Just Mix Once: Worst-group Generalization by Group Interpolation

Giorgio Giannone\textsuperscript{1}  
DTU\textsuperscript{2}  

Serhii Havrylov  
Amazon  

Jordan Massiah  
Amazon  

Emine Yilmaz  
Amazon, UCL  

Yunlong Jiao\textsuperscript{1}  
Amazon

\textbf{Abstract}

Advances in deep learning theory have revealed how average generalization relies on superficial patterns in data. The consequences are brittle models with poor performance with shift in group distribution at test time. When group annotation is available, we can use robust optimization tools to tackle the problem. However, identification and annotation are time-consuming, especially on large datasets. A recent line of work leverages self-supervision and oversampling to improve generalization on minority groups without group annotation. We propose to unify and generalize these approaches using a class-conditional variant of mixup tailored for worst-group generalization. Our approach, Just Mix Once (JM1), interpolates samples during learning, augmenting the training distribution with a continuous mixture of groups. JM1 is domain agnostic and computationally efficient, can be used with any level of group annotation, and performs on par or better than the state-of-the-art on worst-group generalization. Additionally, we provide a simple explanation of why JM1 works.

\section{Introduction}

Supervised learning aims to fit a model on a train set to maximize a global metric at test time (Vapnik, 1999; James et al., 2013; Hastie et al., 2009). However, the optimization process can exploit spurious correlations between the target $y$ and superficial patterns $c$ in the data (Geirhos et al., 2018a,b; Hermann et al., 2020; Recht et al., 2019). In this work, we are especially interested in the setting in which labels and patterns can be categorized into groups $g = (c, y)$ and we study generalization performance in the presence of shifts in group distribution at test time.

For example, in the case of \texttt{coloredMNIST} (Figure 1), a model can use information about color (spurious correlation with the target class) instead of shape to classify a digit (almost all 6s are blue at train time) during training. Then, in the presence of a shift in group distribution at test time (Hendrycks et al., 2021), the model relies on the color bias with degradation in generalization performance (for example, 6s are green and blue with equal proportion at test time). Powerful deep learning models exploit easy superficial correlations (Geirhos et al., 2018a, 2017), such as texture, color, background (Hermann et al., 2020; Hendrycks and Dietterich, 2019) to improve average generalization. Solving this issue involves constraining the model (Ilyas et al., 2019; Tsipras et al., 2018; Schmidt et al., 2018), for example leveraging explicit constraints (Zemel et al., 2013; Hardt et al., 2016; Zafar et al., 2017), invariances (Arjovsky et al., 2019; Creager et al., 2021), and causal structure (Oneto and Chiappa, 2020).

Similarly, in the presence of minority groups in a dataset, the model will tend to ignore such groups, relying on frequent patterns in majority groups. Recent work has shown how to tackle unbalanced subpopulation or groups in data using explicit supervision (Sagawa et al., 2019; Yao et al., 2022), self-supervision (Nam et al., 2020), and oversampling (Liu et al., 2021), improving generalization on minority groups. However, these methods have limitations in terms of applicability: such methods can be applied only to specific settings, like full group annotation or absence of group annotation, but fail to handle hybrid scenarios, such as partial labeling, group clustering, and changes in the train scheme.

In this work, we focus on improving worst-group generalization in realistic scenarios with any level of group annotation, devising a group annotation-agnostic method. Our approach, Just Mix Once (JM1), builds on supervised (Sagawa et al., 2019) and self-supervised approaches (Liu et al., 2021) to improve worst-group performance, generalizing and unifying such methods. JM1 leverages an augmentation scheme based on a class-conditional variant of mixup (Zhang et al., 2017; Verma et al., 2019; Carratino et al., 2020) to interpolate groups and improve worst-group generalization. JM1 is scalable, does not require undersampling or oversampling, and is flexible, easily adaptable to a variety of problem con-
figurations (full, partial, absence of group annotation) with minimal modifications.

**Contribution.** Our contributions are the following:

- We propose a simple, general mechanism, JM1, based on a class-conditional variant of mixup, to improve generalization on minority groups in the data. JM1 is group annotation agnostic, i.e., it can be employed with different levels of annotations, generalizing and unifying proposed methods to improve group generalization.

- We propose a novel interpretation of why interpolating majority and minority groups is effective in improving worst-group generalization, and justify it theoretically in an ideal case and empirically in realistic scenarios.

- We perform extensive experiments with different levels of group annotation and ablation for the mixup and class-conditional strategy. JM1 is successfully employed with I) fine-grained annotation, II) coarse annotation, and III) only a small validation set annotated, demonstrating that our method outperforms or is on par with the SOTA on vision and language datasets for worst-group generalization.

![Figure 1: Groups Identification Phase.](image1)

**Figure 1:** Groups Identification Phase. Left: train set. Right: train set partitioned. In the first phase, the training dynamics are exploited to split the data into two partitions. The assumption is that, among the samples identified as difficult (samples in the orange circle), there are typically samples from minority groups. The misclassification rate in the early stage of training is used as a clustering signal to estimate the group distribution: patterns that are superficially frequent in the data (e.g., color, texture, background) are easy to classify and have a small loss, partitioning a *majority* group; infrequent patterns (e.g., shape) are challenging to model and are misclassified during the early stage of training, partitioning a *minority* group.

![Figure 2: Class-conditional Mixing Phase.](image2)

**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. We use the label information to select two samples from different partitions, i.e., \((h^y_i, y^g_i)\) and \((h^g_j, y^y_j)\) where \(y^g_i = y^y_j = y, g \neq y\). Then we create a mixed-up sample \((h_{mix}, y_{mix})\) s.t. \(h_{mix} = \alpha h^y_i + (1 - \alpha) h^g_j, y_{mix} = y\). Note that \(h\) can either be an input (Zhang et al., 2017) or a learned representation (Verma et al., 2019). This simple mechanism gives us a principled and domain-agnostic way to augment samples marginalizing the group information in the data: by sampling \(\alpha\), we augment the training data and build a continuous distribution of groups in the data. Therefore, the model cannot rely on frequent patterns because each sample has a “slightly different” group pattern.

