Just Mix Once: Mixing Samples with Implicit Group Distribution

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent work has unveiled how average generalization frequently relies on superficial patterns in the data. The consequence are brittle models with poor performance in the presence of domain shift in group distribution at test time. When the subgroups in the train data are known we can use tools from robust optimization and regularization mechanism to tackle the problem. However group annotation and identification are daunting and time consuming tasks, seldom performed on large datasets. A recent line of research [12] is trying to solve this problem with implicit group distribution at train time, leveraging self-supervision and oversampling to improve generalization on minority groups. Following such ideas we propose a new class-conditional variant of mixup [22] for worst-group generalization, augmenting the train distribution with a continuous distribution of groups. Our method, called Just Mix Once, is domain agnostic, computationally efficient and performs on par or better than state-of-the-art on worst-group generalization.

1 Introduction

In supervised learning we fit a model $p(y|x;\theta)$ to maximize a relevant global metric over a dataset $D = \{x_s, y_s\}_{s=1}^{S}$. However, even when the classes are perfectly balanced, the optimization process can exploit spurious correlations between the target y and groups g in the data $D = \{x_s, g_s, y_s\}_{s=1}^{S}$. For example in the case of coloured MNIST (Fig 1), a model can use information about the color (spurious correlation with the target class) instead of the shape to classify a digit (i.e. all 1s are yellow) at training time. Then this bias will be used at test time with a strong decrease in generalisation performance in the presence of shift in the group distribution (1s are green and 3s are yellow). A powerful enough model [7] will exploit easy superficial correlations (texture, color, background, frequent patterns in the data) to solve a task if not constrained in some way [21, 8, 20, 2, 14, 5]. Similarly, in the presence of minority groups in a dataset, the model will tend to disregard such groups relying on patterns that are frequent in majority groups. Recent work has shown how to tackle this problem in a supervised [17] and self-supervised [13] way. We will focus on improving worst-group generalization with implicit group distribution in this work. Self-supervision and oversampling have been successfully employed to deal with implicit group distribution in Just Train Twice (JTT [12]).
Contribution. Following this line of work, our contributions are: **I)** We propose a simple mechanism, called Just Mix Once (JM1), based on a class-conditional variant of mixup [22] to improve generalization on minority groups in the data. **II)** Compared with JTT, our method does not rely on computationally expensive oversampling and the need to tune the oversampling rate. **III)** We perform extensive experiments with different levels of group annotation, demonstrating that JM1 outperforms the SOTA on vision and language datasets.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

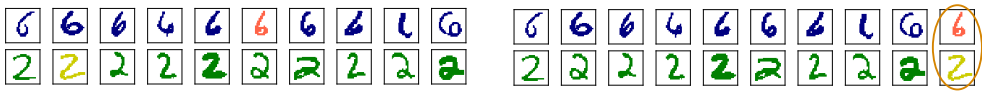

Figure 1: Group Identification Phase. Left: training data. Right: training data partitioned. In the first phase a self-supervised signal is exploited to split the data in two clusters or partitions. The per-sample loss magnitude in early stage training is used as signal: patterns that are frequent in the data (color, texture, background) are easy to model and have small loss. Patterns that are infrequent (minority groups, noisy samples) are difficult to model and will tend to have higher loss magnitude (samples in the orange circle). The assumption is that, among the samples identified as difficult, there will be also samples from minority groups because for the model a minority group in the data is equivalent to noise.

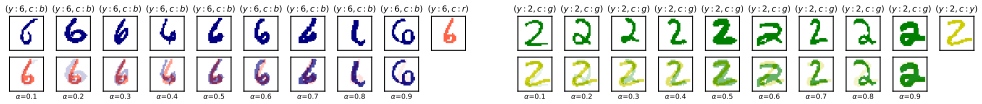

Figure 2: Class-conditional Mixing Phase. After the identification phase, we exploit label information to augment the training data, using a mixUp inspired strategy to augment samples from different partitions. Even if we do not have group information, we have label information for the data in the partitions. We can use the label information to select two samples from different partitions $(x_p^i, x_{\bar{p}}^i)$ where $y(x_p^i) = y(x_{\bar{p}}^i)$. Then we mix samples using $x_{mix} = \alpha\, x_p^i + (1-\alpha)\, x_{\bar{p}}^i$ in input or representation space. This simple mechanism gives us a principle and domain agnostic way to augment samples marginalizing group information in the data: sampling α we build a continuous distribution of groups in the data; doing so we augment the data with a large combinations of groups and the model cannot rely on frequent patterns because each sample has a slightly different pattern from all the other samples in the data.

