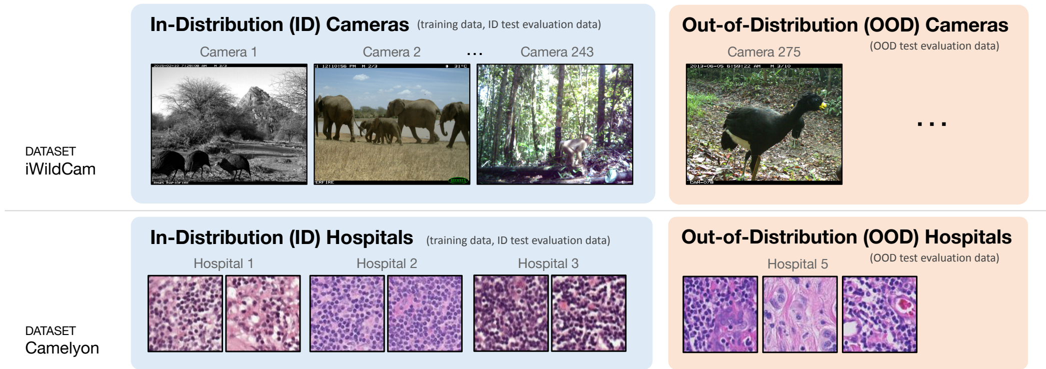


Figure 1: We study two domain generalization datasets, IWILDCAM2020-WILDS (top) and CAMELYON17-WILDS (bottom). They consist of data from different domains, which vary along factors such as location for IWILDCAM2020-WILDS and stain intensities for CAMELYON17-WILDS.

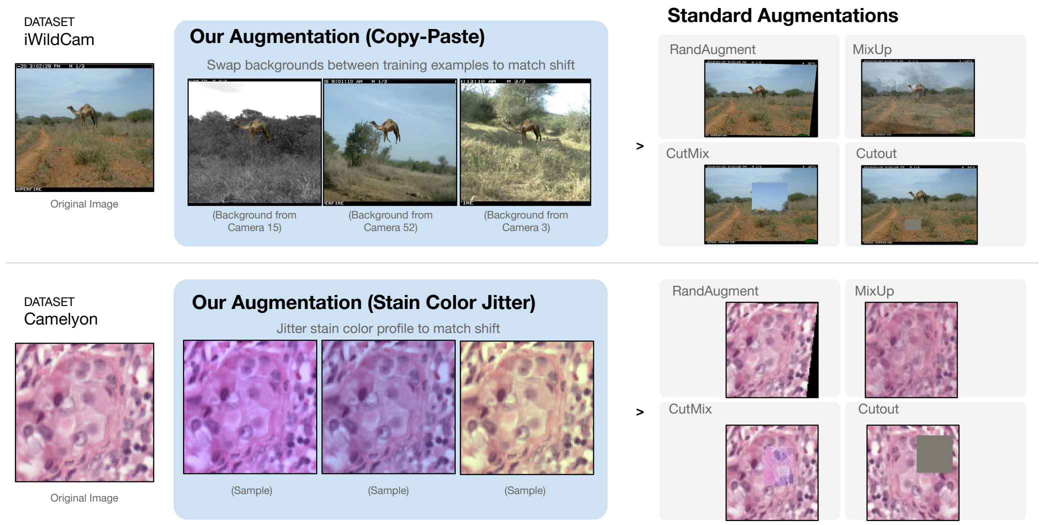


Figure 2: Our targeted augmentations randomize camera backgrounds (for IWILDCAM2020-WILDS) and average stain color intensities (for CAMELYON17-WILDS), eliminating a selected factor of variation between domains shown in Figure 1.

CAMELYON17-WILDS, we jitter the average stain color for each patch to be invariant to staining variations [11]; this augmentation randomizes stain levels, without affecting cell shapes.

These targeted augmentations achieve state-of-the-art performance by a wide margin: 3.2 points on IWILDCAM2020-WILDS and 14.4 points on CAMELYON17-WILDS over the previous best, outperforming two sets of baselines: standard augmentations in computer vision applications, and domain invariance methods. Standard augmentations encourage invariance to specific transformations, but they do not necessarily target cross-domain variations. We observe that these augmentations can improve both out-of-distribution (OOD) and in-distribution (ID) performance, but their OOD gains do not outpace their ID gains as much as targeted augmentations; in other words, they do not improve effective robustness [12]. On the other hand, domain invariance methods encourage broad invariance across domains [13, 14, 15], but unlike our targeted augmentations, they do not selectively target *spurious* cross-domain variation. We observe that while these methods can improve effective robustness, they have substantially worse ID *and* OOD performance compared to targeted augmentations, and we speculate that their broad, untargeted nature makes them both less effective at encouraging invariance to spurious variation and at preserving predictive variation. Altogether, our results suggest that targeted augmentations, which isolate and randomize spurious cross-domain variations using prior knowledge, are a promising avenue for improving real-world OOD performance.