2 Background

Our work tackles the problem of improving worst-group generalization with different levels of group annotation available during training. Here we present two main classes of methods to deal with groups in the data with and without group annotation on the train set.

Assuming that the training data \(D = \{x^i, y^i\}_{i=1}^n\) are partitioned into (non-overlapping) groups \(G_1, \ldots, G_m\), through sub-populations represented by tuples \((y, c)\) of label \(y\) and confounding factors \(c\), we can define the following per-group and group-average loss:

\[
J(\theta; G) = \frac{1}{|G|} \sum_{(x,y) \in G} l(x,y; \theta)
\]

\[
J(\theta) = \frac{1}{m} \sum_{k=1}^m J(\theta; G_k).
\]

Notice that in case \(m = n\), each group contains a single data point, and the group-average loss collapses to the standard ERM loss \(J_{\text{ERM}}\). Note that if we define a uniform per-group loss weight \(p_k = 1/m, k = 1, \ldots, m\), we can rewrite the group-average loss as:

\[
J(\theta) = \sum_k p_k J(\theta; G_k) = \mathbb{E}_{p(G)} J(\theta; G).
\]

**Distributional Robust Optimization (DRO)** (Sagawa et al., 2019) aims to optimize:

\[
J_{\text{DRO}} = \max_k J(\theta; G_k),
\]

which corresponds to a pointwise Dirac distribution \(p_k = 1\) if \(k = \arg \max_k J(\theta; G_k)\) else 0 in Eq. 2. This approach is different from the standard training routine, where the goal is optimizing the average error among groups. If group information on the train set is absent, methods such as Just Train Twice (JTT) (Liu et al., 2021; Nam et al., 2020) can be used. JTT is a powerful approach that solves worst-group generalization in a simple two-stage approach: in the first phase, the goal is to partition the data into two clusters: one with majority groups (frequent superficial patterns); and one with minority groups (uncommon patterns).
The assumption is that samples from minority patterns are difficult to model and frequently misclassified in the early-stages of training. JTT uses the misclassified samples in the early stage of training as a signal to partition the data. Specifically, a supervised learner, parameterized by $\theta$, is trained on the data $D = \{(x_i, y_i)\}_{i=1}^n$ using ERM loss (up to constant) $J_{\text{ERM}}(\theta) := \sum_{(x,y) \in D} l(x, y; \theta)$ with early stopping. To avoid overfitting, a small validation set with group annotation is used to select the best identification epoch $T$ to create the appropriate partitions. Then misclassified samples (Nam et al., 2020) are saved in a buffer $B := \{(x_b, y_b) \text{ s.t. } \hat{f}_T(x_b) \neq y_b\}$. Once the partitions are identified, the same learner is trained for the second time on a re-weighted version of the data, where samples in $B$ are oversampled $\lambda \gg 1$ times. The intuition is that, if the examples in the error set $B$ come from challenging groups, such as those where the spurious correlation does not hold, then up-weighting them will lead to better worst-group performance (Liu et al., 2021).

Specifically, the loss for phase II) can be written as (up to constant):

$$J_{\text{JTT}}(\theta) := \lambda \sum_{(x_b, y_b) \in B} l(x_b, y_b; \theta) + \sum_{(x_b, y_b) \not\in B} l(x_b, y_b; \theta),$$

where $\lambda$ is the up-sampling rate for samples in $B$ identified from phase I). A similar approach was proposed in (Nam et al., 2020) with the difference that the groups are re-weighted after each epoch using the loss magnitude in phase one. This approach is more general, but in practice more difficult to train and scale.

**Limitations.** GroupDRO and JTT are effective methods but can be used only in specific scenarios: full-group annotation on the train set for the former, and group annotation on the validation set for the latter. These methods are challenging to adapt to hybrid scenarios where group annotation can be fine-grained for some classes (group annotation on each sample) and coarse for others (partition-based, cluster-based annotation). Such "sparse" and incomplete group annotation is the most common in practice for large, diverse datasets. To deal with such issues, we propose a simple method to improve worst-group accuracy with different levels of annotation granularity available.

### 3 Just Mix Once

We address the limitations of GroupDRO and JTT by proposing Just Mix Once (JM1). Our goal is to improve the classification accuracy of minority groups, with or without explicit annotation on the groups at training time. In the following we will use mixing and interpolation interchangeably. JM1 consists of two phases: phase I) discovers and clusters minority groups in the data, and phase II) uses class-conditional mixing to improve worst-group performance. JM1 can be used with and without group annotation on the train set.

**I) (Optional) Groups Identification.** In absence of group annotation on the train set, similarly to phase I) of JTT, we assume that we have annotated groups on a small validation set and resort to using a self-supervised signal (Nam et al., 2020; Arazo et al., 2019) based on the misclassification rate (and the loss magnitude) in the early stage of training similarly to JTT. Fig. 1 illustrates how relevant samples from minority groups are identified. Note that self-supervised training of group identification is used only in absence of group annotation on the train set: if group annotations are available on the train set, we always use such oracle groups to identify the majority and minority groups.

