Selective Mixup Helps with Distribution Shifts, But Not (Only) because of Mixup

Damien Teney [1] **Jindong Wang** [2] **Ehsan Abbasnejad** [3]

Abstract

Context. Mixup is a highly successful technique to improve generalization by augmenting training data with combinations of random pairs. Selective mixup is a family of methods that apply mixup to specific pairs e.g. combining examples across classes or domains. Despite remarkable performance on benchmarks with distribution shifts, these methods are still poorly understood.

Findings. We find that an overlooked aspect of selective mixup explains some of its success in a completely new light. The non-random selection of pairs affects the training distribution and improves generalization by means completely unrelated to the mixing. For example in binary classification, mixup across classes implicitly resamples the data to uniform class distribution — a classical solution to label shift. We verify empirically that this resampling explains some of the improvements reported in prior work. Theoretically, the effect relies on a "regression toward the mean", an accidental property we find in several datasets.

Outcomes. We now better understand why selective mixup works. This lets us predict a yet-unknown failure mode and conditions where the method is detrimental. We also use the equivalence to resampling to design variants with better combinations of mixing and resampling.

1. Introduction

Mixup and its variants are some of the few methods that improve generalization across tasks and modalities with no domain-specific information (Zhang et al., 2017). Standard mixup replaces training data with linear combinations of random pairs of examples, proving successful e.g. for image classification (Yun et al., 2019b), semantic segmentation (Islam et al., 2023), natural language processing (Verma et al., 2019), and speech processing (Meng et al., 2021).

This paper focuses on scenarios of distribution shift and variants of mixup that improve out-of-distribution (OOD) generalization. We examine the family of methods that apply mixup on selected pairs of examples, which we refer to as *selective mixup* (Hwang et al., 2022; Li et al., 2023; Lu et al., 2022a; Palakkadavath et al., 2022; Tian et al., 2023; Xu et al., 2020; Yao et al., 2022b). These methods propose various selection critera,[2] e.g. combining examples across classes (Yao et al., 2022b) (Figure 1) or across domains (Xu et al., 2020; Li et al., 2023; Lu et al., 2022a). These heuristics have claimed remarkable improvements on benchmarks such as DomainBed, WILDS, and Wild-Time (Gulrajani and Lopez-Paz, 2020; Koh et al., 2021; Yao et al., 2022a).

Despite impressive empirical performance, the theoretical mechanisms of selective mixup remain obscure. For example, the selection criteria in Yao et al. (2022b) include the selection of pairs of the same class / different domains but also the exact opposite. This raises these questions:

1. What makes each selection criterion suitable to any specific dataset?
2. Are there multiple mechanisms responsible for the improvements obtained with selective mixup?

This paper presents surprising answers, highlighting an overlooked side effect of selective mixup. **The non-random selection of pairs implicitly biases the training distribution, which can improve generalization by means completely unrelated to the mixing**. We verify empirically that forming mini-batches with the instances of the selected pairs (without mixing them) sometimes produces the same improvements as mixing them. This critical ablation was absent from prior studies.

We also analyze theoretically the resampling induced by different selection criteria. We find that conditioning on a "different attribute" (e.g. combining examples across classes or domains) brings the training distribution of this attribute

[1]Idiap Research Institute, Martigny, Switzerland [2]Microsoft Research Asia, Beijing, China [3]AIML, University of Adelaide, Australia. Correspondence to: Damien Teney <damien.teney@idiap.ch>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[2]We focus on the basic implementation of Yao et al. (2022b) with no modification to the standard learning objective.

Figure 1. Selective mixup is a family of methods that replace the training data with combined pairs of examples fulfilling a predefined criterion, e.g. pairs from different classes. An overlooked side effect is that this modifies the training distribution: here, sampling classes more uniformly. This explains much of the observed improvements in OOD generalization.

closer to a uniform one. Consequently, the imbalances in the data often "regress toward the mean" with selective mixup. We verify empirically that several datasets do indeed shift toward a uniform class distribution in their test split (see Figure 10). We also find remarkable correlation between improvements in performance and the reduction in divergence of training/test distributions due to selective mixup. **This also predicts a new failure mode of selective mixup when the above property does not hold** (see Section 4.7).

Our contributions are summarized as follows.

- We point out an implicit resampling effect when applying selective mixup (Section 3).
- We show theoretically that certain selection criteria induce a bias in the distribution of features and/or classes equivalent to a "regression toward the mean" (Theorem 3.1). In binary classification, selecting pairs across classes is equivalent to sampling uniformly over classes, the standard approach to address label shift/imbalance.
- We verify empirically that multiple datasets indeed contain a regression toward uniform class distribution between training → test splits (Section 4.6). We also find that improvements with selective mixup correlate with reductions in divergence of training/test distributions over labels and covariates. This supports resampling being a cause of the improvements. An interventional experiment (Section 4.7) further supports this causal effect.
- We compare selection and resampling criteria on five datasets. In all cases, improvements with selective mixup are partly or fully explained by resampling (Section 4).

Implications for future research are summarized as follows.

- We establish an equivalence between disconnected areas: selective mixup and resampling, a classical baseline for distribution shifts (Idrissi et al., 2022). Other methods from the label shift literature might be leveraged for future progress on OOD benchmarks (Garg et al., 2023).
- We now understand why different selective mixup criteria benefit different datasets: they affect feature/label distributions, addressing covariate/label shift respectively.
- There is a **risk of overfitting to the benchmarks**, since much of the improvements obtained rely on an accidental "regression toward the mean" in the datasets.

2. Background: mixup and selective mixup

Notations. We consider a classification model $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to [0,1]^C$ of learned parameters $\boldsymbol{\theta}$. It maps an input vector $\boldsymbol{x} \in \mathbb{R}^d$ to a vector $\boldsymbol{y}$ of scores over C classes. The training data is typically a set of labeled examples $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i, d_i)\}_{i=1}^n$ where $\boldsymbol{y}_i$ are one-hot vectors encoding ground-truth labels, and $d_i \in \mathbb{N}$ are optional domain indices (e.g. image styles in Li et al. (2017) or time periods in Koh et al. (2021)).

Training with ERM. Standard empirical risk minimization (ERM) optimizes the model's parameters for $\min_{\boldsymbol{\theta}} \mathcal{R}(f_{\boldsymbol{\theta}}, \mathcal{D})$. For a loss $\mathcal{L}$, the training risk is:

$$\mathcal{R}(f_{\boldsymbol{\theta}}, \mathcal{D}) \;=\; \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}}\; \mathcal{L}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}\big). \tag{1}$$

An empirical estimate is obtained with an arithmetic mean over instances of the dataset $\mathcal{D}$.

Training with mixup. Standard mixup replaces training examples with linear combinations of random pairs in input and label space. This redefines the training risk as:

$$\mathcal{R}_{\text{mixup}}(f_{\boldsymbol{\theta}}, \mathcal{D}) \;= \\ \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}}\; \mathcal{L}\big(f(c\,\boldsymbol{x}+(1-c)\widetilde{\boldsymbol{x}},\; c\,\boldsymbol{y}+(1-c)\,\widetilde{\boldsymbol{y}})\big) \\ \text{with mixing coefficients } c \sim \mathcal{B}(2,2) \\ \text{and paired examples } (\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}) \sim \mathcal{D}. \tag{2}$$

The expectation is approximated by sampling coefficients and pairs at every training iteration.

Selective mixup. Standard mixup combines random pairs. Selective mixup only combines pairs that fulfil a predefined criterion. To select these pairs, the method starts with the original data $\mathcal{D}$, then for every $(\boldsymbol{x}, \boldsymbol{y}, d) \in \mathcal{D}$ it selects $(\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{y}}, \widetilde{d}) \in \mathcal{D}$ to they verify the predicate $\text{Paired}(\cdot,\cdot)$. For example, the criterion *same class, different domain* ("intra-label LISA" in Yao et al. (2022b)) is implemented as:

$$\text{Paired}\big((\boldsymbol{x}_i, \boldsymbol{y}_i, d_i), (\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{y}}_i, \widetilde{d}_i)\big) = \textit{true} \\ \text{iff } (\widetilde{\boldsymbol{y}} = \boldsymbol{y}) \wedge (\widetilde{d} \neq d) \quad \textit{(same class, diff. domain)} \tag{3a}$$

Other examples:

$$\text{Paired}\big((\boldsymbol{x}_i, \boldsymbol{y}_i, d_i), (\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{y}}_i, \widetilde{d}_i)\big) = \textit{true} \\ \text{iff } (\widetilde{\boldsymbol{y}} \neq \boldsymbol{y}) \quad \textit{(different class)} \tag{3b}$$

$$\text{Paired}\big((\boldsymbol{x}_i, \boldsymbol{y}_i, d_i), (\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{y}}_i, \widetilde{d}_i)\big) = \textit{true} \\ \text{iff } (\widetilde{d} = d) \quad \textit{(same domain)} \tag{3c}$$

3. Theoretical predictions: selective mixup modifies the training distribution

The new claims of this paper comprise two parts.

1. Estimating the training risk with selective mixup (Eq. 2) uses a different sampling of examples from $\mathcal{D}$ than ERM (Eq. 1). We demonstrate this theoretically below.
2. We hypothesize that this different sampling of training examples influences the generalization properties of the learned model, regardless of the mixing operation. We verify this empirically in Section 4 using ablations of selective mixup that omit the mixing operation — a critical baseline absent from prior studies.