2 Related Work

Spurious versus predictive cross-domain variations. In this work, we decompose features which vary between domains into spurious and predictive features. A related decomposition has been used in the context of causal approaches to robust learning [16, 17, 18], where prior knowledge is used to map all non-causal features to spurious features, treating only causal features as predictive. Our experiments on iWildCam2020-wilds suggest that such a restrictive definition for a robust feature (which excludes background habitat features, for example) can hurt task performance. Orthogonally, domain invariance methods penalize reliance on any (conditional) variations across domains [19, 13]. This set may include both spurious and robust features if these distributions vary across domains [18]. For example, in Camelyon17-wilds, this set includes features impacted by cancer staging, which varies across hospitals.

Data augmentations for OOD robustness. Data augmentations are a cornerstone of in-distribution (ID) image classification [20, 21, 22, 23]. The mechanism by which augmentations are helpful is not well-understood, although in the ID setting, prior work has framed augmentation as providing variance reduction or other regularization [24, 25, 26]. ID-successful augmentations have also been evaluated in OOD settings, where they can sometimes outperform domain generalization algorithms [4, 1]. Others have designed augmentations specifically for OOD generalization [27, 2, 28]. Our work suggests that augmentations are most successful OOD when they are targeted to a particular distribution shift, eliminating spurious cross-domain variations while preserving predictive ones.

3 Setup

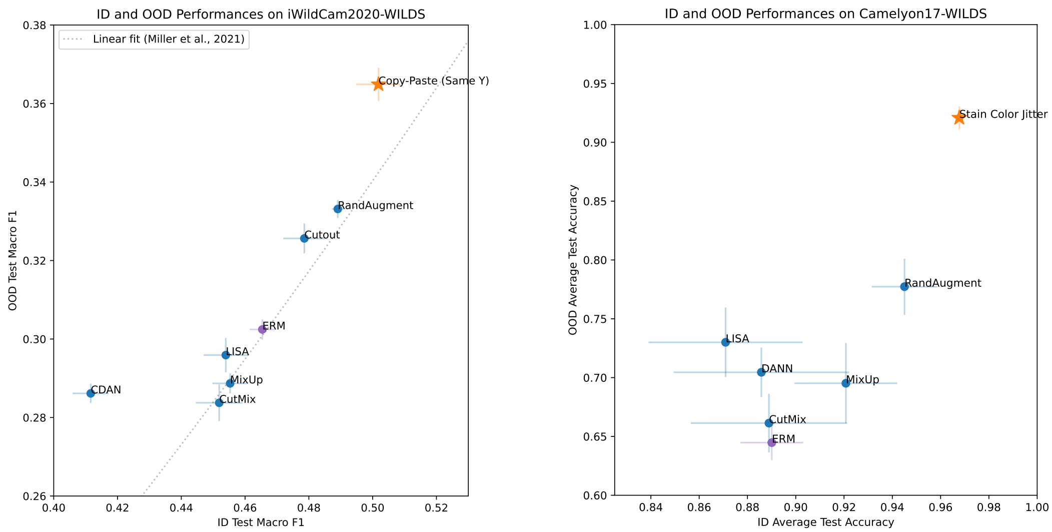
Domain generalization. We consider a domain generalization setting based on Koh et al. [1], where the goal is to generalize to test domains $\mathcal{D}^{\text{test}}$ which are disjoint from the training domains $\mathcal{D}^{\text{train}}$ (i.e., $\mathcal{D}^{\text{train}} \cap \mathcal{D}^{\text{test}} = \emptyset$). Each domain d corresponds to a data distribution P_d over examples (x, y, d), where x is the input, y is the label, and d is the domain. The training distribution $P^{\text{train}} = \sum_{d \in \mathcal{D}^{\text{train}}} q_d^{\text{train}} P_d$ is a mixture of per-domain data distributions, made up of training domains $\mathcal{D}^{\text{train}}$ with mixture weights q_d^{train}. Similarly, the test distribution $P^{\text{test}} = \sum_{d \in \mathcal{D}^{\text{test}}} q_d^{\text{test}} P_d$ is mixture composed of test domains $\mathcal{D}^{\text{test}}$ with mixture weights q_d^{test}. We train a model $\theta \in \Theta$ on examples drawn from the training distribution P^{train}, with the goal of maximizing its *out-of-distribution (OOD)* performance on the test distribution P^{test}. In addition to the OOD performance, we evaluate the model's *in-distribution (ID)* performance on held-out samples from the training distribution P^{train}.

Targeted augmentations. We study application-tailored augmentations that randomize the spurious factors of variation across domains. These targeted augmentations rely on a decomposition of the input into predictive and spurious components, as defined using prior, application-specific knowledge: $x = f(x_{\text{core}}, x_{\text{spu}})$, where x_{core} refers to the robustly predictive features and x_{spu} is the spurious feature. A targeted augmentation A applies a spurious transform A_{spu} on the spurious features, while preserving the predictive features: $A(x) = f(x_{\text{core}}, A_{\text{spu}}(x_{\text{spu}}))$, where A_{spu} is a stochastic function that intuitively transforms the spurious features into those from another domain. We train a model by minimizing the average loss on the augmented examples: $\hat{\theta} = \text{argmin}_{\theta \in \Theta} \mathbb{E}_{\hat{P}^{\text{train}}} [\ell(\theta; (A(x), y))]$.