2 Robust Classification with Implicit Group Distribution

In the presence of sub-populations or groups $g \in G$ in the data $D = \{x_i, y_i\}_{i=1}^N$, defined by tuples $g = (y, c)$ where y is a label and c is a confounding factor, we can define the following loss:

$$J(\theta, g) = \frac{1}{|G|} \sum_{g \in G} l_g(x_g, y_g) = \frac{1}{|G|} \sum_{g \in G} \frac{1}{n_g} \sum_{i=1}^{n_g} l_\theta(x_g^i, y_g^i), \tag{1}$$

where n_g is the cardinality of group g and can vary between groups. Notice that in case $n_g = 1$ for all groups, $|G| = N$ and Eq. 1 collapses to the standard Empirical Risk formulation: $J(\theta, g) = J_{\texttt{ERM}}$.

Just Train Twice ([12], JTT) presents a simple two stages approach to deal with sub-groups:
I) group identification - a first learner $f(\theta)$ is trained on the data using $J_{\texttt{ERM}}$ and samples with high loss [13] are saved in a buffer $B = \{x_b, y_b\}_{b=1}^B$. Ideally the goal is to partition the data in two cluster: one with majority groups (small loss, frequent superficial patterns); and one with minority groups (high loss, uncommon patterns).
II) group reweighting - a second learner is trained on a reweighted version of the data, where samples in B are oversampled $\lambda >> 1$ times. The loss is: $J_{\texttt{JTT}} = \lambda\, J_{\texttt{ERM}}(x_b, y_b) + J_{\texttt{ERM}}(x, y)$.
The core idea in phase I) is that samples that are difficult to model will have larger loss in the early stage of training. The underlying assumption is that among them there are samples from the minority groups. Intuitively, it is easier to fit large groups in the data than small one. This follows a common assumption that samples with high loss have also high chance to be from a minority or confounded group. In the I) phase is important to not overfit using small capacity models or early stopping. At test time we have interest in the maximum error on the worst-group (our minimization objective). This is different from the standard case, where we measure the average error among groups. **Limitations.** For JTT to work, the author emphasize that it is critical to choose two relevant hyper-parameters using the annotated validation set: re-weighting λ in the II) phase and number of identification epochs T in the I) phase. Another limitation of JTT is that oversampling a portion of the training data λ times makes training JTT slower than ERM and increases sample complexity.

3 Just Mix Once

We extend JTT and propose Just Mix Once (JM1). Our goal is to improve the classification accuracy on minorities groups without explicit annotation on the groups at training time. We can have annotated groups on a small validation set.

I) Group Identification. Similarly to JTT, in the I) phase (Fig. 1) we identify relevant samples from minority groups. Given the absence of explicit group annotation, we resolve to use a self-supervised signal [13, 1] based on the per-sample loss magnitude. The intuition is that a minority group can be seen as equivalent to noise in the early stage of training for a model.

II) Class-conditional Mixing. To perform better on minority groups at test time, we do not want to rely on spurious correlation during training (texture vs shape). We want to use a better augmentation to improve generalization for low density groups (Fig. 2). Our mixing strategy is inspired by the mixup strategy [18, 22], and we propose a novel class-conditional variant that achieves robust worst-group generalization. we empirically show that naively implementing the standard mixup fails to generalise to worst-group performance (Fig. 3). We mix samples from different partitions (i.e. majority vs minority groups) and same class. In the case of a convex interpolation [18, 22]:

$$z(h_g, h_{\bar{g}}, y) = \alpha\, h_g(y = y) + (1 - \alpha)\, h_{\bar{g}}(y = y),$$

where $h_{\bar{g}}$ is a sample identified as difficult in the first phase, h_g is an easy one from the same class y, and α is the mixing parameter (details of how to choose α are deferred to Sec. 4). Note that naive mixup without the proposed label constraint fails to achieve good generalization performance (Fig. 3).