Training distribution. This distribution refers to the examples sampled from $\mathcal{D}$ to estimate the training risk (Eq. 1 or 2) — whether these are then mixed or not. The following discussion focuses on distributions over classes ($\boldsymbol{y}$) but analogous arguments apply to covariates ($\boldsymbol{x}$) and domains (d).

With ERM, the training distribution equals the dataset distribution because the expectation in Eq. (1) is over uniform samples of $\mathcal{D}$. We obtain an empirical estimate by averaging all one-hot labels, giving the vector of discrete probabilities $\mathbf{p}_{\mathrm{Y}}(\mathcal{D}) = \oplus_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \boldsymbol{y} \,/\, |\mathcal{D}|$ with $\oplus$ the element-wise sum.

With selective mixup, evaluating the risk (Eq. 2) requires pairs of samples. The first element of a pair is sampled uniformly, yielding the same $\mathbf{p}_{\mathrm{Y}}(\mathcal{D})$ as ERM. The second element is selected as described above, using the first element and one chosen predicate $\mathrm{Paired}(\cdot,\cdot)$ e.g. from (3a–3c). For our analysis, we denote these "second elements" of the pairs as the virtual data:

$$\widetilde{\mathcal{D}} = \big\{(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{y}}_i, \widetilde{d}_i) \sim \mathcal{D} : \mathrm{Paired}\big((\boldsymbol{x}_i, \boldsymbol{y}_i, d_i), (\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{y}}_i, \widetilde{d}_i)\big) = \textit{true},\ \forall\, i\big\}. \quad (4)$$

We can now analyze the overall training distribution of selective mixup. An empirical estimate is obtained by combining the distributions from the two elements of the pairs, which gives the vector $\mathbf{p}_{\mathrm{Y}}(\mathcal{D}\cup\widetilde{\mathcal{D}}) = (\mathbf{p}_{\mathrm{Y}}(\mathcal{D}) \oplus \mathbf{p}_{\mathrm{Y}}(\widetilde{\mathcal{D}}))\,/\,2$.

Regression toward the mean. With the criterion *same class*, it is obvious that $\mathbf{p}_{\mathrm{Y}}(\widetilde{\mathcal{D}}) = \mathbf{p}_{\mathrm{Y}}(\mathcal{D})$. Therefore these variants of selective mixup are not concerned with resampling effects.[3] In contrast, the criteria *different class* or *different domain* do bias the sampling. In the case of binary classification, we have $\mathbf{p}_{\mathrm{Y}}(\widetilde{\mathcal{D}}) = 1 - \mathbf{p}_{\mathrm{Y}}(\mathcal{D})$ and therefore $\mathbf{p}_{\mathrm{Y}}(\mathcal{D}\cup\widetilde{\mathcal{D}})$ is uniform. This means that selective mixup with the *different class* criterion has the side effect of balancing the training distribution of classes, a classical mitigation of class imbalance (Japkowicz, 2000; Kubat et al., 1997). For multiple classes, we have a more general result.

Theorem 3.1. *Given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_i$ and paired data $\widetilde{\mathcal{D}}$ sampled according to the "different class" criterion, i.e. $\widetilde{\mathcal{D}} = \{(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{y}}_i) \sim \mathcal{D}$ s.t. $\widetilde{\boldsymbol{y}}_i \neq \boldsymbol{y}_i\}$, then the distribution of classes in $\mathcal{D}\cup\widetilde{\mathcal{D}}$ is more uniform than in $\mathcal{D}$. Formally, the entropy $\mathbb{H}\big(\mathbf{p}_{\mathrm{Y}}(\mathcal{D})\big) \leq \mathbb{H}\big(\mathbf{p}_{\mathrm{Y}}(\mathcal{D}\cup\widetilde{\mathcal{D}})\big)$.*
Proof: see Appendix D.

Theorem 3.1 readily extends in two ways. First, the same effect also results from the *different domain* criterion: if each domain contains a different class distribution, the resampling from this criterion averages them out, yielding a more uniform aggregated training distribution. Second, this averaging applies not only to class labels ($\boldsymbol{y}$) but also covariates ($\boldsymbol{x}$). An analysis using distributions is ill-suited but the mechanism similarly affects the sampling of covariates when training with selective mixup.

When does one benefit from the resampling (regardless of mixup)? The above results mean that selective mixup can implicitly reduce imbalances (a.k.a. biases) in the training data. When these are not spurious and also exist in the test data, effects on predictive performance could be detrimental.

We expect benefits (verified in Section 4) on datasets with distribution shifts. By definition, their training/test splits contain different imbalances. Softening imbalances in the training data is then likely to bring the training and test distributions closer, in particular with extreme shifts such as the complete reversal of a spurious correlation (e.g. *waterbirds* dataset, see Section 4.1).

We also expect benefits on worst-group metrics (e.g. *civilComments* dataset, see Section 4.5). The challenge in these datasets comes from the imbalance of class/domain combinations. Prior work has indeed shown that balancing is beneficial (Idrissi et al., 2022; Sagawa et al., 2019).

4. Empirical Verification

We performed a large number of experiments to understand the contribution of the different effects of selective mixup and other resampling baselines (complete results in Appendix C).

Datasets. We focus on five datasets that previously showed improvements with selective mixup. We selected them to cover a range of modalities (vision, NLP, tabular), settings (binary, multiclass), and types of shifts (covariate, label, and subpopulation shifts).

- **Waterbirds** (Sagawa et al., 2019) is a popular artificial dataset used to study distribution shifts. The task is to classify images of birds into two types. The image backgrounds are also of two types, and the correlation between birds and backgrounds is reversed across the training and test splits. The type of background in each image serves as its domain label.

[3] The absence of resampling effects holds for *same class* and *same domain* alone, but not in conjunction with other criteria. Compare ***same domain / diff. class*** with ***any domain / diff. class*** in Figure 3 for example.

- **CivilComments** (Koh et al., 2021) is a widely-used dataset of online text comments to be classified as toxic or not. Each example is labeled with a topical attribute (e.g. Christian, male, LGBT, etc.) that is spuriously associated with ground truth labels in the training data. These attributes serve as domain labels. The target metric is the worst-group accuracy where the groups correspond to all toxicity/attribute combinations.
- **Wild-Time Yearbook** (Yao et al., 2022a) contains yearbook portraits to be classified as male or female. It is part of the Wild-Time benchmark, which is a collection of real-world datasets captured over time. Each example belongs to a discrete time period that serves as its domain label. Distinct time periods are assigned to the training and OOD test splits (see Figure 10).
- **Wild-Time arXiv** (Yao et al., 2022a) contains titles of arXiv preprints. The task is to predict each paper's category among 172 classes. Time periods serve as domain labels.
- **Wild-Time MIMIC-Readmission** (Yao et al., 2022a) contains hospital records (sequences of codes representing diagnoses and treatments) to be classified into two classes. The positive class indicates the readmission of the patient at the hospital within 15 days. Time periods serve as domain labels.

Methods. We train standard architectures with the methods below (details in Appendix B). We perform early stopping i.e. reporting metrics of each run at the epoch of highest ID or worst-group validation performance (for *Wild-Time* and *waterbirds*/*civilComments* datasets respectively). We plot the average of these metrics in bar charts over 9 different seeds with error bars representing $\pm$ one standard deviation. **ERM** and **vanilla mixup** are the standard baselines. The **resampling** baselines use training examples with equal probability from each class, domain, or combinations thereof (Idrissi et al., 2022; Sagawa et al., 2019). **Selective mixup** (■) is run with all possible selection criteria based on classes and domains. We avoid ambiguous terminology from earlier works because of inconsistent usage (e.g. "intra-label LISA" means "different domain" in Koh et al. (2021) but not in Yao et al. (2022a)). **Selective sampling** (■) is a novel ablation of selective mixup where the selected pairs are not mixed, but the instances are appended one after another in the mini-batch. Half are dropped at random to keep the mini-batch size identical to the other methods. Therefore any difference between selective sampling and ERM is attributable only to resampling effects. We also include **novel sampling/mixup combinations** (■). For each dataset, we tune the distribution of mixing coefficients (c in Eq. 2) by cross-validation for vanilla mixup. We found little variation across choices e.g. $\mathcal{B}(2, 2)$ or even a constant $c = 0.5$.

4.1. Results on the *waterbirds* dataset

The target metric for this dataset is the worst-group accuracy, with groups defined as the four class/domain combinations. The two difficulties are (1) a class imbalance (77 / 23%) and (2) a correlation shift (spurious class/domain association reversed at test time).