4 Datasets and Augmentations

We study targeted augmentations from prior work on two datasets from the Wilds benchmark [1], as summarized in Figures 1 and 2. Appendix A provides additional details on the augmentations.

Copy-Paste on iWildCam2020-wilds. In iWildCam2020-wilds [1, 29], the input x is a color photograph, the label y is either an animal species or "empty", and the domain d is the identity of the static camera trap that captured the image. There are 243 ID cameras and 48 OOD cameras. We study the Copy-Paste augmentation [30, 31, 32, 33], which targets cross-camera variations in image backgrounds, extending earlier work by Beery et al. [6] (see Appendix D). Copy-Paste randomizes image backgrounds while fixing the animal foreground; specifically, this is a spurious transform A_{spu} that samples a background from the subset of training domains in which the *same animal species has been observed*. This roughly corresponds to sampling backgrounds within the same habitat (see



(a) IWILDCAM2020-WILDS results

(b) CAMELYON17-WILDS results

Figure 3: ID Test (horizontal) vs. OOD Test (vertical) performances of targeted augmentations and baselines over 5 replicates for IWILDCAM2020-WILDS and 10 replicates for CAMELYON17-WILDS. Points are mean performances, and error bars are standard errors. Targeted augmentations significantly outperform standard augmentations and domain invariance methods.

Appendix A), so it preserves not only animal foregrounds, but also high-level habitat features in the background, while encouraging invariance to spurious, low-level background variations.

Stain Color Jitter on CAMELYON17-WILDS. In CAMELYON17-WILDS [1, 7], the input x is a colored image of a tissue patch, the label y is whether the patch contains a tumor, and the domain d is the identity of the hospital that collected the sample. There are 3 ID hospitals and 1 OOD hospital, with a 5th hospital used for validation. We study the Stain Color Jitter augmentation from Tellez et al. [11], which targets cross-hospital variations in staining procedures. Stain Color Jitter randomizes the average stain level of each patch, while fixing all other information as predictive features, including the cell structures and relative stain levels within each patch. This is contrast with standard augmentations like MixUp [21], which average stain colors between examples but also affect the resulting image's cell shapes. Specifically, the spurious transform A_{spu} applies a random affine transform to the average staining level for each stain.

5 Experiments

We compare targeted augmentations with two **domain invariance methods**: (C)DANN [19, 13], which penalizes representations from which a (label-conditioned) discriminator can predict domain; and LISA [15], a data augmentation that interpolates between examples of the same class from different domains. We also compare to **standard data augmentations** in computer vision: RandAugment [20], CutMix [22], MixUp [21], and Cutout [34], which were designed to improve ID performance but have also improved OOD performance in some settings [35, 3, 36]. Additional baseline and training details are in Appendices B and C.

Results. Targeted augmentations significantly improve OOD performance (Figure 3), achieving state-of-the-art performance on both datasets. Compared to the best-performing baseline on the WILDS leaderboard [1] (RandAugment [20]), targeted augmentations improve OOD Macro F1 on IWILDCAM2020-WILDS from 33.3% $\rightarrow$ 36.5% and OOD average accuracy on CAMELYON17-WILDS from 77.7% $\rightarrow$ 92.1%. We note that these targeted augmentations also match or outperform unsupervised domain adaptation methods as benchmarked by Sagawa et al. [35], where previous bests were 32.1% OOD Macro F1 on IWILDCAM2020-WILDS (set by Noisy Student [37]) and

	Copy-Paste (Same Y)	Copy-Paste (All Backgrounds)	Foreground Only
OOD Macro F1	36.5 (0.4)	34.7 (0.4)	32.9 (0.5)
ID Macro F1	50.2 (0.7)	47.1 (1.1)	42.5 (0.7)

Table 1: Ablation on preserving predictive features in iWildCam2020-wilds. Performance degrades when habitat-based predictive features in the background are randomized (center column) or removed (right column).

91.4% OOD average accuracy on Camelyon17-wilds (set by SwAV [38]). Both unsupervised methods also rely on data augmentation as a core subroutine.

On iWildCam2020-wilds, Miller et al. [12] showed that the ID and OOD performance of a wide variety of models were strikingly linearly correlated; we plot their linear fit on Figure 3a. We found that our targeted Copy-Paste augmentation conferred what Miller et al. [12] term *effective robustness*, which is represented in the plot by a vertical offset from the line. In contrast, none of the standard augmentations we tested improved effective robustness. While the domain invariance methods we tested also showed effective robustness, their overall ID and OOD performances are substantially worse, even when compared to standard empirical risk minimization (ERM). On Camelyon17-wilds, Miller et al. [12] did not establish a clear linear trend. Nevertheless, we found that targeted augmentations compare similarly to baselines on Camelyon17-wilds as on iWildCam2020-wilds; consistent with prior work exploring stain color jitter in histopathology applications, [8, 12], it significantly improves OOD accuracy.