How JM1 works. Given training samples from a distribution P, distributional robust optimization [17] defines an uncertainty set $U(P)$ over distributions that are within some distance of P, and then minimize the expected loss over the worst-case scenario in this uncertainty set. Given a finite number of groups in the training data, $g \in \{g_1, g_2, ..., g_K\}$, in distributional robust optimization [15] we aim to optimize: $J_{\texttt{DRO}} = \texttt{max}_g J(\theta, g)$. The uncertainty set is defined using grouping in the data. Using mixup with α continuous rate, we are effectively creating a continuous of new samples. Instead to consider such samples as samples from the same training distribution, we interpret them as samples from a continuous mixture $p(g; \alpha)$. The mixture has as limiting case the training distribution. In this view, each mixed sample can be seen as drawn from a different distribution parametrised by α. If we assume a perfect partition, each mixed sample is mixing two groups in the data, generating an infinite continuous of subpopulations. We can then write our loss as marginalization wrt the groups:

$$J_{\texttt{mix}}(\theta) = \int_g p(g; \alpha)\; J(\theta, g)\; dg = \mathbb{E}_{p(g;\alpha)}[J(\theta, g)] = \frac{1}{N} \sum_{i=1}^{N} \int_g p(g; \alpha)\; l(x_i, y_i, g)\; dg.$$

With a discrete number of groups, we collapse to the standard problem. Algorithmically we can approximate the expectation by MC sampling, draw a sample from $g \sim p(g)$, apply mixup [22, 18] with rate α to a sample (x_i, y_i): $z = \alpha\, x_i^g + (1 - \alpha)\, x_i^{\bar{g}}$, repeat the process for a mini-batch and optimize. In practice we are marginalizing wrt the group distribution that we build using mixup. For a visualization of JM1 in the presence of minority groups refer to Appendix D.

4 Experiments

We report experiment on the benchmark proposed in [17]. The benchmarks consists of two image and two language based datasets. In all four datasets there are confounders (background, sex, hair color) that strongly correlate with the label information. The goal is to improve worst-group performance for classification problems in vision and language without group annotation at train time. Just Train Twice (JTT, [12]) is SOTA for this benchmark. We sample α using a coupled strategy: we use $\alpha \sim U(0, 1)$ for half of the iterations. And $\alpha \sim Beta(2, 5)$ for the other half. This approach assures that the full domain [0, 1] is span; and at the same gives more importance to the identified samples in the mixing. Refer to Appendix E for experiment details.

Worst-Group Performance. Our approach is competitive with SOTA on improving worst-group performance. Our approach is scalable, domain agnostic, adding minimal computation overhead.

We remove the need for oversampling and λ. The main goal of JM1 is to improve the classification accuracy on the minority groups mixing samples from different groups with the same class. As for JTT, we use an annotated validation set to select the best model. Table 1 compares our approach with JTT. We select the best identification strategy for JTT and then use these samples for the second phase. In Table 2 we report the per-group accuracy on CUB for JTT and JM1.

Worst-Group with Oracle Groups. We expect JM1 to perform better than JTT when training with oracle groups. In Table 3 we consider an oracle identification setting where all the samples in the identified set are all and only the one with spurious correlations (minority and confounded groups). Without the identification phase the performance drops for JTT. This can be explained by the fact that we are oversampling too much the minority groups. However JM1 with class-conditional mixup is able to achieve good performance even in this setting. We also compare with mixup and a class-conditional mixup baseline (Fig. 3) where we mix samples from random groups - basically not using the identification phase. We see that standard mixup fails to improve worst-group performance, and our proposed mixing strategy is a simple yet effective approach when dealing with implicit domain shift in group distribution. JM1 outperforms the mixup baselines and JTT by a large margin, corroborating the idea that mixing groups is helpful for worst-group generalization.

Table 1: Results on CUB, CelebA and MultiNLI datasets for average accuracy and worst-group accuracy. JM1 performs better or comparable with the state-of-the-art on worst-group performance without using augmentation and additional hyper-parameters.