**II) Class-conditional Interpolation.** To generalize on minority groups at test time, a model should not rely on spurious correlations during training (e.g., texture or color) but on the signal of interest (e.g., shape). Unlike the oversampling approach taken by JTT, we resort to a better augmentation (Fig. 2) to improve generalization for low-density groups. Our mixing strategy is inspired by MixUp (Verma et al., 2019; Zhang et al., 2017; Carratino et al., 2020), and we propose a novel class-conditional variant that achieves robust worst-group generalization. Specifically, for any two samples $(h^g_i, h^\bar{g}_i)$ in the input (or representation) space from the two partitions $g, \bar{g}$ (i.e., difficult/misclassified/minority group $g$ vs the other majority group $\bar{g}$) with the same class label $y^g_i = y^\bar{g}_i = y$, we mix them up using a convex interpolation:

$$h_{\text{mix}} = \alpha h^g_i + (1 - \alpha) h^\bar{g}_i, \quad y_{\text{mix}} = y,$$

where $\alpha$ is the mixing parameter (details on how to choose $\alpha$ are deferred to Sec. 4). Notably, we empirically show that naively implementing the standard MixUp without the proposed two-stage approach fails to generalize in terms of worst-group performance (Fig. 8).

Finally, JM1 uses a simple ERM loss over mixed data:

$$J_{\text{ERM}}(\theta; G_\alpha) := \sum_{h^g, g \in G} \sum_{h^\bar{g}, \bar{g} \in G} l(h_{\text{mix}}, g).$$

#### 3.1. How JM1 Works

In this subsection we explain why JM1, based on augmenting the group distribution by interpolating samples, works. First, we discuss how our mixing strategy enables JM1 to build a "continuous" spectrum of groups in the data by sampling $\alpha$. Using MixUp with $\alpha$ mixing rate, we generate new samples drawn from a continuous mixture (Eq. 4), which has as limiting case the training distribution. In this view, we can interpret each mixed-up sample as drawn from a "mixed-up group" $G_\alpha$ parametrized by $\alpha$. This means, assuming an oracle partitioning in phase I), JM1 mixes a majority and a minority group in the data at each iteration while generating a continuous group distribution. We denote by $J(\theta; G_\alpha)$ our per-group loss for an interpolated group $G_\alpha$ with the mixing rate $\alpha$, and write the group-average loss
training distribution closer to the target in Fig. 4.

**Group Interpolation.** Here we provide mathematical intuition of why JM1 is effective. Consider that group is encoded as a 2d variable $g$ that is represented jointly by two 1d variables confounder $c \in \{0, 1\}$ and label $y \in \{0, 1\}$. Suppose that a 2d sample representation is sampled from the conditional distribution $h[c, y] \sim N((c, y), \sigma)$ where we denote by $g := (c, y)$ and set $\sigma = 1$ for simplicity. Since minority groups ($c = 0, y = 1$) and ($c = 1, y = 0$) are only present in the source domain (Fig. 3) but not in the target domain (Fig. 4), the decision boundary if naively trained on the source domain will perform very poorly in the target domain.

For a pair of samples from a minority group $g$ (under-represented, for example $g = (0, 1)$) and a majority group $\bar{g}$ (over-represented, for example $\bar{g} = (1, 1)$) respectively, now that $h \sim N(g, \sigma)$ and $\bar{h} \sim N(\bar{g}, \sigma)$, we see that the mixed sample with rate $\alpha$ follows an augmented distribution:

$$h_{\text{mix}} = \alpha h + (1 - \alpha) \bar{h}$$

where $c_{\text{mix}} \equiv (\alpha c + (1 - \alpha) \bar{c})$. Given a representation space where content (associated with label information $y$) and style (associated with the group confounder $c$) factorize, we can write $h = \mu f(c)$ and $\bar{h} = \mu f(\bar{c})$, where $\mu$ is a common content, because we are interpolating samples from different groups but with the same label $y = y_\varepsilon$, and $f$ is a mapping from group space to representation space. If $f$ respects linearity (for example the expectation operator), interpolating samples is equivalent to interpolating groups. More formally:

**Theorem 1 (Group Interpolation).**

- Let $f$ be a continuous function from group to representation space $f : G \to \mathbb{R}^d$ that respects linearity and $g = (c, y)$ a group with label $y$ and factor $c$.

- Let $(x, \bar{x}) \in \mathbb{R}^D$ be samples from the same class $y_x = y_{\bar{x}}$ but different groups, that is, $x \in g = (c, y)$ and $\bar{x} \in \bar{g} = (\bar{c}, y)$.

- Let $\mathbb{R}^d$ be a representation space such that content $\mu$ and factor $c$ can be factorized as $h = \mu(y_x) f(c)$ and $\bar{h} = \mu(y_{\bar{x}}) f(\bar{c})$ where $h$ and $\bar{h}$ are encoding for $x$ and $\bar{x}$.

\[ \Rightarrow \text{Then } h_{\text{mix}} = \mu f(c_{\text{mix}}). \]
Proof. We follow a concise constructive approach:

$$h_{\text{mix}} := \alpha h + (1 - \alpha) \bar{h}$$

factorization

$$= \alpha \mu(y_x) f(c) + (1 - \alpha) \mu(y_{\bar{x}}) f(\bar{c})$$

class-conditional

$$= \alpha \mu f(c) + (1 - \alpha) \mu f(\bar{c})$$

linearly

$$= \mu f(c_{\text{mix}}),$$

where $\mu(y_x) = \mu(y_{\bar{x}}) \equiv \mu$ because $y_x = y_{\bar{x}} \equiv y$.

Corollary 1.1. Mixing class-conditional samples from groups $g$ and $\bar{g}$ with rate $\alpha$ is equivalent to drawing samples from group $g_{\text{mix}} = (c_{\text{mix}}, y) = (\alpha c + (1 - \alpha) \bar{c}, y)$ as presented in Eq. 6.

If $f = I$, we obtain Eq. 6. We can see that, given a continuous distribution for $\alpha$, $g_{\text{mix}}$ is a sample from a continuous distribution that augments the training groups by convex combination. It is easy to see that a group sampled from the mixing distribution conditional on the same class $y$ (Fig. 6) will be closer to the target than the source domain, achieving a similar effect of the oversampling approach taken by JTT (Fig. 5). It is also clear that without the class-conditional constraint, mixing will result in an augmented source distribution whose decision boundary will be inconsistent with the target distribution. A similar idea is visualized in Fig. 11 and Fig. 12 where we assume that all samples for a group are concentrated in a small region around a normal distribution.