Preserving predictive features. To illustrate the importance of preserving predictive features that vary between domains, we ran an ablation on the Copy-Paste augmentation for iWildCam2020-wilds. Our augmentation Copy-Paste (Same Y) attempts to preserve the plausibility of the augmented image by only swapping backgrounds with from training domains in which the same animal species has been observed. In contrast, we now study a Copy-Paste (All Backgrounds) variant that swaps backgrounds randomly with all other training domains, including training domains in which the same animal species was not observed. This variant does not preserve habitat-based predictive features—e.g., the fact that camels are found in arid regions (Figure 2)—and its OOD performance correspondingly drops by 1.8% (Table 1), illustrating the value of designing targeted augmentations to preserve predictive cross-domain variations.

Predictive features need not be causal. The iWildCam2020-wilds setting also illustrates that robustly predictive factors need not be causal factors: for example, in iWildCam2020-wilds, a camel placed in a jungle is still a camel, but in realistic wildlife conservation settings we would expect a jungle background to be robustly indicative that the animal is unlikely to be a camel. While only utilizing causal features guards against worst-case variations across domains, keeping predictive, possibly non-causal features aims to guard against only realistic shifts across domains. Fully utilizing such predictive, non-causal factors is particularly important in real-world settings where causal features have high noise rates, such as in iWildCam2020-wilds, where animal foregrounds may be blurry, dimly lit, or camouflaged. To illustrate this, we trained a model on iWildCam2020-wilds using images containing only the animal foreground (i.e., the backgrounds are replaced with a solid color). We then evaluated this model on a transformed version of the evaluation set, such that images only contain the animal foreground. In this setting, the model is trained and evaluated on its ability to predict animal species using only the causal feature (foreground). However, this model attains an average OOD performance of 32.9%, underperforming Copy-Paste's 36.5% (Table 1). This suggests that leveraging robustly predictive, non-causal features can preserve task performance across realistic shifts.

Conclusion. Altogether, our results show that using prior knowledge to design targeted augmentations, which randomize spurious cross-domain variations while preserving predictive variations, can lead to significant improvements in out-of-distribution robustness. We hope that such an approach can be helpful in other real-world applications seeking to generalize across domains.

References

[1] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[2] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

[3] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Taylan Cemgil, et al. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

[4] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[5] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[6] Sara Beery, Yang Liu, Dan Morris, Jim Piavis, Ashish Kapoor, Neel Joshi, Markus Meister, and Pietro Perona. Synthetic examples improve generalization for rare classes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 863–873, 2020.

[7] Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018.

[8] David Tellez, Geert Litjens, Péter Bándi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen van der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical image analysis*, 58, 2019.

[9] Brian E Henderson, Norman H Lee, Victoria Seewaldt, and Hongbing Shen. The influence of race and ethnicity on the biology of cancer. *Nature Reviews Cancer*, 12(9):648–653, 2012.

[10] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M Stuart, Cancer Genome Atlas Research Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10), 2013.

[11] David Tellez, Maschenka Balkenhol, Irene Otte-Höller, Rob van de Loo, Rob Vogels, Peter Bult, Carla Wauters, Willem Vreuls, Suzanne Mol, Nico Karssemeijer, et al. Whole-slide mitosis detection in h&e breast histology using phh3 as a reference to train distilled stain-invariant convolutional networks. *IEEE transactions on medical imaging*, 37(9):2126–2136, 2018.

[12] 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-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, pages 7721–7735. PMLR, 2021.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[14] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[15] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. *arXiv preprint arXiv:2201.00299*, 2022.

[16] Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pages 2839–2848. PMLR, 2016.

[17] Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *Machine Learning*, 110(2):303–348, 2021.

[18] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *International Conference on Machine Learning*, pages 7313–7324. PMLR, 2021.

[19] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

[20] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[21] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[22] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[23] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Keepaugment: A simple information-preserving data augmentation approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1055–1064, 2021.

[24] Shuxiao Chen, Edgar Dobriban, and Jane H Lee. A group-theoretic framework for data augmentation. *The Journal of Machine Learning Research*, 21(1):9885–9955, 2020.

[25] Tri Dao, Albert Gu, Alexander Ratner, Virginia Smith, Chris De Sa, and Christopher Ré. A kernel theory of modern data augmentation. In *International Conference on Machine Learning*, pages 1528–1537. PMLR, 2019.

[26] Sen Wu, Hongyang Zhang, Gregory Valiant, and Christopher Ré. On the generalization effects of linear transformations in data augmentation. In *International Conference on Machine Learning*, pages 10410–10420. PMLR, 2020.

[27] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

[28] Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Dawn Song, and Jacob Steinhardt. Pixmix: Dreamlike pictures comprehensively improve safety measures. *arXiv preprint arXiv:2112.05135*, 2021.

[29] Sara Beery, Arushi Agarwal, Elijah Cole, and Vighnesh Birodkar. The iwildcam 2021 competition dataset. *arXiv preprint arXiv:2105.03494*, 2021.

[30] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021.

[31] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1301–1310, 2017.

[32] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 364–380, 2018.

[33] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 682–691, 2019.

[34] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[35] Shiori Sagawa, Pang Wei Koh, Tony Lee, Irena Gao, Sang Michael Xie, Kendrick Shen, Ananya Kumar, Weihua Hu, Michihiro Yasunaga, Henrik Marklund, et al. Extending the wilds benchmark for unsupervised adaptation. *arXiv preprint arXiv:2112.05090*, 2021.

[36] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021.