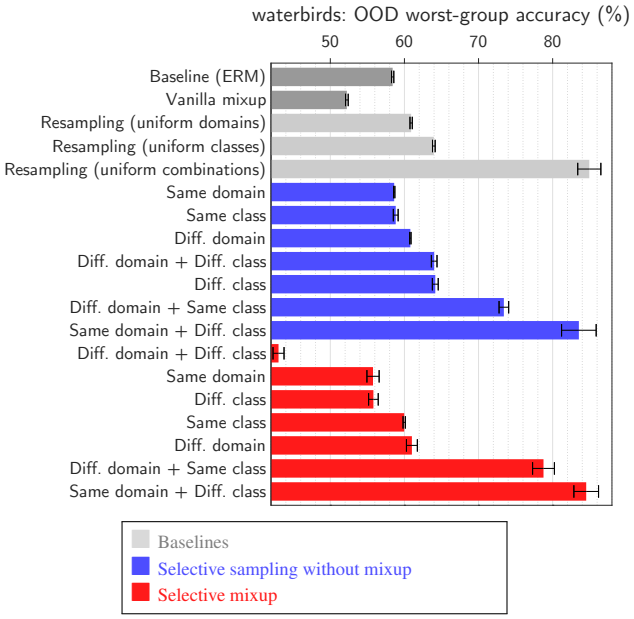

Figure 2. Main results on *waterbirds*. We first observe that vanilla mixup is detrimental over ERM. Resampling with uniform class/domain combinations is hugely beneficial, for the reasons explained in Figure 3. The ranking of selective sampling criteria is similar with or without mixup. Most interestingly, the best criterion performs similarly (no better) than the best resampling.

On this dataset, these results suggest that **the excellent performance of the best version of selective mixup is entirely due to resampling**. The efficacy of resampling on this dataset is not a new finding (Idrissi et al., 2022; Sagawa et al., 2019). What is new is its equivalence with the best variant of selective mixup. Figure 3 further supports this claim by comparing proportions of classes and domains sampled by each method.

4.2. Results on the *yearbook* dataset

The difficulty of this dataset comes from a slight class imbalance and the presence of covariate/label shift (see Figure 10). The test split contains several domains (time periods). The target metric is the worst-domain accuracy. Figure 5 shows that vanilla mixup is slightly detrimental compared to ERM. Resampling for uniform classes gives a clear improvement because of the class imbalance. With selective sampling (no mixup), the only criteria that improve over ERM contain "different class". This is expected because this criterion implicitly resamples for a uniform class distribution.

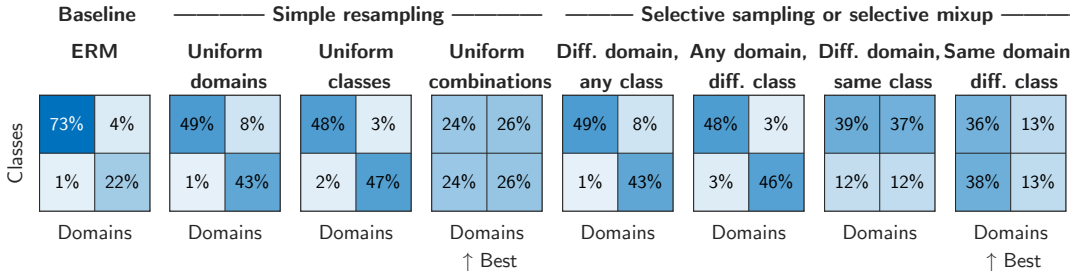

Figure 3. On *waterbirds*, the sampling ratios of classes/domains explain the performance of the best variants of resampling and selective mixup. **Resampling uniform combinations** gives them equal weights, just like the worst-group metric. **Selective mixup with same domain / diff. class** gives equal weights to the classes but also breaks the spurious pattern between groups and classes, unlike any other criterion.

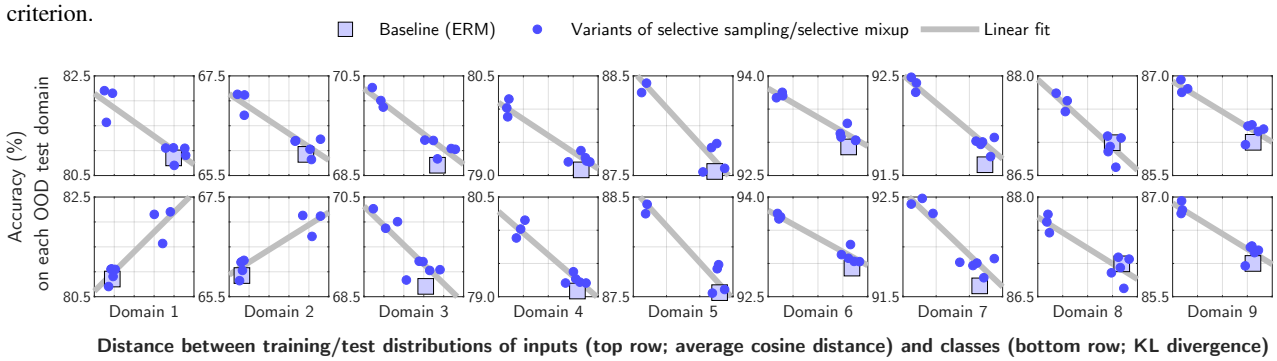

Figure 4. On *yearbook*, different selection criteria (•) modify the distribution of both covariates and labels (upper and lower rows). The resulting reductions in divergence between training and test distributions correlate remarkably well with test performance.[3] This confirms the contribution of resampling to the overall performance of selective mixup.

To investigate whether some of the improvements are due to resampling, we measure the divergence between training and test distributions of classes and covariates (details in Appendix B). Figure 4) shows first that there is a clear variation among different criteria (• blue dots) i.e. some bring the training/test distributions closer to one another. Second, there is a remarkable correlation between the test accuracy and the divergence, on both classes and covariates.[4] This means that resampling effects do occur and also play a part in the best variants of selective mixup.

Finally, the improvements from simple resampling and the best variant of selective mixup suggest a new combination. We train a model with uniform class sampling and selective mixup using the "same class" criterion, and obtain performance superior to all existing results (last row in Figure 4). This confirms the **complementarity of the effects of resampling and within-class selective mixup**.

4.3. Results on the *arXiv* dataset

This dataset has difficulties similar to *yearbook* and also many more classes (172). Simple resampling for uniform classes is very bad (literally off the chart in Figure 6) because it overcorrects the imbalance (the test distribution being closer to the training than to a uniform one). Uniform *domains* is much better since its effect is similar but milder.

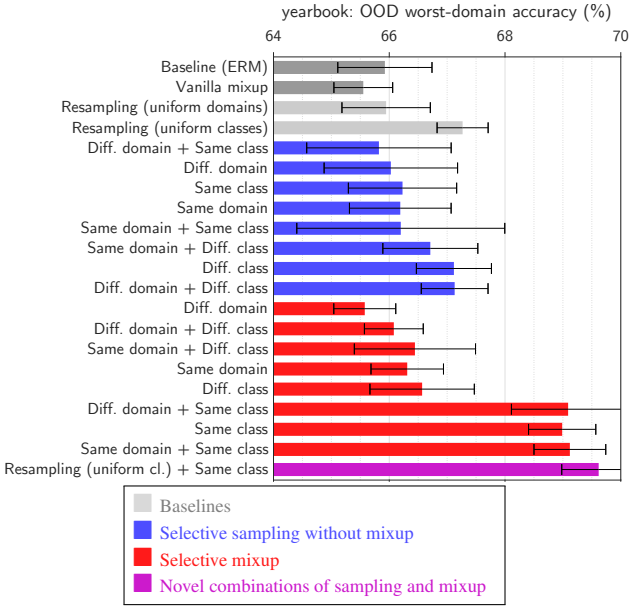

Figure 5. Main results on *yearbook*. With selective mixup, the "different class" criterion is not useful, but "same class" performs significantly better than ERM. Since this criterion alone does not have resampling effects, it indicates a genuine benefit from mixup restricted to pairs of the same class.

[4] As expected, the correlation is reversed for the first two test domains in Figure 4 because they are even further from a uniform class distribution than the average of the training data, as seen in Figure 10.

All variants of selective mixup (■) perform very well, but they improve over ERM even without mixup (■). And the selection criteria rank similarly with or without mixup, suggesting that parts of the improvements of selective mixup is due to the resampling. Given that vanilla mixup also clearly improves over ERM, the performance of **selective mixup is explained by cumulative effects of vanilla mixup and resampling effects**. This also suggests new combinations of methods (■) among which we find one version marginally better than the best variant of selective mixup (last row).

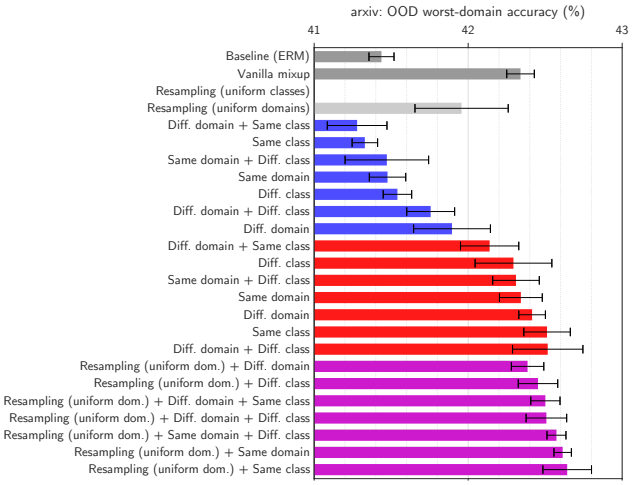

Figure 6. Main results on *arXiv*. To investigate the contribution of resampling, we measure the divergence between training/test class distributions and plot them against the test accuracy (Figure 9). We observe a strong correlation across methods. Mixup essentially offsets the performance by a constant factor. This suggests again the independence of the effects of mixup and resampling.