	CUB		CelebA		MultiNLI		group
	avg acc	worst acc	avg acc	worst acc	avg acc	worst acc	labels
ERM [12]	97.6	72.6	95.6	47.2	82.4	67.9	✗
CVaR-DRO [10]	96.5	69.5	82.4	64.4	82.0	68.0	✗
LfF [13]	97.3	75.2	86.0	70.6	80.8	70.2	✗
EIIL [5]	96.9	78.6	-	-	-	-	✗
JTT [12]	93.6	86.0	88.0	81.1	80.4	**72.3**	✗
JM1 (ours)	92.9	**87.8**	87.9	**84.4**	80.1	**72.0**	✗
GroupDRO [17]	93.5	91.4	92.9	88.9	81.4	77.7	✓

Table 2: Results on CUB dataset split by group. The goal is to classify land birds y_l vs water birds y_w in the presence of a confounders: land background c_l vs water background c_w. JM1 is competitive with JTT in terms of per-group generalization and improves accuracy on both minority groups (y_l, c_w) and (y_w, c_l)

	(y_l, c_l)	(y_l, c_w)	(y_w, c_l)	(y_w, c_w)	avg acc	worst acc
JTT	94.3	86.7	87.5	91.6	93.3	86.7
JM1	93.8	87.8	89.7	90.8	92.9	**87.8**

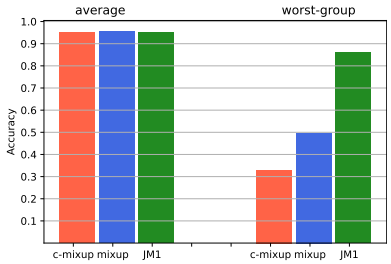

Figure 3: Results on CUB dataset upsampling only the minority groups (oracle groups) for JM1 vs class-conditional (c-mixup) and unconditional mixup baselines. Naive mixup does not improve worst-group accuracy.

Table 3: Results on CUB and CelebA dataset for average and worst-group accuracy. In the oracle configuration we have access to group annotation on the train set. JM1 with class conditional mixing generalizes better than JTT.

	CUB		CelebA	
	avg	worst	avg	worst
JTT (oracle) [12]	92.8	75.9	93.4	57.2
JM1 (oracle)	95.0	**86.1**	92.3	**85.6**

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pages 312–321. PMLR, 2019.

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

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

[4] Luigi Carratino, Moustapha Cissé, Rodolphe Jenatton, and Jean-Philippe Vert. On mixup regularization. *arXiv preprint arXiv:2006.06049*, 2020.

[5] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *International Conference on Machine Learning*, pages 2189–2200. PMLR, 2021.

[6] Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

[7] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[8] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

[9] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H Chi. Fairness without demographics through adversarially reweighted learning. *arXiv preprint arXiv:2006.13114*, 2020.

[10] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. *arXiv preprint arXiv:2010.05893*, 2020.

[11] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[12] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *International Conference on Machine Learning*, pages 6781–6792. PMLR, 2021.

[13] Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. *arXiv preprint arXiv:2007.02561*, 2020.

[14] Luca Oneto and Silvia Chiappa. Fairness in machine learning. *Recent Trends in Learning From Data*, pages 155–196, 2020.

[15] Yonatan Oren, Shiori Sagawa, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust language modeling. *arXiv preprint arXiv:1909.02060*, 2019.

[16] Judea Pearl. *Causality*. Cambridge university press, 2009.

[17] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

[18] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019.

[19] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research*, 20(1):2737–2778, 2019.

[20] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970. PMLR, 2017.

[21] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.

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

[23] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? *arXiv preprint arXiv:2010.04819*, 2020.

A Additional Experiments

Worst-Group without Validation Set. JTT and JM1 do not use group annotation on the train set, but rely on group annotation on the validation set. A natural question to investigate is if we can completely remove the need for group annotation and still perform well. In Table 4 we report results for three different identification scenarios (20, 40, 60 epochs) using the CUB dataset. In the second phase we select the JM1 model using average accuracy; and JTT using worst-group accuracy. JM1 without group annotation loses around 5 % compared to a JTT with access to group annotation on validation set. This is a preliminary result that indicates the potential of training accounting for subpopulations in the data in a fully self-supervised manner.

Table 4: Results on CUB dataset for average accuracy and worst-group accuracy without validation set. We fix the identification epochs in the first phase (20, 40, 60 epochs for identification). We select JM1 using average accuracy, and JTT using worst-group accuracy. JM1 deals well with the variety in the identified set. S: number of spurious correlation detected. NS: number of not spurious correlation detected. P: precision. R: recall.