[37] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[38] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[39] Sara Beery, Dan Morris, and Siyu Yang. Efficient pipeline for camera trap image review. *arXiv preprint arXiv:1907.06772*, 2019.

[40] Vighnesh Birodkar, Zhichao Lu, Siyang Li, Vivek Rathod, and Jonathan Huang. The surprising impact of mask-head architecture on novel class segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7015–7025, 2021.

[41] Kazuki Irie, Imanol Schlag, Róbert Csordás, and Jürgen Schmidhuber. Improving baselines in the wild. *arXiv preprint arXiv:2112.15550*, 2021.

[42] Arnout C Ruifrok, Dennis A Johnston, et al. Quantification of histochemical staining by color deconvolution. *Analytical and quantitative cytology and histology*, 23(4):291–299, 2001.

[43] Bo Fu Mingsheng Long Junguang Jiang, Baixu Chen. Transfer-learning-library. `https://github.com/thuml/Transfer-Learning-Library`, 2020.

[44] Junguang Jiang, Yang Shu, Jianmin Wang, and Mingsheng Long. Transferability in deep learning: A survey, 2022.

[45] Yufei Wang, Haoliang Li, and Alex C Kot. Heterogeneous domain generalization via domain mixup. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3622–3626. IEEE, 2020.

[46] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alex Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. *arXiv preprint arXiv:2106.04732*, 2021.

[47] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020.

[48] Yabin Zhang, Haojian Zhang, Bin Deng, Shuai Li, Kui Jia, and Lei Zhang. Semi-supervised models are strong unsupervised domain adaptation learners. *arXiv preprint arXiv:2106.00417*, 2021.

[49] Raphael Gontijo-Lopes, Sylvia J Smullin, Ekin D Cubuk, and Ethan Dyer. Affinity and diversity: Quantifying mechanisms of data augmentation. *arXiv preprint arXiv:2002.08973*, 2020.

[50] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

[51] Chaitanya Ryali, David J Schwab, and Ari S Morcos. Learning background invariance improves generalization and robustness in self-supervised learning on imagenet and beyond. 2021.

[52] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

[53] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[54] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision*, 73(2):213–238, 2007.

[55] Rakshith Shetty, Bernt Schiele, and Mario Fritz. Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8226, 2019.

[56] Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv preprint arXiv:1808.03305*, 2018.

[57] Zhuotun Zhu, Lingxi Xie, and Alan L Yuille. Object recognition with and without objects. *arXiv preprint arXiv:1611.06596*, 2016.

A Augmentation Details

A.1 Copy-Paste on iWildCam2020-wilds

The full Copy-Paste protocol is given in Algorithm 1. We consider two strategies for selecting the set of valid empty backgrounds $B^{(i)}$.

1. **Copy-Paste (All Backgrounds): all empty train split images.** $B^{(i)} = \{(x, y, d) \in \mathcal{D}_{\text{train}} : y = \text{“empty”}\}$, i.e., all augmented examples should have a single distribution of backgrounds. There is a large set of training backgrounds to choose from when executing the procedure – of $129,809$ training images, $48,021$ are empty images.

2. **Copy-Paste (Same Y): empty train split images from cameras that have observed $y^{(i)}$.** Let $\mathcal{Y}(d)$ represent the set of labels domain d observes. Then $B^{(i)} = \{(x, y, d) \in \mathcal{D}_{\text{train}} : y = \text{“empty” and } y^{(i)} \in \mathcal{Y}(d)\}$.

Algorithm 1: Copy-Paste

Input: Labeled example $(x^{(i)}, y^{(i)}, d^{(i)})$, binary segmentation mask $m^{(i)}$, set of images to sample empty images from to use as backgrounds $B^{(i)}$

if $y^{(i)}$ = *“empty” or* $|B^{(i)}| = 0$ **then**
  return $x^{(i)}$
Copy out foreground by applying segmentation mask $f^{(i)} := m^{(i)} \circ x^{(i)}$
Randomly select a background $b \in B^{(i)}$
Paste $f^{(i)}$ onto b and **return** $\tilde{x}^{(i)} := \text{Paste}(f^{(i)}, b)$

Segmentation Masks. The iWildCam2020-wilds dataset is curated from real camera trap data collected by the Wildlife Conservation Society and released by Koh et al. [1], Beery et al. [29]. Beery et al. [29] additionally compute and release segmentation masks for all labeled examples in iWildCam2020-wilds. These segmentation masks were extracted by running the dataset through MegaDetector [39] and then passing regions within detected boxes through an off-the-shelf, class-agnostic detection model, DeepMAC [40]. We use these segmentation masks for our Copy-Paste augmentation.

Intuition. Most cameras in iWildCam2020-wilds observe a very limited set of labels; although there are 182 classes in iWildCam2020-wilds overall, Irie et al. [41] report that more than 50% of domains observe fewer than 6 labels. The label support of each camera is strongly correlated with the *habitat* that a camera observes – cameras in the jungle are unlikely to include camels in their support. This suggests that when cameras overlap classes, the cameras are themselves related: i.e., , the cameras observe the same ecological habitat. We thus expect that Copy-Paste (Same Y) to randomize low-level variations in background (e.g., , the particular location observed within a habitat) while preserving high-level background variations between cameras (e.g., keeping jungle animals on jungle backgrounds and desert animals on desert backgrounds).