4.4. Results on the *MIMIC-Readmission* dataset

This dataset contains a class imbalance (about 78/22% in training data), label shift (the distribution being more balanced in the test split), and possibly covariate shift. It is unclear whether the task is causal or anticausal (labels causing the features) because the inputs contain both diagnoses and treatments. The target metric is the area under the ROC curve (AUROC) which gives equal importance to both classes. We report the worst-domain AUROC, i.e. the lowest value across test time periods.

Vanilla mixup performs a bit better than ERM. Because of the class imbalance, resampling for uniform classes also improves ERM. As expected, this is perfectly equivalent to the selective sampling criterion "diffClass" and they perform therefore equally well. Adding mixup is yet a bit better, which suggests again that **the performance of selective mixup is merely the result of the independent effects of vanilla mixup and resampling**. We further verify this explanation with the novel combination of simple resampling and vanilla mixup, and observe almost no difference whether the mixing operation is performed or not (last two rows in Figure 7).

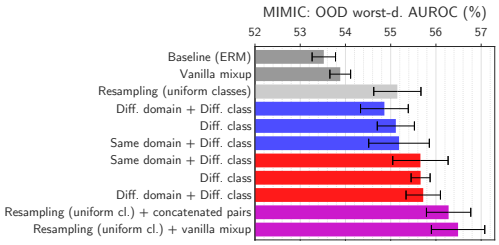

Figure 7. Main results on *MIMIC-Readmission*.

Table 1. The performance of the various methods on *MIMIC-Readmission* is explained by their correction of a class imbalance. The best training methods (boxed numbers) sample the majority class in a proportion much closer to that of the test data.

Proportion of majority class	**(%)**
In the dataset (training)	78.2
In the dataset (validation)	77.8
In the dataset (OOD test)	66.5
Sampled by different training methods	
Resampling (uniform classes)	50.0
Diff. domain + diff. class	50.0
Diff. class	50.1
Same domain + Diff. class	49.9
Resampling (uniform cl.) + concatenated pairs	64.3
Resampling (uniform cl.) + vanilla mixup	64.3

To further support the claim that these methods mostly address label shift, we report in Table 1 the proportion of the majority class in the training and test data. We observe that the distribution sampled by the best training methods brings it much closer to that of the test data.

4.5. Results on the *civilComments* dataset

This dataset mimics a subpopulation shift because the worst-group metric requires high accuracy on classes and domains under-represented in the training data. It also contains an implicit correlation shift because any class/domain association (e.g. "*Christian*" comments labeled as toxic more often than not) becomes spurious when evaluating individual class/domain combinations.

4.6. Checking the *regression toward the mean* in the data

We hypothesized in Section 3 that resampling helps because of a "regression toward the mean" between training and test splits. We now check for this property. We find indeed a shift toward uniform class distribution in all datasets studied. For Wild-Time datasets, we plot in Figure 10 the ratio of the minority class (for binary tasks: *yearbook*, *MIMIC*) and class distribution entropy (for the multiclass task: *arXiv*).

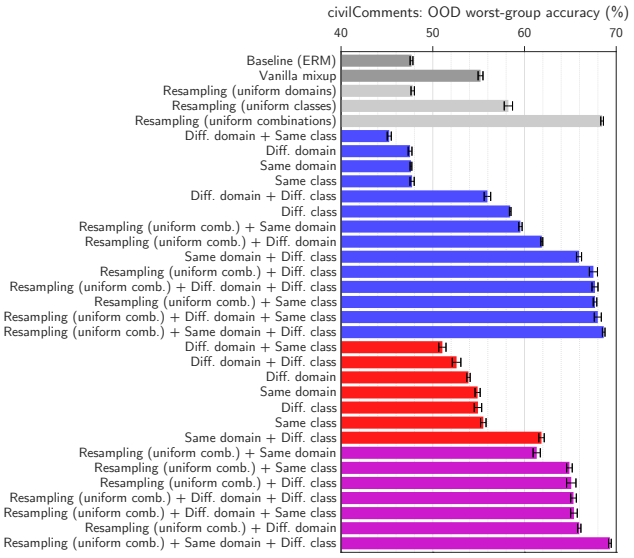

Figure 8. Main results on *civilComments*. Given the reasons stated in Section 4.5, it makes sense that resampling for uniform classes or combinations greatly improves performance, as shown in prior work (Idrissi et al., 2022). With selective mixup (■), some criterion (same domain/diff. class) performs clearly above all others. But it works **even better without mixup**! (■) Among many other variations, **none surpasses the uniform-combinations baseline**.

Finding this property agrees with the proposed explanation and with the fact that we selected datasets that showed improvements from selective mixup in Yao et al. (2022a).

The shift toward uniformity also holds in *waterbirds* and *civilComments*, artificially through the worst-group metric. The training data contains imbalanced groups (class/domain combinations) while the worst-group accuracy gives uniform importance to all groups.

4.7. New failure mode

The proposed theory implies that the resampling is beneficial because a "regression toward the mean" is present in the datasets. As a corollary, it implies that the effect would be detrimental if the opposite property holds (i.e. increased imbalance from training → test data).

We test this prediction on the *yearbook* dataset by switching the ID and OOD data. The original dataset uses data from years 1930–1970 as training data and ID test data (as shown in Figure 10), we now use this data as the OOD test data. Vice versa for data from years 1970–2010. As a result, the training → test label shift is now an increased imbalance rather than a regression toward a uniform one. The results on *yearbook-reversed* confirm the predicted failure (cf. Figures 12–13 in Appendix). Methods with improved performance on the original dataset are now detrimental.

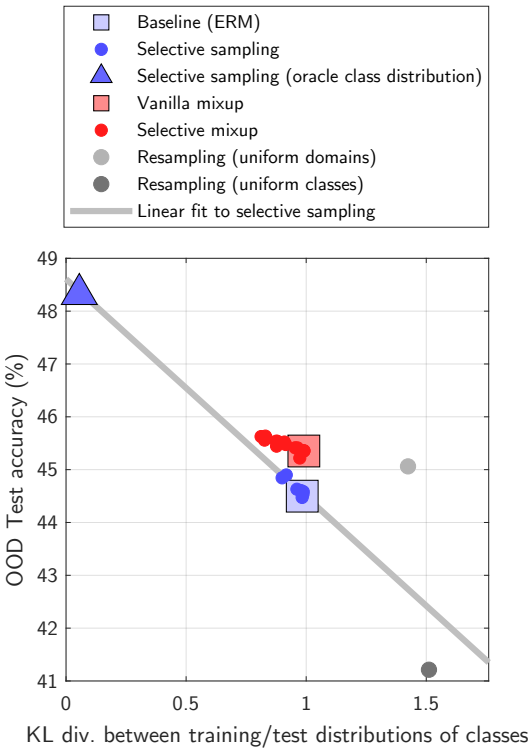

Figure 9. Divergence of training/test class distributions vs. test accuracy (*arXiv*). The resampling baselines (●●) also roughy agree with a linear fit to the "selective sampling" points. We therefore hypothesize that **all these methods are mostly addressing label shift**. We verify this hypothesis with the remarkable fit of an additional point (▲) of a model trained by resampling according to the test set class distribution, i.e. cheating. It represents an upper bound that might be achievable in future work with methods for label shift (Azizzadenesheli et al., 2019; Lipton et al., 2018). We replicated these observations on every test domain (Figure 15 in the appendix).

5. Related work

Mixup and variants. Mixup was originally introduced in Zhang et al. (2017) and numerous variants followed (Cao et al., 2022). Many propose modality-specific mixing operations: CutMix (Yun et al., 2019a) replaces linear combinations with collages of image patches, Fmix (Harris et al., 2020) combines image regions based on frequency contents, AlignMixup (Venkataramanan et al., 2022) combines images after spatial alignment. Manifold-mixup (Verma et al., 2019) mixes learned representations rather than raw inputs, making it applicable e.g. to text embeddings.