	JM1		JTT-best		$\Delta\%$					
	avg	worst	avg	worst	avg	worst	S	NS	P	R
20id	92.3	72.9	78.9	67.5	+ 17.0	+ 8.0	107	290	0.269	0.445
40id	93.6	83.0	90.0	86.9	+ 4.0	- 4.5	115	153	0.429	0.479
60id	93.2	82.2	90.3	86.7	+ 3.2	- 5.2	110	128	0.462	0.458

B Related Work

Robust Classification and Implicit Group Distribution. Optimizing average accuracy with overparametrized neural networks tends to get rid of minority group information. This fact is related to how the model is trained and the way the model exploits superficial correlations in the data [7]. Recently approaches to preserve group information have been proposed [17] when full group annotation is available. However assuming sub-population annotation in large diverse datasets is unrealistic both in terms of time effort and most important because the process is prone to errors. Learning from Failure [13] proposes an annotation free and domain agnostic procedure to improve worst-group performance. The author propose a self-supervised approach to identify bias-aligned (majority) vs bias-contrastive (minority) groups in the data. The self-supervised signal consists of the per-sample loss magnitude in the early stage of training.. Then the dataset is re-weighted based on the loss magnitude. Inspired by this approach Just Train Twice [12], Environment Inference for Invariant Learning [5]

Equality of Opportunities in Machine Learning. Mixing information from different groups generates a mixture of distributions with a continuous amount of groups. Having an infinite collection of groups means that the model cannot rely on group information during training and samples in different groups should be considered similarly. Fairness constraints [20, 19], Invariant Risk Minimization [2], Causality [14]. In this setting we assume to have full annotation for the groups and robust optimization methods methods are employed. The supervised scenario is closely related to the fairness one: where with minority groups we want to improve the worst-group performance, in fairness we want to learn features and distributions that are invariant to different groups, using explicit constraints [20, 19], auxiliary tasks [6, 9], and causal structure [16, 14].

Self-Supervised Learning. Recently a large corpus of works dealt with self-supervised learning. With the amount and complexity of the data increasing exponentially, the availability of annotation is decreasing fast. Self-supervised learning tries to exploit structure in the data to cluster data and improve generalization. Loss magnitude [13, 7, 12, 1] can be used to cluster data. MixUp [22] is frequently employed to improve generalization and robustness to noise. The idea is to increase the variety in the data interpolating the training distribution in sample or representation space [18]. MixUp based methods have been extensively used for semi-supervised learning [11, 3, 1, 10] and improve generalization [23, 4].

C Benchmark

Figure 4: Figure adapted from [12]. Examples from the benchmarks. Label information y is spuriously correlated with confounding factors a.

Waterbirds (CUB). The task is to classify land birds y_l and water birds y_w. The label information is correlated with a confounding factor: land background c_l and water background c_w. The majority groups are (land bird, land background) - (y_l, c_l) and (water bird, water background) - (y_w, c_w). The minority groups are (land bird, water background) - (y_l, c_w) and (water bird, land background) - (y_w, c_l). During training the minority groups are present in 5% of the data. At test time the groups are evenly distributed in the data.

CelebA. The task is to classify blond person y_b and not blond person y_{nb}. The label information is correlated with a confounding factor: male c_m and female c_f. The majority groups are (not blond, male) - (y_{nb}, c_m) and (blond, female) - (y_b, c_f). The minority groups are (not blond, female) - (y_{nb}, c_f) and (blond, male) - (y_b, c_m).

MultiNLI. The task is to classify an inference procedure as entailment y_e, contradiction y_c and neutral y_n. The label information is correlated with a confounding factor: presence of a negation c_{neg} and absence of a negation c_{nneg} in the reasoning.

D Modelling the Augmented Group Distribution

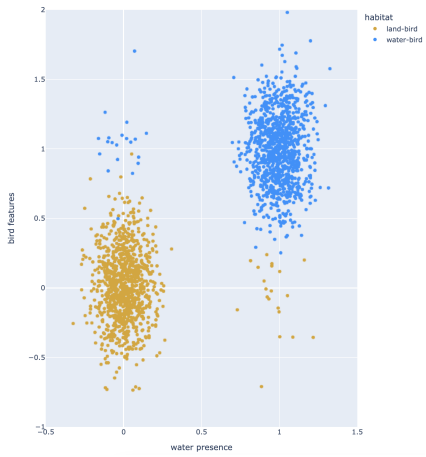

Figure 5: Source domain.