A.2 Stain Color Jitter on Camelyon17-wilds

The full Stain Color Jitter protocol, originally from Tellez et al. [11], is given in Algorithm 2. The augmentation uses a pre-specified Optical Density (OD) matrix from Ruifrok et al. [42] to project images from RGB space to a three-channel hematoxylin, eosin, and DAB space before applying a random linear combination.

Intuition. Hospitals in Camelyon17-wilds vary in their class-separated color histograms (Figure 4). The means of these color distributions are spuriously correlated with the label. In the three training hospitals (top 3 panels), the negative class color distribution has a larger mean than the positive class color distribution; this trend is reversed in the OOD test hospital (bottom panel). We aim for stain color jitter to remove the correlation between mean color and label in the training data.

Algorithm 2: Stain Color Jitter Augmentation

Input: Labeled example $(x^{(i)}, y^{(i)}, d^{(i)})$, normalized OD matrix M [42], tolerance $\epsilon = 1^{-6}$
$S = -\log(x^{(i)} + \epsilon)M^{-1}$
Sample $\alpha \sim \text{Uni}(1-\sigma, 1+\sigma)$
Sample $\beta \sim \text{Uni}(-\sigma, \sigma)$
$P = \exp[-(\alpha S + \beta)M] - \epsilon$
return P with each cell clipped to $[0, 255]$

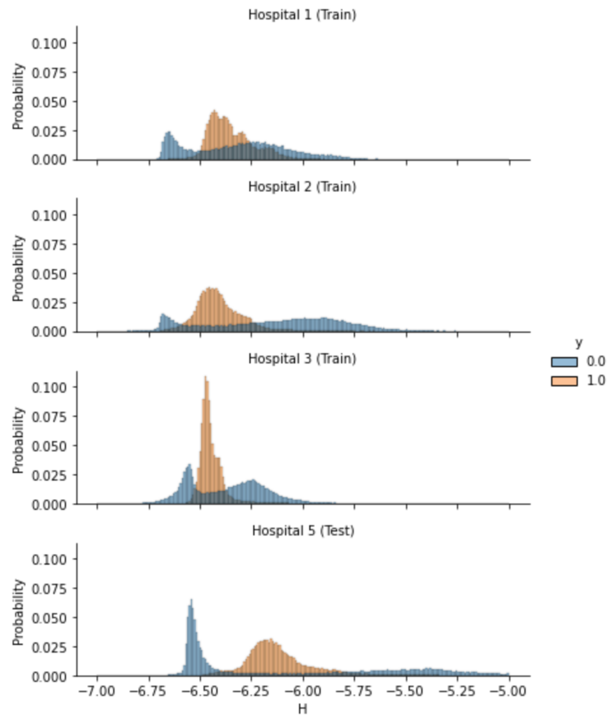


Figure 4: Class-separated color histograms for CAMELYON17-WILDS.

B Baselines

We compare our augmentations to baseline methods which have improved OOD performance in prior work, including both adversarial methods (e.g., DANN, CDAN) and other data augmentations. Some of these methods were designed for domain generalization – they require domain annotations $d^{(i)}$ during training and optimize for some notion of domain invariance. Other methods (e.g., standard data augmentations) were designed for ID generalization but have been applied to domain generalization problems in prior work.

Below, we describe each baseline and additional implementation decisions.

CDAN [19] – adversarial training, uses domain information. CDAN optimizes for domain invariance by penalizing representations from which a discriminator can easily predict domains, conditioned on y. In other words, CDAN penalizes feature variance across domains *within* y. Given features $\Phi(x)$, classification head g, and a domain discriminator h, the CDAN loss is

$$\text{CrossEntropy}(y, g \circ \Phi(x)) - \lambda \text{CrossEntropy}\left(d, h\left(\Phi(x), y\right)\right)$$

We use the implementation of CDAN from Gulrajani and Lopez-Paz [4], which uses an MLP for h. CDAN has four hyperparameters: λ, a featurizer learning rate, a classifier learning rate, and a discriminator learning rate. We run CDAN on IWILDCAM2020-WILDS. Because each domain in CAMELYON17-WILDS is class-balanced, we swap CDAN for DANN (below) on CAMELYON17-WILDS.

DANN [13] – adversarial training, uses domain information. DANN optimizes for domain invariance by penalizing representations from which a discriminator can easily predict domains. Given features $\Phi(x)$, classification head g, and a domain discriminator h, the DANN loss is

$$\text{CrossEntropy}(y, g \circ \Phi(x)) - \lambda \text{CrossEntropy}(d, h \circ \Phi(x))$$

We use the implementation of DANN from Junguang Jiang [43], Jiang et al. [44], which uses a 3-layer MLP for h and four hyperparameters: λ, a featurizer learning rate, a classifier learning rate, and a discriminator learning rate. We discuss our tuning strategy in Appendix C. We run DANN on CAMELYON17-WILDS because each domain is class-balanced. On IWILDCAM2020-WILDS, where domains have extreme class imbalances, we instead run CDAN (above).