Mixup for OOD generalization. Mixup has been integrated into existing techniques for domain adaptation (DomainMix (Xu et al., 2020)), domain generalization (FIXED (Lu et al., 2022b)), and with meta learning (Reg-Mixup (Pinto et al., 2022)). This paper focuses on variants

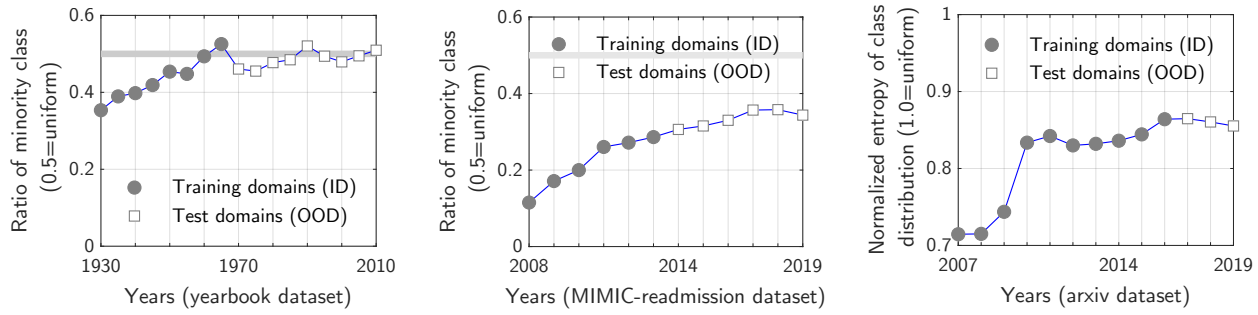

Figure 10. The class distribution shifts toward uniformity in these Wild-Time datasets. This agrees with the explanation that the benefits from resampling rely on a "regression toward the mean".

we call "selective mixup" that use non-uniform sampling of the pairs of mixed examples. LISA (Yao et al., 2022b) proposes two heuristics, same-class/different-domain and vice versa, used in proportions tuned by cross-validation on each dataset. Palakkadavath et al. (2022) use same-class pairs and an additional objective to encourage invariance of the representations to the mixing. CIFair (Tian et al., 2023) uses same-class pairs with a contrastive objective to improve algorithmic fairness. SelecMix (Hwang et al., 2022) proposes a selection heuristic to handle biased training data: same class/different biased attribute, or vice versa. DomainMix (Xu et al., 2020) uses different-domain pairs for domain adaptation. DRE (Li et al., 2023) uses same-class/different-domain pairs and regularize their Grad-CAM explanations to improve OOD generalization. SDMix (Lu et al., 2022a) applies mixup on examples from different domains with other improvements to improve cross-domain generalization for activity recognition.

Explaining the benefits of mixup has invoked regularization (Zhang et al., 2020) and augmentation (Kimura, 2021) effects, the introduction of label noise (Liu et al., 2023), and the learning of rare features (Zou et al., 2023). These works focus on the mixing and in-domain generalization, whereas we focus on the selection and OOD generalization.

Training on resampled data. We find that selective mixup is sometimes equivalent to training on resampled or reweighted data. Both are standard tools to handle distribution shifts in a domain adaptation setting (Japkowicz, 2000; Kubat et al., 1997) and are also known as importance-weighted empirical risk minimization (IW-ERM) (Shimodaira, 2000; Gretton et al., 2009). For covariate shifts, IW-ERM assigns each training point $\boldsymbol{x}$ of label $\boldsymbol{y}$ a weight equal to the likelihood ratio $p_{\text{target}}(\boldsymbol{x})/p_{\text{source}}(\boldsymbol{x})$, and for label shifts, $p_{\text{target}}(\boldsymbol{y})/p_{\text{source}}(\boldsymbol{y})$ (Azizzadenesheli et al., 2019; Lipton et al., 2018), Several works recently showed that reweighting and resampling are competitive with the state of the art in various OOD (Idrissi et al., 2022; Park et al., 2022; Perrett et al., 2023; Sagawa et al., 2019) and label-shift settings (Garg et al., 2023).

6. Conclusions and open questions

In conclusion, the experiments show that the effects of mixup and resampling are additive and largely independent. There is no single best method since different datasets benefit differently from mixing and/or resampling. Overall, this paper helps understand selective mixup, one of the most successful and general methods for distribution shifts. Since part of the improvements are unrelated to the mixing, they can be obtained with much simpler, well-known resampling methods. On datasets where mixup does bring benefits, we can now obtain even better results by combining independent effects of the best mixup and resampling variants.

The mixing is still sometimes useful, right? Yes indeed (see *yearbook*, *arxiv*, *civilComments*). The effects of mixup and resampling are additive and largely independent. Different datasets benefit differently from one and/or the other.

Is the mixing sometimes harmful? Yes, as seen in Figure 2 for example: vanilla mixup is worse than the baseline on the *waterbirds* dataset. It was already known empirically that vanilla mixup is not always beneficial.

Limitations. We focused on the simplest version of selective mixup from Yao et al. (2022b). Other works proposed modifications to the learning objective, whose interplay with the resampling is unknown (Hwang et al., 2022; Li et al., 2023; Lu et al., 2022a; Palakkadavath et al., 2022; Tian et al., 2023; Xu et al., 2020). We evaluated "only" five datasets, but we hope to see re-evaluations of others, since we showed that simple ablations can single out resampling effects.

Open questions. Our results leave open the question of the applicability of selective mixup to real situations. The "regression toward the mean" explanation indicates that much of the observed improvements are accidental since they rely on an artefact of some datasets. This is a reminder of the risk of overfitting to popular benchmarks (Liao et al., 2021) and the relevance of Goodhart's law to machine learning (Teney et al., 2020) ("*When a measure becomes a target, it ceases to be a good measure*").

Impact Statement

The overarching goal of this work is to improve generalization, which is a desirable property of any machine learning system. More specifically, we contribute to the understanding an existing family of methods and help pinpoint their limitations. We identify new failure modes and conditions in which these methods would be harmful. We therefore see these contributions as overwhelmingly positive.

References

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734*, 2019.

Chengtai Cao, Fan Zhou, Yurou Dai, and Jianping Wang. A survey of mix-based data augmentation: Taxonomy, methods, applications, and explainability. *arXiv preprint arXiv:2212.10888*, 2022.

Saurabh Garg, Nick Erickson, James Sharpnack, Alex Smola, Sivaraman Balakrishnan, and Zachary C Lipton. Rlsbench: Domain adaptation under relaxed label shift. *arXiv preprint arXiv:2302.03020*, 2023.

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 2009.

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

Ethan Harris, Antonia Marcu, Matthew Painter, Mahesan Niranjan, Adam Prügel-Bennett, and Jonathon Hare. Fmix: Enhancing mixed sample data augmentation. *arXiv preprint arXiv:2002.12047*, 2020.

Inwoo Hwang, Sangjun Lee, Yunhyeok Kwak, Seong Joon Oh, Damien Teney, Jin-Hwa Kim, and Byoung-Tak Zhang. Selecmix: Debiased learning by contradicting-pair sampling. *arXiv preprint arXiv:2211.02291*, 2022.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, 2022.

Md Amirul Islam, Matthew Kowal, Konstantinos G Derpanis, and Neil DB Bruce. Segmix: Co-occurrence driven mixup for semantic segmentation and adversarial robustness. *International Journal of Computer Vision*, 131(3): 701–716, 2023.

Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proc. of the Int'l Conf. on artificial intelligence*, volume 56, pages 111–117, 2000.

Masanari Kimura. Why mixup improves the model performance. In *International Conference on Artificial Neural Networks (ICANN)*, 2021.

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*, 2021.

Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, page 179. Citeseer, 1997.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *IEEE International Conference on Computer Vision*, pages 5542–5550, 2017.

Tang Li, Fengchun Qiao, Mengmeng Ma, and Xi Peng. Are data-driven explanations robust against out-of-distribution data? *arXiv preprint arXiv:2303.16390*, 2023.

Thomas Liao, Rohan Taori, Inioluwa Deborah Raji, and Ludwig Schmidt. Are we learning yet? a meta review of evaluation failures across machine learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, 2018.

Zixuan Liu, Ziqiao Wang, Hongyu Guo, and Yongyi Mao. Over-training with mixup may hurt generalization. *arXiv preprint arXiv:2303.01475*, 2023.

Wang Lu, Jindong Wang, Yiqiang Chen, Sinno Jialin Pan, Chunyu Hu, and Xin Qin. Semantic-discriminative mixup for generalizable sensor-based cross-domain activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2022a.

Wang Lu, Jindong Wang, Han Yu, Lei Huang, Xiang Zhang, Yiqiang Chen, and Xing Xie. Fixed: Frustratingly easy domain generalization with mixup. *arXiv preprint arXiv:2211.05228*, 2022b.

Linghui Meng, Jin Xu, Xu Tan, Jindong Wang, Tao Qin, and Bo Xu. Mixspeech: Data augmentation for low-resource automatic speech recognition. In *ICASSP 2021-2021*

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7008–7012. IEEE, 2021.

Ragja Palakkadavath, Thanh Nguyen-Tang, Sunil Gupta, and Svetha Venkatesh. Improving domain generalization with interpolation robustness. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022.

Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *Proc. IEEE/CVF Conf. Comp. Vis. Patt. Recogn.*, 2022.

Toby Perrett, Saptarshi Sinha, Tilo Burghardt, Majid Mirmehdi, and Dima Damen. Use your head: Improving long-tail video recognition. In *Proc. IEEE/CVF Conf. Comp. Vis. Patt. Recogn.*, 2023.

Francesco Pinto, Harry Yang, Ser-Nam Lim, Philip HS Torr, and Puneet K Dokania. Regmixup: Mixup as a regularizer can surprisingly improve accuracy and out distribution robustness. *arXiv preprint arXiv:2206.14502*, 2022.

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.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 2000.

Damien Teney, Ehsan Abbasnejad, Kushal Kafle, Robik Shrestha, Christopher Kanan, and Anton Van Den Hengel. On the value of out-of-distribution testing: An example of goodhart's law. *Advances in Neural Information Processing Systems*, 33:407–417, 2020.

Damien Teney, Yong Lin, Seong Joon Oh, and Ehsan Abbasnejad. ID and OOD performance are sometimes inversely correlated on real-world datasets. *Proc. Advances in Neural Inf. Process. Syst.*, 2023.

Huan Tian, Bo Liu, Tianqing Zhu, Wanlei Zhou, and S Yu Philip. Cifair: Constructing continuous domains of invariant features for image fair classifications. *Knowledge-Based Systems*, 2023.