3.2 GroupJM1

Note that JM1 only needs access to a coarse group annotation on the train set: it only requires the oracle majority and minority group assignment for each sample, or infers it based on self-supervised group identification (Phase I).

In the presence of fine-grained group annotation where we have per-sample group labels on train (and validation set), we present a variant of JM1, called GroupJM1, to utilize all available information. This variant is motivated by GroupDRO. (Sagawa et al., 2019) proposed to approximate the non-smooth DRO loss in Eq. 3 by a smooth loss:

$$J_{\text{GroupDRO}} = \sum_{i=1}^{N} \sup_{g} \sum_{g=1}^{G} q_{g} l(x_{g}^{i}, y_{g}^{i}),$$

where $q_{g}$ is a per-group reweighting for the sample loss $l$; $q$ is learned iteratively $q_{g} \leftarrow q_{g} \exp(l(x_{g}^{i}, y_{g}^{i}))$ and exponentially increases the importance of samples with large loss, which can quickly converge to supreme loss in the limiting case.

For GroupJM1 we follow a similar procedure: for each batch we find the worst performing group based on the group loss:

$$l_{g} = 1/n_{g} \sum_{i=1}^{n_{g}} l(h_{g}^{i}, y_{g}^{i}), \quad \forall g,$$

and select the worst-group through $w_{g} = \arg \max_{g} \{l_{g}\}_{g=1}^{G}$. We then mix the samples from the worst performing group with samples from other groups conditional on the same class label, that is, $h_{\text{mix}} = \alpha h_{g} + (1 - \alpha) h_{\bar{g}}$. Finally, GroupJM1 uses the smooth approximate of the DRO loss with a minor adjustment to assign the group to each mixed sample:

$$J_{\text{GroupJM1}} = \sum_{g, \bar{g} \in G} \sum_{w_{g}, w_{\bar{g}} \in W_{g}} q_{g_{\text{mix}}} l(h_{\text{mix}}, y),$$

where $q_{\text{mix}} = q_{g}$ if $\alpha > 0.5$ or $q_{\text{mix}} = q_{\bar{g}}$ if $\alpha < 0.5$. In words, we use the class-conditional mixed data of a pair of samples from a minority (i.e. worst performing) and a majority (i.e. all other) group, but re-weight per-group loss based on the proximity of the mixed sample to the original samples.

4 Experiments

Our focus for all experiments is to improve the worst-group performance for classification problems in the vision and language domains with or without group annotation at train time and with different levels of annotation granularity. We mainly compare JM1 to GroupDRO and JTT given the similarity with such methods.

We report experiments on the benchmark proposed in (Sagawa et al., 2019). The benchmark consists of two vision and one language dataset (Fig. 7). In these datasets, the model can easily exploit superficial, majority group patterns (background in CUB, sex in CelebA, negation in MultiNLI) to solve the classification tasks at train time. When tested on a set with a different group distribution (same groups but represented in significantly different proportions), the model performance drops, showing a lack of generalization capacity.

Settings. We initially found that mixing groups by sampling $\text{Beta}(1, 1)^{4}$ uniformly, even though improving worst-group accuracy over ERM, underperforms compared to JTT. Therefore, we use a mixing strategy with a slight emphasis on the region closer to the minority groups and propose a coupled strategy: we sample $\text{Beta}(1, 1)$ for half of the epochs, and $\text{Beta}(2, 5)$ for the other half. This simple heuristic ensures that the full mixing domain is spanned,
with a focus on the region closer to the misclassified samples in phase I). We used this coupled sampling strategy for CUB and CelebA. For MultiNLI, we sample \( \alpha \) only from the uniform prior before the mixing phase runs for a very small number of epochs. More details of JM1 training are described in Appendix C.

It is worth noting that: I) like JTT, JM1 is domain-agnostic and can be applied to vision and language datasets with minimal architectural changes; II) unlike JTT’s oversampling approach, JM1 introduces an interpolation operation between samples. Note that we can apply JM1 to mix samples at different levels of abstraction: either the input (Zhang et al., 2017) or the representation space (Verma et al., 2019). As the comparison of different mixup levels is not the focus of our study, we refer the readers to Appendix B for additional ablations.

Datasets. 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, minority groups are present in 5% of the data. At test time the groups are evenly distributed.

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: the presence of a negation \( c_{neg} \) and the absence of a negation \( c_{neg} \) in the reasoning.

4.1 Worst-Group Generalization

To evaluate the generalization capacity of our approach, we consider three different scenarios in terms of group annotation.

1. Fine-grained group annotation on the train set.
2. Coarse group annotation on the train set.
3. Validation set group annotated.

Fine-grained group annotation on the train set. In the presence of multiple groups in the data, we have access to fine-grained group annotation on the train set. Consequently, group-aware methods can be employed. To test JM1 using oracle groups (i.e. without the Group Identification phase I) and its variant GroupJM1 (end Sec.3), which exploit known group annotation during training, we compare them with GroupDRO (Sagawa et al., 2019) and LISA (Yao et al., 2022). Note that LISA exploits a mixture of intra-label and intra-domain mixup with fine-grained multi-class group annotation on the train set, while JM1 exploits class-conditional mixup using a majority vs minority binary group assignment. GroupDRO finds the worst-performing group in each batch, computing the per-group loss and reweighting the training loss using ad-hoc hyperparameters to adjust for minority groups. As shown in Table 1, GroupJM1 is very competitive in this scenario. Particularly in the MultiNLI data set, JM1 is the best performing method even using a simple ERM on the mixed sample, corroborating the class-conditional mixing procedure as a general approach to improve worst-group accuracy.