LISA or Domain Mix-Up [15, 45] – data augmentation, uses domain information. LISA encourages domain invariance by mixing examples from the same label across different domains. It has improved OOD performance on some datasets [15, 45, 3]. Given an example $x^{(i)}$ from domain $d^{(i)}$, LISA samples another example $x^{(j)}$ where $y^{(i)} = y^{(j)}$ but $d^{(i)} \neq d^{(j)}$. LISA then generates synthetic examples that interpolate between $x^{(i)}, x^{(j)}$, either via MixUp or CutMix with parameter α (see below). We follow Yao et al. [15] and fix $\alpha = 2$, grid searching over the use of MixUp versus CutMix to interpolate between $x^{(i)}, x^{(j)}$.

Vanilla MixUp [21] – data augmentation. MixUp has improved OOD performance on some distribution shifts [15, 36]. MixUp generates synthetic examples that smoothly interpolate between pairs of real examples. Concretely, two examples $x^{(i)}$ and $x^{(j)}$, MixUp samples a mixing parameter $\lambda \sim \text{Beta}(\alpha, \alpha)$, where α is a hyperparameter, and combines $x^{(i)}, x^{(j)}$ to produce

$$\tilde{x}^{(i)} := \lambda x^{(i)} + (1 - \lambda) x^{(j)} \tag{1}$$

$$\tilde{y}^{(i)} := \lambda y^{(i)} + (1 - \lambda) y^{(j)} \tag{2}$$

and corresponding $\tilde{x}^{(j)}, \tilde{y}^{(j)}$. We follow Zhang et al. [21] and grid search over $\alpha \in \{0.2, 0.4\}$.

Vanilla CutMix [22] – data augmentation. CutMix has improved OOD performance on some datasets [36]. CutMix, like Copy-Paste, involves pasting pixels from some training examples onto other examples. Given two examples $x^{(i)}$ and $x^{(j)}$, CutMix randomly samples a rectangle of area parameterized by $\lambda \sim \text{Beta}(\alpha, \alpha)$, where α is a hyperparameter, pastes that rectangle from $x^{(i)}$ onto $x^{(j)}$ and vice versa, and mixes the labels according to the ratio of pixels, i.e.,

$$\tilde{y}^{(i)} := \left(1 - \frac{\text{number of pixels from } x^{(j)}}{\text{total number of pixels}}\right) y^{(i)} + \left(\frac{\text{number of pixels from } x^{(j)}}{\text{total number of pixels}}\right) y^{(j)} \tag{3}$$

We follow Yun et al. [22] and grid search over $\alpha \in \{0.5, 1.0\}$.

RandAugment [20] – data augmentation. RandAugment is a common augmentation explored for OOD generalization [3, 35] and features as a subroutine in Berthelot et al. [46], Sohn et al. [47], Xie et al. [37], Sagawa et al. [35]. For each example, RandAugment samples a sequence of k PIL operations (e.g., rotate, shear, autocontrast, RGB color jitter) and applies these operations in sequence, each with a randomly sampled magnitude, followed by a random horizontal flip. We use the implementation of RandAugment from Zhang et al. [48] and search over $k \in \{1, 2\}$, following Sagawa et al. [35].

Cutout [34] – data augmentation, uses bounding boxes. Cutout has improved OOD performance on some datasets [36]. For each example, Cutout samples a random rectangle of the image to erase (i.e., replace with gray pixels), followed by a random horizontal flip. Because Cutout may accidentally occlude the animal foreground in IWILDCAM2020-WILDS, we also implement a version of Cutout with bounding box knowledge, such that no rectangle occludes more than 50% of animal bounding boxes. We grid search over the original and bounding box-aware version of Cutout.

C Hyperparameter strategy

We tuned all benchmarked methods by fixing a budget of 10 tuning runs per method with one replicate each. For each method, we selected final hyperparameters and carried out early stopping using the OOD validation splits of IWILDCAM2020-WILDS and CAMELYON17-WILDS.

C.1 Hyperparameter grids for iWildCam2020-wilds

All experiments used a ResNet-50, pretrained on ImageNet, with no weight decay and batch size 24. We applied all data augmentations stochastically with some *transform probability*, since we found that doing so improved performance as in prior work [49].

Method	Hyperparameters
ERM	Learning rate $\sim 10^{\text{Uni}(-5,-2)}$
Copy-Paste	Learning rate $\sim 10^{\text{Uni}(-5,-2)}$ Transform probability $\sim \text{Uni}(0.5, 0.9)$
LISA	Learning rate $\sim 10^{\text{Uni}(-5,-2)}$ Transform probability $\sim \text{Uni}(0.5, 0.9)$ Interpolation method $\in$ {MixUp, CutMix}
Vanilla MixUp	Learning rate $\sim 10^{\text{Uni}(-5,-2)}$ Transform probability $\sim \text{Uni}(0.5, 0.9)$ $\alpha \in \{0.2, 0.4\}$
Vanilla CutMix	Learning rate $\sim 10^{\text{Uni}(-5,-2)}$ Transform probability $\sim \text{Uni}(0.5, 0.9)$ $\alpha \in \{0.5, 1.0\}$
RandAugment	Learning rate $\sim 10^{\text{Uni}(-5,-2)}$ Transform probability $\sim \text{Uni}(0.5, 0.9)$ $k \in \{1, 2\}$
CutOut	Learning rate $\sim 10^{\text{Uni}(-5,-2)}$ Transform probability $\sim \text{Uni}(0.5, 0.9)$ Version $\in$ {Original, Bounding box-aware}
CDAN	Classifier learning rate $\sim 10^{\text{Uni}(-5.5,-4)}$ Discriminator learning rate $\sim 10^{\text{Uni}(-5.5,-4)}$ $\lambda \sim 10^{\text{Uni}(-0.3,1)}$

Table 2: Hyperparameter search spaces for methods on iWildCam2020-wilds.