Shashanka Venkataramanan, Ewa Kijak, Laurent Amsaleg, and Yannis Avrithis. Alignmixup: Improving representations by interpolating aligned features. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19174–19183, 2022.

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*, 2019.

Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *AAAI Conference on Artificial Intelligence*, 2020.

Huaxiu Yao, Caroline Choi, Bochuan Cao, Yoonho Lee, Pang Wei W Koh, and Chelsea Finn. Wild-time: A benchmark of in-the-wild distribution shift over time. *Proc. Advances in Neural Inf. Process. Syst.*, 2022a.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In *International Conference on Machine Learning*, 2022b.

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 *IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019a.

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, 2019b.

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

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.

Difan Zou, Yuan Cao, Yuanzhi Li, and Quanquan Gu. The benefits of mixup for feature learning. *arXiv preprint arXiv:2303.08433*, 2023.

Appendices

A. Reviewers' FAQ

We include questions from reviewers of this paper with our responses, which could also be informative to the readers.

No new method, not enough novelty.

The paper is not about a novel method but about improving our scientific understanding of a family of methods (selective mixup) that are highly successful (cf. WILDS and Wild-Time leaderboards). We find that the success claimed in prior work is due in part to a resampling effect, which had been completely missed in prior explanations.

We also find that prior work systematically omitted important ablations that would have revealed this effect.

Deep understanding of why/how things work is the whole point of science. Our findings yield the discovery of an unknown failure mode (Section 4.7) and new combinations of mixing/resampling that sometimes give better performance (violet bar charts).

Not enough evidence to claim that resampling helps.

The evidence:

- The **theoretical part** (Section 3) demonstrates from first principles that resampling does happen. The theory predicts that its effect would be beneficial if there is a "regression toward the mean" in the data.
- The **empirical evidence** overwhelmingly supports that resampling has indeed an effect independent from mixup. E.g. on waterbirds, the best "selective mixup" method gets the same improvements as the equivalent resampling without mixup. Other datasets (e.g. yearbook) show a combined effect from mixup + resampling.
- **Even stronger empirical evidence** comes from the correlation (Fig. 4) between the improvements in performance, and reductions in divergence between training and test distributions induced by the resampling. This is evidence that the resampling does contribute to the improvements in performance even when mixup also helps (see text in bold in Sections 4.2, 4.3, and 4.5).
- We find **support for the theoretical prediction** (that improvements would happen if there is a regression toward the mean) by checking that every dataset investigated (on which the resampling is beneficial) does indeed contain such a regression toward the mean (Section 4.6).
- **Even stronger evidence for this prediction** with a counterfactual experiment. We swap the training/test sets (Section 4.7) such that the "regression toward the mean" is reversed. As predicted, selective mixup is now detrimental. This failure mode was completely unsuspected and is only explained through resampling.

Not clear which method is best across all datasets.

There is no single best method: different datasets benefit differently from mixup and/or resampling, Unlike the simplistic story suggested in prior work.

Results on each dataset deserve an individual analysis: in some cases, selective mixup brings improvements explainable entirely by resampling effects. In other cases, the mixing also plays a role. The fact that the mixing helps *sometimes* is exactly what is stated in the title of the paper.

The mixing is still sometimes useful, right?

Yes indeed. Excerpts from the paper:

- yearbook dataset: "*(...) confirms the complementarity of the effects of resampling and within-class selective mixup*".
- arxiv dataset: "*the performance of selective mixup is explained by cumulative effects of vanilla mixup and resampling effects*".
- civilComments dataset: "*the performance of selective mixup is the result of the independent effects of vanilla mixup and resampling*".

So the mixing is clearly sometimes useful. The effects of mixup and resampling are additive and largely independent. Different datasets benefit differently from one or the other.

Is the mixing sometimes harmful?

Yes, as seen in Figure 2 for example: vanilla mixup is worse than the baseline on the *waterbirds* dataset. Prior work had already shown that vanilla mixup is not always beneficial.

What supports the causal link: resampling $\rightarrow$ better OOD performance?

Causality is supported by:

- The interventional experiments where we independently manipulate the two possible causes (mixing/resampling, cf. red/blue bar charts for every dataset).
- The conceptual understanding of the mechanism by which resampling is beneficial under distribution shifts (Section 3) and the verification that the necessary assumptions hold in the datasets (regression toward the mean, Section 4.6).

Not enough theory. The whole paper originates from one theoretical realization i.e. that selective mixup induces a resampling. Section 3 examines its implications from first principles, making theoretical predictions about the cases where it would be beneficial. Section 4 then presents an empirical investigation that essentially verifies these theoretical predictions (i.e. that the method is helpful with distribution shifts when there is "regression toward the mean").

This is pretty much a textbook description of the scientific

method (proposing a novel explanation, making new predictions theoretically, verifying the predictions empirically).

Does the resampling effect apply similarly to covariates and labels? Yes, as suggested in Section 3, the same mechanisms apply (regression towards uniformity). The analysis focuses on labels because their distributions are easier to formalize and estimate. It is less clear (to us) what would be an appropriate definition of "uniformity" in input or feature space. The extension to covariates could be an interesting line of inquiry, but seems unlikely to yield different insights.

B. Experimental details

We follow prior work on each dataset for the **architectures and hyperparameters** of our experiments. For each dataset, all methods compared use hyperparameters initially validated with the ERM baseline. All experiments use early stopping i.e. recording metrics for each run at the epoch of highest ID or worst-group validation performance (for *Wild-Time* and *waterbirds*/*civilComments* datasets respectively). Each dataset/method is run with 9 different seeds unless otherwise noted. The bar charts report the average over these seeds and error bars represent $\pm$ one standard deviation.

We noticed considerable **variability in the results reported in prior work**, sometimes for datasets/methods supposedly identical (e.g. resampling baselines on *waterbirds*). Therefore we only make comparisons across results obtained within a unique code base after re-running all baselines in the same setting.

We also found some **issues in existing code** that we could not clear up with their authors despite multiple requests. This includes inconsistent preprocessing and duplicated data in the preprocessing of civilComments in Idrissi et al. (2022), "magic constants" in the implementation of selective mixup (LISA) in Yao et al. (2022b), inappropriate architectures for *MIMIC* in (Yao et al., 2022a). We fixed these issues in our codebase. Therefore we refrain from claims or direct comparisons with the absolute state of the art.

Dataset-specific notes:

- On *waterbirds*, we use ImageNet-pretrained ResNet-50 models. The results in the main paper use linear classifiers trained on frozen features. We report similar results with fine-tuned ResNet-50 models in Figure 11.
- On *CivilComments*, we use a standard pretrained BERT. To limit the computational expense for our large number of experiments, we use the BERT-tiny version (2 layers, 2 attention heads, embeddings of size 128). The results in the main paper use linear classifiers on frozen features. We report similar results with fine-tuned models in Figure 17 (using only one seed).
- On *Wild-Time Yearbook*, we train the small CNN architecture described in Yao et al. (2022a) from scratch. In the analysis of Figure 4, we measure the distance between the training and test distributions of inputs (vectorized grayscale images). To do so, we measure the distance between every pair across the two sets. For each test example, we keep the minimum distance (i.e. closest training example), then average these distances over the test set.
- On *Wild-Time arXiv*, we use random subset of 10% of the dataset. We verified on a small number of experiments that this produces very similar results to the full dataset at a fraction of the computational expense.
- On *Wild-Time MIMIC-Readmission*, the baseline transformer architecture proposed in Yao et al. (2022b) seems inappropriate. Its ID and OOD performance is surpassed by random guessing or even by constant predictions of the majority training class. The issue probably went unnoticed because the standard accuracy metric is misleading with imbalanced data (70% ID accuracy of that ERM baseline is worse than chance).
  To remedy this, we first switch to the AUROC metric. It gives equal weight to the classes and 50% is then unambiguously equivalent to random chance.
  Second, we use a much simpler architecture. We train a "bag of embeddings" where each token (diagnosis/treatment code) is assigned a learned embedding, which are summed across sequences then fed to a linear classifier.

All experiments were run on a single laptop with an Nvidia GeForce RTX 3050 Ti GPU.

C. Additional results

We show below results from the main paper while including in-domain (ID), out-of-distribution (OOD) average-domain/average-group, and OOD worst-domain/worst-group performance. The OOD metrics are always strongly correlated across methods and training epochs, but ID and OOD performance sometimes require a trade-off, as noted in Teney et al. (2023).