Coarse group annotation on the train set. The second set of experiments compares JM1 and JTT in a different scenario: we assume that we have access to the ground-truth minority group assignment on the train set. Note that this does not equate to the fine-grained per-sample group annotation: we only need a binary mask to know which sample has a group-conflicting confounder against its label. Now we can use such an oracle majority vs. minority groups in place of the identified set from phase I). As noted in Appendix C.2 in (Liu et al., 2021), the performance of JTT drops if not using identified error sets from phase I), since JTT can excessively oversample the minority group. Given its inner working of mixing groups rather than oversampling, we expect JM1 to outperform JTT in this oracle configuration. Table 2 validates our intuition. JM1 outperforms JTT (oracle) on both datasets.

Group annotation on the validation set. The last set of experiments evaluate JM1 without group annotation on the

### Table 1: Fine-grained group annotation. Experiments using GroupDRO and GroupJM1 variants. We assume fine-grained group annotation on the train set. We find the worst performing group in each mini batch and use such information to improve JM1. We samples \( \alpha \sim U(0, 1) \). GroupDRO results from (Sagawa et al., 2019) and LISA results from (Yao et al., 2022). LISA and JM1 use a ERM loss as training objective. GroupDRO and GroupJM1 use the DRO loss.

|           | CUB | CelebA | MultiNLI |
|-----------|-----|--------|----------|
|           | avg | worst  | avg      | worst  |
| LISA      | 91.8| 89.2   | 92.4     | 89.3   |
| GroupDRO  | 93.5| 91.4   | 92.9     | 88.9   | 81.4     | 77.7 |
| JM1 (ours)| 90.9| 90.5   | 92.0     | 86.1   | 82.1     | 81.5 |
| GroupJM1  | 93.3| 91.3   | 93.0     | 90.0   | 81.9     | 79.3 |

### Table 2: Coarse group annotation. Results on CUB and CelebA datasets with coarse group annotation, where we use oracle majority and minority group annotations (instead of trained groups from Phase I). JM1 with class-conditional mixing generalizes better than oversampling-based JTT.

|           | CUB | CelebA |
|-----------|-----|--------|
|           | avg | worst  | avg | worst |
| JTT (w/o I)| 96.7| 75.9   | 93.4| 57.2  |
| JM1 (w/o I)| 95.0| 86.1   | 92.3| 85.6  |
dataset. The goal is to classify land birds vs water birds in the presence of a confounder: land background $c_l$ vs water background $c_w$. JM1 is competitive with JTT improves accuracy on both minority groups $(y_l, c_l)$ and $(y_w, c_l)$ and competitive on the other two majority groups. We select the best run among 5 in terms of worst-group accuracy.

### Table 3: Validation-set per-group accuracy

|                | CUB |     |     |     |     |     |     |
|----------------|-----|-----|-----|-----|-----|-----|-----|
| $(y_l, c_l)$   | 94.3| 86.7| 87.5| 91.6| 93.3| 86.7|
| $(y_l, c_w)$   | 88.5| 89.9| 90.5| 93.1|     |     |
| $(y_w, c_l)$   |     |     |     |     |     |     | 88.5|
| $(y_w, c_w)$   |     |     |     |     |     |     |     |
| **avg**        |     |     |     |     |     |     | 87.5|
| **worst**      |     |     |     |     |     |     | 86.7|

### Table 4: Validation-set per-group accuracy

|                | CelebA |     |     |     |     |     |     |
|----------------|--------|-----|-----|-----|-----|-----|-----|
| $(y_b, c_m)$   | 90.9   | 87.9| 82.8| 89.0| 89.1| 82.8|
| $(y_b, c_f)$   | 88.8   | 87.7| 84.4| 86.6| 87.9| 84.4|
| **avg**        |        |     |     |     |     |     |     |
| **worst**      |        |     |     |     |     |     |     |

Fig. 8 shows that, while standard MixUp fails to improve worst-group performance, the proposed mixing strategy performs significantly better than both MixUp baselines by a large margin, despite that all three methods perform very similarly in terms of average performance. In Fig. 9 we investigate the performance of JM1 with and without the class-conditional constraint. We see that empirically, conditional JM1 performs better than unconditional JM1. The worst-group accuracy decreases as the dataset complexity (in terms of the number of samples, classes, and groups) increases for both models.

### 5 Related Work

**Robust classification.** 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 (Geirhos et al., 2018a). Recently approaches to preserve group information have been proposed, leveraging robust optimization (Sagawa et al., 2019; Mohri et al., 2019; Zhang et al., 2020), data augmentation (Goel et al., 2020), debiasing approaches (Hong and Yang, 2021; Bahng et al., 2020), and group reweighting (Shimodaira, 2000).
Table 5: Validation-set group annotation. Results on CUB, CelebA and MultiNLI datasets for average accuracy and worst-group accuracy. JM1 performs comparably or better than the SOTA in worst-group performance without group annotation on the train-set, closing the gap with GroupDRO that requires full group annotations on the train set. 1st / 2nd best worst-group accuracies bolded / underlined. Our method is better or comparable with sota with the advantage of being computationally inexpensive, used with different level of annotation, different data modalities, without introducing regularization or additional complexity.

| Method                | CUB       | CelebA    | MultiNLI  |
|-----------------------|-----------|-----------|-----------|
|                        | avg acc   | worst acc | avg acc   | worst acc | avg acc   | worst acc |
| ERM (Liu et al., 2021)| 97.6      | 72.6      | 95.6      | 47.2      | 82.4      | 67.9      | ❌         |
| CVaR-DRO (Levy et al., 2020)| 96.5    | 69.5      | 82.4      | 64.4      | 82.0      | 68.0      | ❌         |
| LfF (Nam et al., 2020)| 97.3      | 75.2      | 86.0      | 70.6      | 80.8      | 70.2      | ❌         |
| EIL (Creager et al., 2021)| 96.9    | 78.6      | 85.7      | 81.7      | -        | -         | ❌         |
| JTT (Liu et al., 2021)| 93.3      | 86.7      | 89.1      | 82.8      | 78.6      | 72.6      | ❌         |
| CNC (Zhang and Ré, 2022)| 90.9    | **88.4**  | 89.9      | **88.8**  | -        | -         | ❌         |
| JM1 (ours)            | 93.1      | **88.5**  | 87.9      | 84.4      | 80.3      | **72.5**  | ✓          |
| GroupDRO (Sagawa et al., 2019)| 93.5   | 91.4      | 92.9      | 88.9      | 81.4      | 77.7      | ✓          |