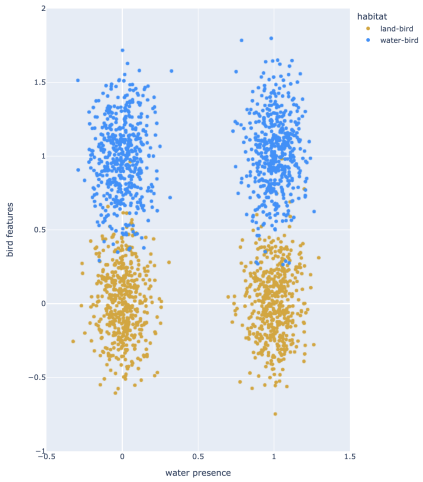

Figure 6: Target domain.

Mixing Groups. Consider a multiplicative interaction between group information and samples. Given a sample $\bar{x}$ from the train set, $x_o = \bar{x}\ g_o$ and $x_u = \bar{x}\ g_u$. Then a sample from the augmented

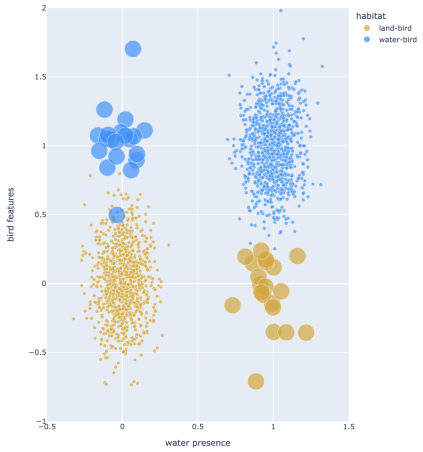

Figure 7: Oversampling (JTT).

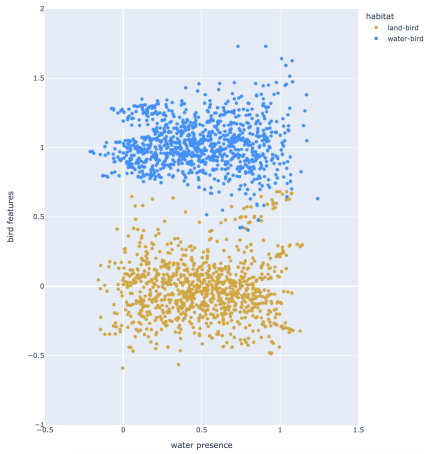

Figure 8: Mixing (JM1).

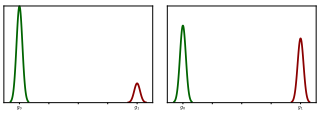

Figure 9: Train and test.

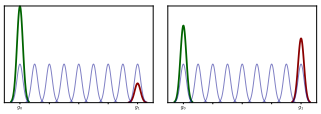

Figure 10: Augmented.

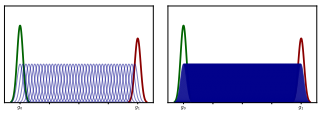

Figure 11: Augmented.

distribution is obtained using

$$x_{mix} = \alpha\, x_o + (1-\alpha)\, x_u = (\alpha\, g_o + (1-\alpha)\, g_u)\, \bar{x} = g_{mix}\, \bar{x}.$$

We can see that, given a continuous distribution for α, g_{mix} is a sample from a continuous distributions over groups $P_{aug}(g) = P_{mix}(g)$. It is also easy to see that a generic group sample from the mixing distribution will be closer to g_u than g_o because g_{mix} is a convex combination of the two groups.

$$g_{mix} = \alpha\, g_o + (1-\alpha)\, g_u.$$

E Experimental Details

Table 5: Training details for JM1.

	CUB	CelebA	MultiNLI	CUB (oracle)	CelebA (oracle)
Batch Size	32	64	32	32	64
Identification Epoch	40	1	2	-	-
Early stopping	✓	✓	✓	✓	✓
Epochs (I Phase)	200	40	4	-	-
Epochs (II Phase)	200	40	4	200	80
α	U/B	U/B	U	U/B	U/B
Learning Rate	1e-5	1e-5	1e-5	1e-5	1e-5
$\mathcal{L}$	CE	CE	CE	CE	CE
Number Groups	4	4	6	4	4
Number Classes	2	2	3	2	2
Mixing Layer	random	early layer	output	random	early layer
Pretrained Encoder	ResNet50	ResNet50	BERT	ResNet50	ResNet50
Weight Decay	1.0	0.1	0.1	1.0	0.1