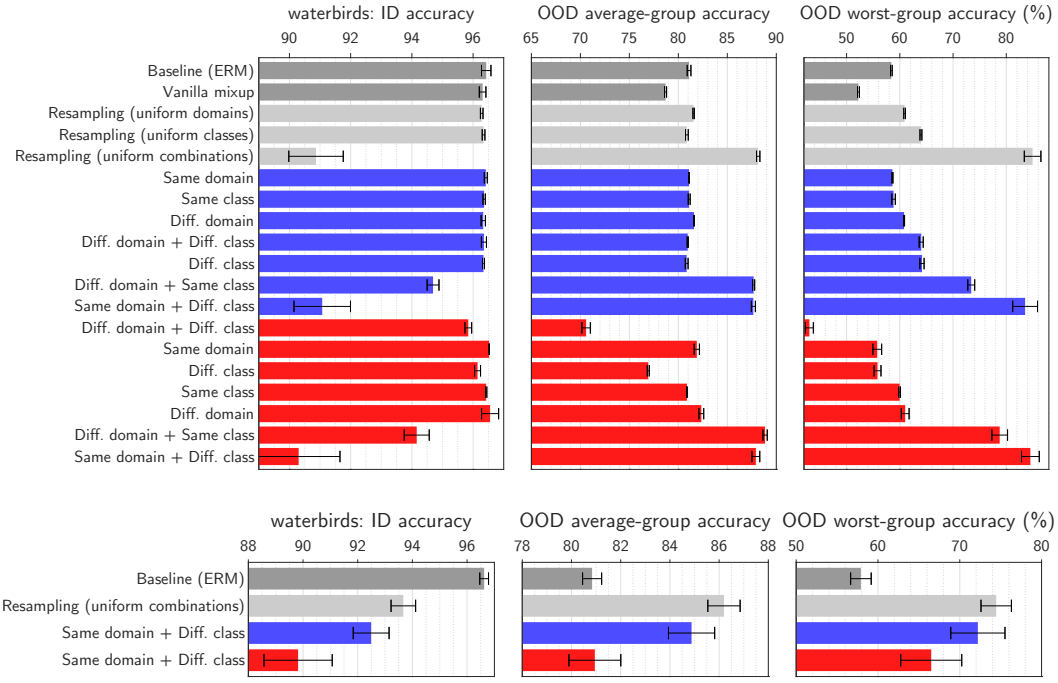

Figure 11. Results on *waterbirds* (top) with linear classifiers on frozen ResNet-50 features and (bottom) with fine-tuned ResNet-50 models (selected methods only).

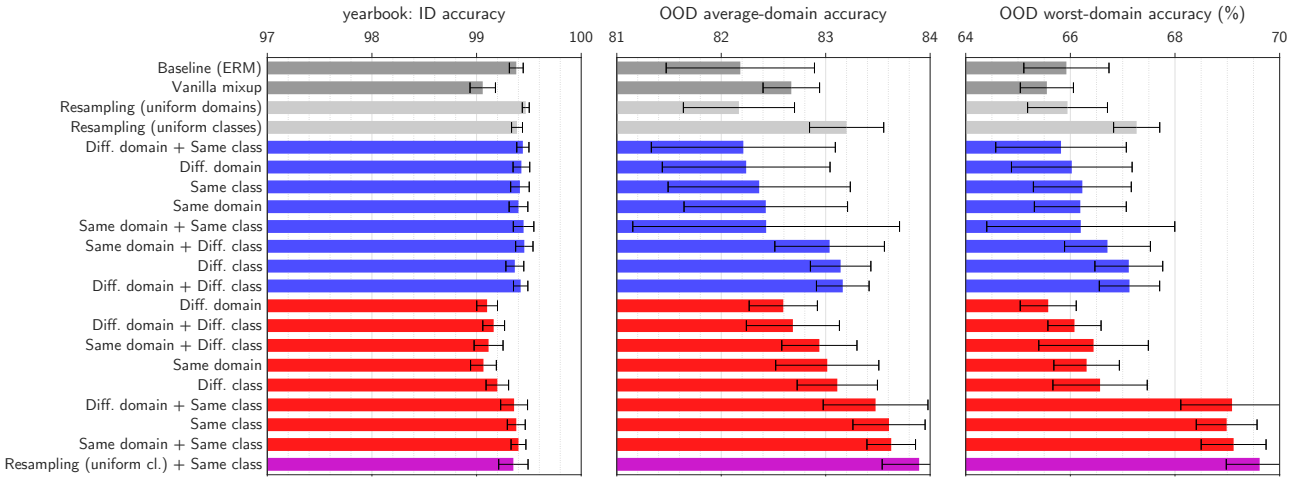

Figure 12. Results on *yearbook*.

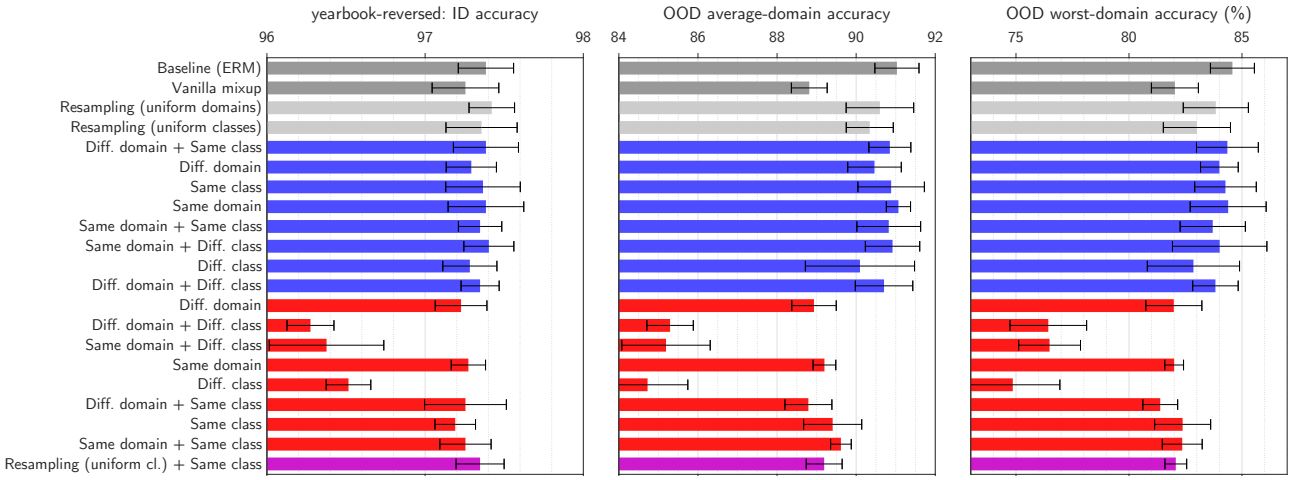

Figure 13. Results on *yearbook-reversed* (swapping ID and OOD data) to test the predicted failure mode. The "regression toward the mean" does not hold, therefore the methods that improved OOD performance on the original dataset are now detrimental (methods presented in the same order as Figure 12).

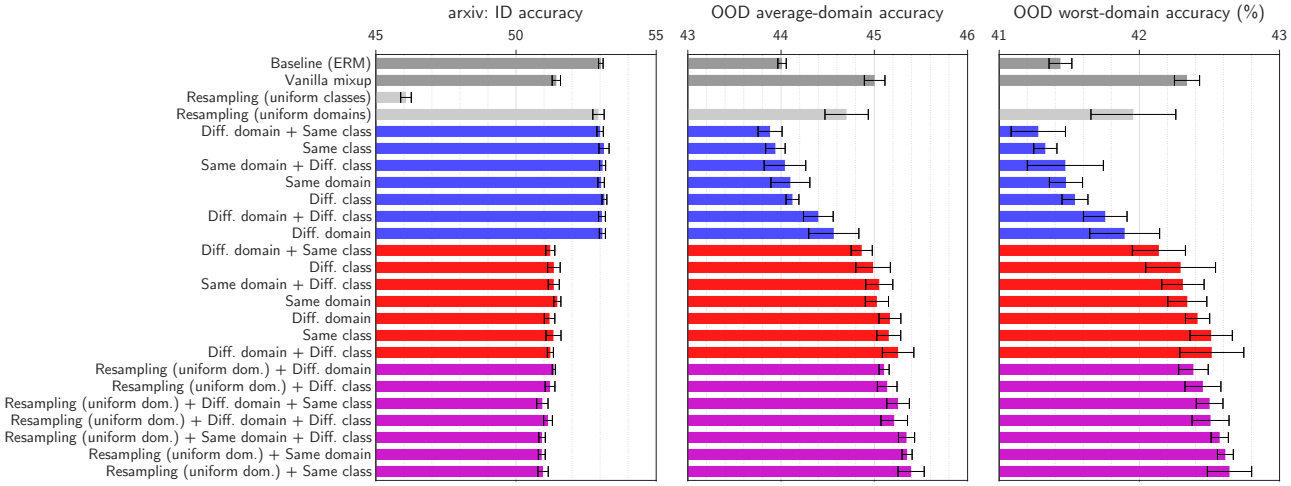

Figure 14. Results on *arXiv*. Interestingly, the methods with selective sampling without mixup are much better than selective mixup on in domain (ID) but worse out of domain (OOD). This shows a clear trade-off between these two objectives.

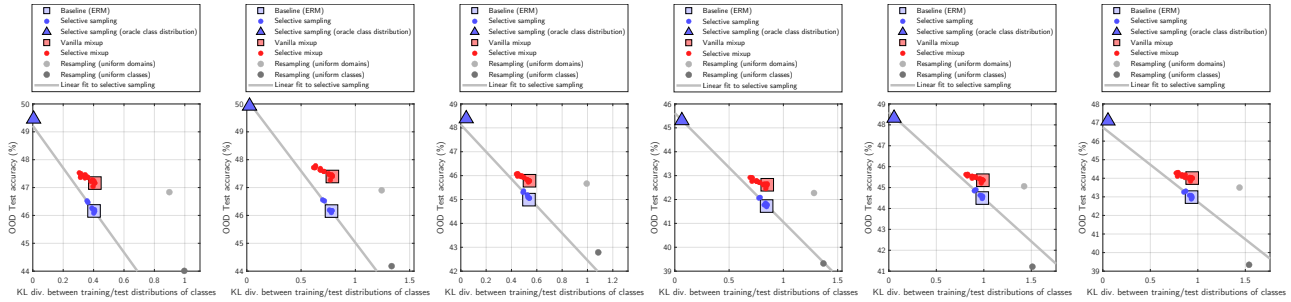

Figure 15. Same analysis as in Figure 9 of the main paper, performed on every test domain. In all cases, we observe a strong correlation between the improvements in accuracy and the reduction in divergence of the class distribution due to resampling effects.