C.2 Hyperparameter grids for Camelyon17-wilds

We followed the hyperparameters used by Sagawa et al. [35] for their ERM experiments. In particular, we fixed the batch size to 168 and the learning rate to 0.0030693212138627936, which was selected in Sagawa et al. [35] after a random search over the distribution $10^{\text{Uni}(-4,-2)}$. For Camelyon17-wilds, we found that the choice of learning rate affected the relative ID vs. OOD accuracies of models, and we therefore standardized the learning rate across algorithms to remove it as a potential confounder for our experimental results. Separately tuning the learning rate for each algorithm did not significantly improve performance. For DANN, we used this learning rate for the featurizer and set the classifier learning rate to be $10\times$ higher, following Sagawa et al. [35]. We encountered optimization issues with adversarial discriminator training; to overcome this, we did a separate hyperparameter search for the discriminator learning rate and penalty strength λ, and selected the hyperparameter setting that resulted in the representation with the most invariant distributions across domains (as measured by a linear probe). We fixed the transform probability of all data augmentations to 1.0, since stochastically applying the augmentations did not seem to significantly affect performance on Camelyon17-wilds, and we took their other hyperparameter values from the original papers.

Because of the large variance in performance between random seeds for some algorithms on Camelyon17-wilds [1, 12], we ran 10 replicates per algorithm after selecting hyperparameters. The error bars were especially large for ERM, so we ran 50 replicates to ensure that we were accurately reporting its performance.

D Related work

Copy-paste augmentation. Copy-paste has previously been studied in object detection and image segmentation tasks, where it has increased ID performance [30, 31, 32, 33]. However, several

Method	Hyperparameters
Stain Color Jitter	Augmentation strength $\in$ {0.05, 0.1}
LISA	$\alpha = 2$ Interpolation method = CutMix
Vanilla MixUp	$\alpha = 0.2$
Vanilla CutMix	$\alpha = 0.5$
RandAugment	$k = 2$
DANN	Discriminator learning rate $\sim 10^{\text{Uni}(-4,-2)}$ $\lambda \sim 10^{\text{Uni}(-1,0)}$

Table 3: Hyperparameter search spaces for methods on CAMELYON17-WILDS.

works have found that its gains are limited, or even negative, in the ID object recognition setting [50, 51], though both of these works only study performance on ImageNet-9 [50]. Unlike ImageNet-9, IWILDCAM2020-WILDS evaluates both ID and OOD performance. It also contains an explicit "empty" class, so augmented examples use natural backgrounds from real examples in the dataset, whereas Xiao et al. [50], Ryali et al. [51] must segment, erase, and inpaint images to retrieve usable backgrounds. We find that, unlike prior work, Copy-Paste significantly boosts both ID and OOD performance on IWILDCAM2020-WILDS.

Copy-paste was also used by Beery et al. [6] to generate synthetic examples for minority classes in CCT-20, a small camera trap dataset with 15 classes and 20 cameras. They find that copy-paste gives a strong performance boost on both ID and OOD cameras; however, this work was on a smaller scale than IWILDCAM2020-WILDS.

The object detection and instance segmentation literature has disagreed as to whether to curate the backgrounds on which object foregrounds are pasted. Ghiasi et al. [30] set $B^{(i)} := \mathcal{D}_{\text{train}}$, i.e., any example may paste onto any other example, including ones already containing other objects. Dwibedi et al. [31] set $B^{(i)}$ to a separate set of empty images. Dvornik et al. [32], Fang et al. [33] argue $B^{(i)}$ should be a set of images that semantically concord with the object $x^{(i)}$, i.e., synthesized examples $\tilde{x}^{(i)}$ should appear realistic. These papers also disagree as to whether the foreground should be intelligently pasted onto images, i.e., whether foregrounds should be translated around the frame such that we avoid floating or incorrectly scaled objects.

Spurious correlations with image backgrounds in image classification. Empirical work has observed that models can learn to rely on spurious features for prediction, leading to a large ID-OOD drop; in image classification, background is a typical example given as a spurious correlation [5, 52, 53]. However, as we and Xiao et al. [50], Zhang et al. [54] find, background can also contain signal for prediction, such that removing backgrounds or swapping them indiscriminately significantly drops performance. Other works have investigated spurious correlations with other objects in the image frame [55, 56, 57]. Ryali et al. [51] experiment with copy-paste data augmentations to reduce reliance on background, but they find that copy-paste degrades performance when used for supervised learning, while it can help when used in contrastive learning.