Just Mix Once: Worst-group Generalization by Group Interpolation

Groups generalization without annotation. All the aforementioned methods assume that fine-grained group annotation is available at train time. However, assuming subpopulation annotation in large diverse datasets is problematic because the process is time consuming and prone to errors. Learning from Failure (Nam et al., 2020) proposes a group annotation-free and domain agnostic procedure to improve worst-group performance. The authors 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 and the procedure applied multiple times during training. Similar ideas are explored in (Yaghoobzadeh et al., 2019; Utama et al., 2020).

2000; Byrd and Lipton, 2019; Sagawa et al., 2020) using importance samples to rebalance minority and majority groups. If full group annotation is provided, robust optimization methods are employed. The supervised scenario is closely related to the fairness one (Hardt et al., 2016; Agarwal et al., 2018; Oneto and Chiappa, 2020; Pleiss et al., 2017; Woodworth et al., 2017): wherein distribution shift we want to improve the worst-group generalization, in fairness we want to learn features and distributions that are invariant to different groups, using explicit constraints (Zafar et al., 2017, 2019; Zemel et al., 2013), invariant learning (Arjovsky et al., 2019), auxiliary tasks (Edwards and Storkey, 2015; Lahoti et al., 2020), and causal structure (Oneto and Chiappa, 2020). In the causality literature (Pearl, 2009) the group information is interpreted as a confounder. If we have access to the confounder (group annotation), we can control for it. Recently risk extrapolation (Krueger et al., 2021) has been proposed to deal with domain shift in the presence of confounders. Similarly, JM1 augments the training data with plausible scenarios not present in the data. A similar approach has been proposed in (Chuang and Mroueh, 2021) in the fairness domain and in (Yao et al., 2022) for group robustness with full group annotation.

Just Train Twice (Liu et al., 2021) proposes a similar solution, in which the identification phase is performed one time and then a reweighting phase is trained. (Sohoni et al., 2020) uses clustering to improve the classification of underrepresented classes. Distributional robust optimization (Ben-Tal et al., 2013; Duchi et al., 2019, 2021), where the optimization problem is defined over a set of distributions around the empirical distribution, has also been widely used to improve group robustness. Invariant Learning (Arjovsky et al., 2019; Creager et al., 2021) is another approach to improve worst-group performance. The goal is to obtain a learner that performs well in different environments, where in each environment we have the same classes but different founders. Concurrent to our work, SelecMix (Hwang et al., 2022) proposes to identify groups using contrastive learning and then use mixup similarly to the second phase of JM1. However, the method is applied only to image data, in one specific setting (validation set group annotation), and relies on distance between samples, where JM1 does not make assumptions on the data modality or the specific setting.

6 Conclusion

We present JM1, a simple and effective method to improve worst-group performance when different level of annotation are available. We show that our approach is competitive with sota and robust to the choice of identification phase. Class-conditional group interpolation provides a cheap, domain-agnostic way to augment the training distribution and improve worst-group accuracy. However, our approach does not guarantee fairness. We cannot use JM1 when there is perfect alignment between founders and labels. We consider the data-driven problem without accounting for the collection, labeling, and source of the data. The results are sensitive to the sampling distribution for α. More robust interpolations mechanisms will be explored in future work.
References

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. (2018). A reductions approach to fair classification. In International Conference on Machine Learning, pages 60–69. PMLR.

Arazo, E., Ortego, D., Albert, P., O’Connor, N., and McGuinness, K. (2019). Unsupervised label noise modeling and loss correction. In International Conference on Machine Learning, pages 312–321. PMLR.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. (2019). Invariant risk minimization. arXiv preprint arXiv:1907.02893.

Bahng, H., Chun, S., Yun, S., Choo, J., and Oh, S. J. (2020). Learning de-biased representations with biased representations. In International Conference on Machine Learning, pages 528–539. PMLR.

Ben-Tal, A., Den Hertog, D., De Waegenaere, A., Melenberg, B., and Remmert, G. (2013). Robust solutions of optimization problems affected by uncertain probabilities. Management Science, 59(2):341–357.

Byrd, J. and Lipton, Z. (2019). What is the effect of importance weighting in deep learning? In International Conference on Machine Learning, pages 872–881. PMLR.

Carratino, L., Cissé, M., Jenatton, R., and Vert, J.-P. (2020). On mixup regularization. arXiv preprint arXiv:2006.06049.

Chuang, C.-Y. and Mroueh, Y. (2021). Fair mixup: Fairness via interpolation. arXiv preprint arXiv:2103.06503.

Creager, E., Jacobsen, J.-H., and Zemel, R. (2021). Environment inference for invariant learning. In International Conference on Machine Learning, pages 2189–2200. PMLR.

Duchi, J. C., Glynn, P. W., and Namkoong, H. (2021). Statistics of robust optimization: A generalized empirical likelihood approach. Mathematics of Operations Research.

Duchi, J. C., Hashimoto, T., and Namkoong, H. (2019). Distributionally robust losses against mixture covariate shifts. Under review.

Edwards, H. and Storkey, A. (2015). Censoring representations with an adversary. arXiv preprint arXiv:1511.05897.

Geirhos, R., Janssen, D. H., Schütt, H. H., Rauber, J., Bethge, M., and Wichmann, F. A. (2017). Comparing deep neural networks against humans: object recognition when the signal gets weaker. arXiv preprint arXiv:1706.06969.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. (2018a). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231.

Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. (2018b). Generalisation in humans and deep neural networks. Advances in neural information processing systems, 31.

Goel, K., Gu, A., Li, Y., and Ré, C. (2020). Model patching: Closing the subgroup performance gap with data augmentation. arXiv preprint arXiv:2008.06775.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. Advances in neural information processing systems, 29:3315–3323.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer.

Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261.

Hermann, K., Chen, T., and Kornblith, S. (2020). The origins and prevalence of texture bias in convolutional neural networks. Advances in Neural Information Processing Systems, 33:19000–19015.

Hong, Y. and Yang, E. (2021). Unbiased classification through bias-contrastive and bias-balanced learning. Advances in Neural Information Processing Systems, 34.

Hwang, I., Lee, S., Kwak, Y., Oh, S. J., Teney, D., Kim, J.-H., and Zhang, B.-T. (2022). Selectmix: Debiasled learning by mixing up contradicting pairs.

Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. Advances in neural information processing systems, 32.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning, volume 112. Springer.

Krueger, D., Caballero, E., Jacobsen, J.-H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., and Courville, A. (2021). Out-of-distribution generalization via risk extrapolation (rex). In International Conference on Machine Learning, pages 5815–5826. PMLR.

Lahoti, P., Beutel, A., Chen, J., Lee, K., Prost, F., Thain, N., Wang, X., and Chi, E. H. (2020). Fairness without demographics through adversarially reweighted learning. arXiv preprint arXiv:2006.13114.

Levy, D., Carmon, Y., Duchi, J. C., and Sidford, A. (2020). Large-scale methods for distributionally robust optimization. arXiv preprint arXiv:2010.05893.
Liu, E. Z., Haghgoo, B., Chen, A. S., Raghunathan, A., Koh, P. W., Sagawa, S., Liang, P., and Finn, C. (2021). Just train twice: Improving group robustness without training group information. In International Conference on Machine Learning, pages 6781–6792. PMLR.

Mohri, M., Sivek, G., and Suresh, A. T. (2019). Agnostic federated learning. In International Conference on Machine Learning, pages 4615–4625. PMLR.

Nam, J., Cha, H., Ahn, S., Lee, J., and Shin, J. (2020). Learning from failure: Training debiased classifier from biased classifier. arXiv preprint arXiv:2007.02561.

Oneto, L. and Chiappa, S. (2020). Fairness in machine learning. Recent Trends in Learning From Data, pages 155–196.

Pearl, J. (2009). Causality. Cambridge university press.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., and Weinberger, K. Q. (2017). On fairness and calibration. arXiv preprint arXiv:1709.02012.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2019). Do imagenet classifiers generalize to imagenet? In International Conference on Machine Learning, pages 5389–5400. PMLR.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2019). Distributionally robust neural networks. In International Conference on Learning Representations.

Sagawa, S., Raghunathan, A., Koh, P. W., and Liang, P. (2020). An investigation of why overparameterization exacerbates spurious correlations. In International Conference on Machine Learning, pages 8346–8356. PMLR.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. (2018). Adversarially robust generalization requires more data. Advances in neural information processing systems, 31.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of statistical planning and inference, 90(2):227–244.

Soheri, N. S., Dunnmon, J. A., Angus, G., Gu, A., and Ré, C. (2020). No subclass left behind: Fine-grained robustness in coarse-grained classification problems. arXiv preprint arXiv:2011.12945.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2018). Robustness may be at odds with accuracy. arXiv preprint arXiv:1805.12152.

Utama, P. A., Moosavi, N. S., and Gurevych, I. (2020). Towards debiasing nlu models from unknown biases. arXiv preprint arXiv:2009.12303.

Vapnik, V. (1999). The nature of statistical learning theory. Springer science & business media.

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. (2019). Manifold mixup: Better representations by interpolating hidden states. In International Conference on Machine Learning, pages 6438–6447. PMLR.

Woodworth, B., Gunasekar, S., Ohannessian, M. I., and Srebro, N. (2017). Learning non-discriminatory predictors. In Conference on Learning Theory, pages 1920–1953. PMLR.

Yaghoobzadeh, Y., Mehr, S., Tachet, R., Hazen, T. J., and Sordoni, A. (2019). Increasing robustness to spurious correlations using forgettable examples. arXiv preprint arXiv:1911.03861.

Yao, H., Wang, Y., Li, S., Zhang, L., Liang, W., Zou, J., and Finn, C. (2022). Improving out-of-distribution robustness via selective augmentation. arXiv preprint arXiv:2201.00299.

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., and Gummadi, K. P. (2019). Fairness constraints: A flexible approach for fair classification. The Journal of Machine Learning Research, 20(1):2737–2778.

Zafar, M. B., Valera, I., Rogriguez, M. G., and Gummadi, K. P. (2017). Fairness constraints: Mechanisms for fair classification. In Artificial Intelligence and Statistics, pages 962–970. PMLR.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dworak, C. (2013). Learning fair representations. In International conference on machine learning, pages 325–333. PMLR.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.

Zhang, J., Menon, A., Veit, A., Bhojanapalli, S., Kumar, S., and Sra, S. (2020). Coping with label shift via distributionally robust optimisation. arXiv preprint arXiv:2010.12230.

Zhang, M. and Ré, C. (2022). Contrastive adapters for foundation model group robustness. arXiv preprint arXiv:2207.07180.
A Preliminary Results without Group Annotation

While JTT and JM1 do not use full group annotation on the train set, they both rely on group annotation on a small validation set to guide group identification (i.e., choosing early stopping epoch). A natural question to investigate is if we can remove the need for group annotation completely, and JM1 can still perform reasonably well. To that end, we test the robustness of JM1 to different identification epochs/sets in phase I) using the CUB dataset. Specifically, we checkpoint three different identification scenarios (20, 40, 60 epochs) in phase I) for both JM1 and JTT. Then we select for JM1 the best phase II) model in terms of average accuracy, while we tune the best-performing JTT using group annotation on a validation set in terms of worst-group accuracy. Note that JM1 is now completely free from group annotations. Table 6 shows that JM1 without group annotation at all loses only around 5% compared to the full JTT with access to group annotation on a validation set. These are preliminary results that indicate the potential of training group-robust models in a fully self-supervised manner. We can hope to achieve worst-group robust generalization with no more data requirement than what we need to train typical ML models.