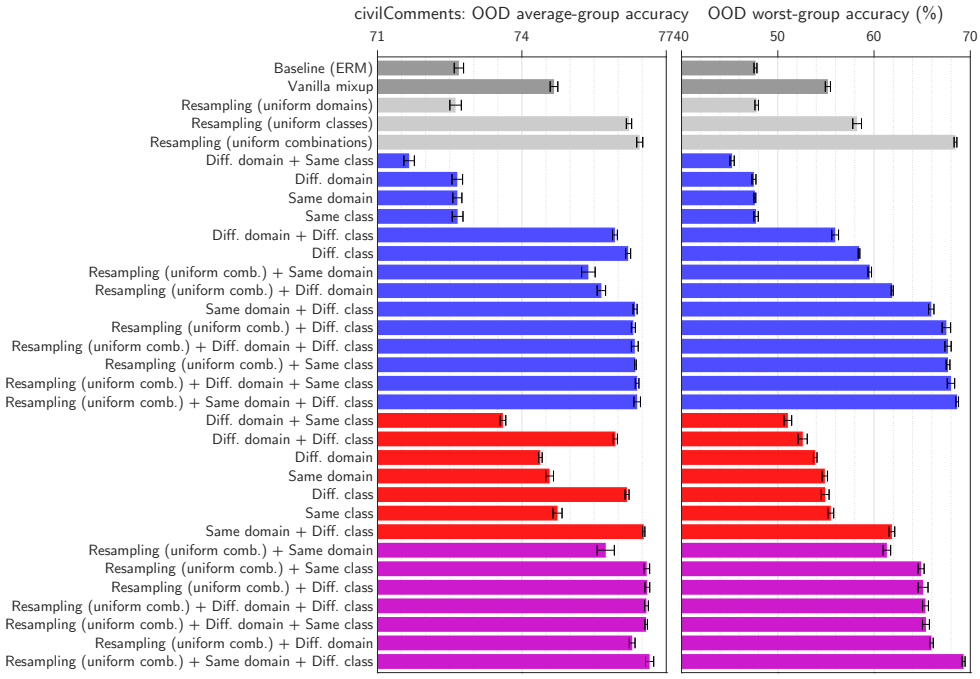

Figure 16. Results on *civilComments* with linear classifiers on frozen embeddings.

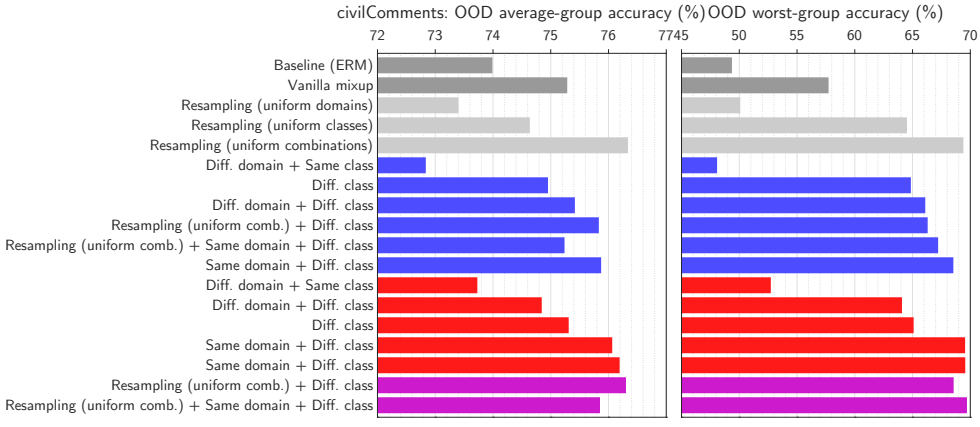

Figure 17. Results on *civilComments* with fine-tuned BERT models (single seed, reduced set of methods). These results are qualitatively identical to those with frozen embeddings above.

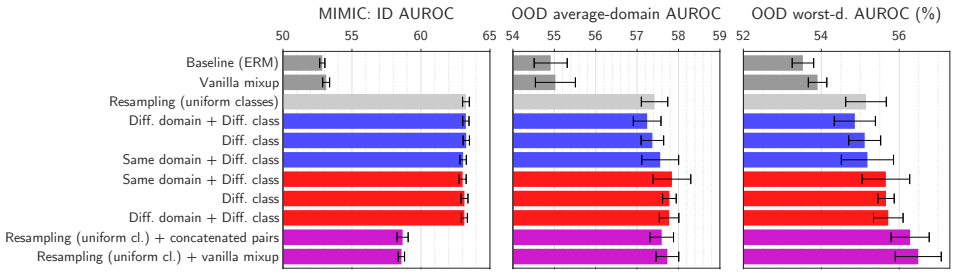

Figure 18. Results on *MIMIC-Readmission*.

D. Proof of Theorem 3.1

Theorem D.1 (Restating Theorem 3.1)**.** *Given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_i$ and paired data $\widetilde{\mathcal{D}}$ sampled according to the "different class" criterion, i.e. $\widetilde{\mathcal{D}} = \{(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{y}}_i) \sim \mathcal{D} \;\; s.t. \;\; \widetilde{\boldsymbol{y}}_i \neq \boldsymbol{y}_j\}$, then the distribution of classes in $\mathcal{D} \cup \widetilde{\mathcal{D}}$ is more uniform than in $\mathcal{D}$.*

Formally, the entropy $\mathbb{H}\big(\mathbf{p}_{\mathrm{Y}}(\mathcal{D})\big) \leq \mathbb{H}\big(\mathbf{p}_{\mathrm{Y}}(\mathcal{D} \cup \widetilde{\mathcal{D}})\big)$.

Proof. Let us define the shorthands $\boldsymbol{p} \stackrel{\text{def}}{=} \mathbf{p}_{\mathrm{Y}}(\mathcal{D})$ and $\widetilde{\boldsymbol{p}} \stackrel{\text{def}}{=} \mathbf{p}_{\mathrm{Y}}(\widetilde{\mathcal{D}})$.

In $\widetilde{\mathcal{D}}$, the ith class gets assigned, in the expectation, on a proportion of points equal to the proportion of all other classes $j \neq i$ in the original data $\mathcal{D}$.

Looking at the individual elements of $\widetilde{\boldsymbol{p}}$, we therefore have, $\forall\, i = 1 \dots C$:

$$\widetilde{p}_i \;=\; \Sigma_{j \neq i}^{C}\, p_j \;/\; (C-1) \tag{5}$$
$$\widetilde{p}_i \;=\; (1-p_i) \;/\; (C-1) \tag{6}$$

We will show that every element of $\widetilde{\boldsymbol{p}}$ is closer to $\frac{1}{C}$ than the corresponding element of $\boldsymbol{p}$:

$$|p_i \;-\; \tfrac{1}{C}| \;\geq\; |\widetilde{p}_i - \tfrac{1}{C}| \tag{7}$$
$$|\tfrac{p_i\, C - 1}{C}| \;\geq\; |\tfrac{(1-p_i)\, C - (C-1)}{C\,(C-1)}| \tag{8}$$
$$|p_i\, C \;-\; 1| \;\geq\; |\tfrac{1 - p_i\, C}{(C-1)}| \tag{9}$$
$$|p_i\, C \;-\; 1| \;\geq\; |\tfrac{p_i\, C - 1}{(C-1)}| \tag{10}$$

Therefore $\widetilde{\boldsymbol{p}}$ is closer to a uniform distribution than $\boldsymbol{p}$, and

$$\mathbb{H}(\boldsymbol{p}) \;\leq\; \mathbb{H}(\widetilde{\boldsymbol{p}}) \tag{11}$$

Since $\mathbf{p}_{\mathrm{Y}}(\mathcal{D} \cup \widetilde{\mathcal{D}}) = \big(\mathbf{p}_{\mathrm{Y}}(\mathcal{D}) \,\oplus\, \mathbf{p}_{\mathrm{Y}}(\widetilde{\mathcal{D}})\big) \,/\, 2$, we also have

$$\mathbb{H}\big(\boldsymbol{p}\big) \;\leq\; \mathbb{H}\big((\boldsymbol{p} \,\oplus\, \widetilde{\boldsymbol{p}})/2\big) \tag{12}$$
$$\mathbb{H}\big(\mathbf{p}_{\mathrm{Y}}(\mathcal{D})\big) \;\leq\; \mathbb{H}\big(\mathbf{p}_{\mathrm{Y}}(\mathcal{D} \cup \widetilde{\mathcal{D}})\big) \tag{13}$$

with an equality iff $\mathbf{p}_{\mathrm{Y}}(\mathcal{D})$ is uniform. □