![Figure 10](image)
Table 7: Ablation 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 a variant of JM1 with additional oversampling. U: uniform distribution $U(0, 1)$. B: beta distribution $B(2, 5)$.

| layer | $\alpha$ | $(y_l, c_l)$ | $(y_l, c_w)$ | $(y_w, c_l)$ | $(y_w, c_w)$ | worst acc |
|-------|-----------|--------------|--------------|--------------|--------------|-----------|
| JTT   | -         | 94.3         | 86.7         | 87.5         | 91.6         | 86.7      |
| JM1   | input     | $U$          | 98.4         | 84.2         | 77.7         | 93.0      | 77.7      |
| JM1   | input     | $U(0.5)$    | 96.7         | 84.8         | 86.8         | 93.5      | 84.8      |
| JM1   | early     | $U$          | 97.5         | 84.7         | 80.1         | 92.2      | 80.1      |
| JM1   | output    | $U$          | 97.3         | 85.9         | 84.7         | 92.7      | 84.7      |
| JM1   | random    | $U$          | 93.8         | 87.8         | 89.7         | 90.8      | 87.8      |
| JM1   | random    | $U/B$        | 94.2         | 88.5         | 88.9         | 90.5      | 88.5      |
| JM1+  | input     | $U$          | 91.9         | 89.4         | 90.5         | 84.9      | 84.9      |
| JM1+  | input     | $U/B$        | 96.3         | 87.6         | 88.8         | 93.3      | 87.6      |
| JM1+  | early     | $U$          | 92.5         | 87.9         | 87.5         | 89.9      | 87.5      |
| JM1+  | random    | $U/B$        | 94.0         | 89.6         | 90.7         | 90.2      | 89.6      |
Table 8: Ablation on CUB dataset for GroupDRO and JM1 variants with different $\alpha$ schedule for the mixing procedure. U: uniform distribution $U(0, 1)$. B: beta distribution $B(2, 5)$.

|                | CUB avg | CUB worst | $\mathcal{L}$ | $\alpha$ mode | mode |
|----------------|---------|----------|---------------|--------------|------|
| GroupDRO       | 96.6    | 84.6     | DRO           | base         |      |
| GroupDRO       | 93.5    | 81.4     | DRO           | base         |      |
| GroupJM1       | 90.3    | 89.3     | DRO           | $B$          | adj  |
| GroupJM1       | 90.5    | 89.9     | DRO           | $U/B$        | adj  |
| GroupJM1       | 92.2    | 91.3     | DRO           | $U$          | adj  |
| JM1            | 94.2    | 86.6     | ERM           | $U/B$        |      |
| JM1            | 94.4    | 86.7     | ERM           | $B$          |      |
| JM1            | 90.9    | 90.5     | ERM           | $U$          |      |

Figure 13: Worst-group and average accuracy for different setups. On the x-axis we have different epochs we evaluate the model performance in the II phase. Each line represents a different epoch to identify samples in the I phase.

Table 9: Results on CUB for different identification epochs. For JM1-avg we choose the model based on average accuracy.

| Identification Epoch | JM1-avg avg | JM1-avg worst | JM1-robust avg | JM1-robust worst |
|----------------------|-------------|---------------|----------------|-----------------|
| 10id                 | 86.6        | 39.7          | 80.6           | 64.2            |
| 20id                 | 92.2        | 72.7          | 89.8           | 84.4            |
| 30id                 | 93.6        | 81.8          | 90.8           | 86.4            |
| 40id                 | 93.7        | 81.8          | 90.3           | 87.5            |
| 50id                 | 94.1        | 82.6          | 90.8           | 87.5            |
| 60id                 | 93.3        | 82.7          | 91.1           | 87.0            |
| 70id                 | 93.3        | 79.1          | 90.2           | 85.8            |
| 80id                 | 92.8        | 79.3          | 90.2           | 85.0            |
| 90id                 | 93.0        | 78.0          | 89.9           | 84.7            |
| 100id                | 91.2        | 69.0          | 86.3           | 81.3            |

Table 10: Per-group accuracy on MultiNLI. The goal is to classify pair of sentences as entailment $y_e$, contradiction $y_c$ and neutral $y_n$ in the presence of a confounder: negation $c_{neg}$ and no negation $c_{nneg}$.

|                  | $(y_e, c_{nneg})$ | $(y_c, c_{neg})$ | $(y_e, c_{nneg})$ | $(y_c, c_{neg})$ | $(y_n, c_{nneg})$ | avg acc | worst acc |
|------------------|------------------|------------------|------------------|------------------|------------------|---------|-----------|
| JTT              | 80.3             | 93.1             | 81.0             | 73.6             | 78.4             | 70.6    | 80.5      |
| JM1              | 79.3             | 91.1             | 81.6             | 73.6             | 78.5             | 73.5    | 80.3      |
C Experimental Details

Table 11: Training details for JM1. CE: cross-entropy. U: uniform distribution $U(0, 1)$. B: beta distribution $Beta(2, 5)$. w/o I: without using phase I for cluster identification. For all the other hyper-parameters we use the one proposed in (Liu et al., 2021).

|                        | CUB | CelebA | MultiNLI | CUB (w/o I) | CelebA (w/o I) |
|------------------------|-----|--------|----------|-------------|----------------|
| 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             |
| $\alpha$               | 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            |