Not Eliminate but Aggregate: Post-Hoc Control over Mixture-of-Experts to Address Shortcut Shifts in Natural Language Understanding

Ukyo Honda[1] **Tatsushi Oka**[2] **Peinan Zhang**[1] **Masato Mita**[1]
[1]CyberAgent, Japan [2]Keio University, Japan
{honda_ukyo,zhang_peinan,mita_masato}@cyberagent.co.jp
tatsushi.oka@keio.jp

Abstract

Recent models for natural language understanding are inclined to exploit simple patterns in datasets, commonly known as *shortcuts*. These shortcuts hinge on spurious correlations between labels and latent features existing in the training data. At inference time, shortcut-dependent models are likely to generate erroneous predictions under distribution shifts, particularly when some latent features are no longer correlated with the labels. To avoid this, previous studies have trained models to eliminate the reliance on shortcuts. In this study, we explore a different direction: pessimistically aggregating the predictions of a mixture-of-experts, assuming each expert captures relatively different latent features. The experimental results demonstrate that our post-hoc control over the experts significantly enhances the model's robustness to the distribution shift in shortcuts. Additionally, we show that our approach has some practical advantages. We also analyze our model and provide results to support the assumption.[1]

1 Introduction

The datasets for natural language understanding (NLU) often contain simple patterns correlated with target labels, which are unintentionally introduced by annotators' simple heuristics, preferences, etc. (Gururangan et al., 2018; Geva et al., 2019). More fundamentally, the compositional nature of natural language inherently introduces tokens that correlate with target labels individually (Gardner et al., 2021). For example, word overlap (McCoy et al., 2019; Zhang et al., 2019) and specific vocabulary, such as negations (Gururangan et al., 2018; Schuster et al., 2019), are known to have such correlations. However, these correlations are not guaranteed to hold in general and are therefore called **spurious correlations** (Feder et al., 2022). The simple patterns are easy to exploit, so recent NLU models are inclined to take advantage of them. This exploitation or the exploited patterns themselves are called **shortcuts** (Makar et al., 2022; Feder et al., 2022; Du et al., 2021; Meissner et al., 2022).[2]

At inference time, shortcuts often result in inaccurate predictions under relevant distribution shifts. The shifts can occur, for example, when test data are collected from annotators with different heuristics or preferences (Geva et al., 2019; McCoy et al., 2019; Zhang et al., 2019; Schuster et al., 2019). Data from the same distribution as the training data is referred to as **in-distribution (ID)** data, while data from a distribution shifted relative to the training data is referred to as **out-of-distribution (OOD)** data. Figure 1 shows the examples of ID and OOD data.

A simple solution to this problem is to eliminate reliance on shortcuts, which is the mainstream approach, including recent studies in NLU (Clark et al., 2019; He et al., 2019; Mahabadi et al., 2020). Typically, those methods up-weight training instances where some known shortcuts cannot predict labels correctly and down-weight the others. A practical deficiency of this approach arises in a performance trade-off between ID and OOD data. It deviates models from ID data by eliminating shortcuts, which are valid features in ID data. Due to this trade-off, another practical problem arises where the hyperparameter search has to be made using OOD test or validation data, as noticed as the limitation of previous work (Clark et al., 2019; Mahabadi et al., 2020; Clark et al., 2020a; Ghaddar et al., 2021; Liu et al., 2021; Creager et al., 2021; Yu et al., 2022; Yang et al., 2023).

[1]The code is available at https://github.com/CyberAgentAILab/posthoc-control-moe.

[2]Shortcuts are also called dataset *bias*. However, we avoid using this term because it is confusing with the social bias or bias of an estimator. Similarly, we do not use the term *debiasing* in this paper.

Transactions of the Association for Computational Linguistics, vol. 12, pp. 1268–1289, 2024. https://doi.org/10.1162/tacl_a_00701
Action Editor: Jacob Eisenstein. Submission batch: 2/2024; Revision batch: 5/2024; Published 10/2024.

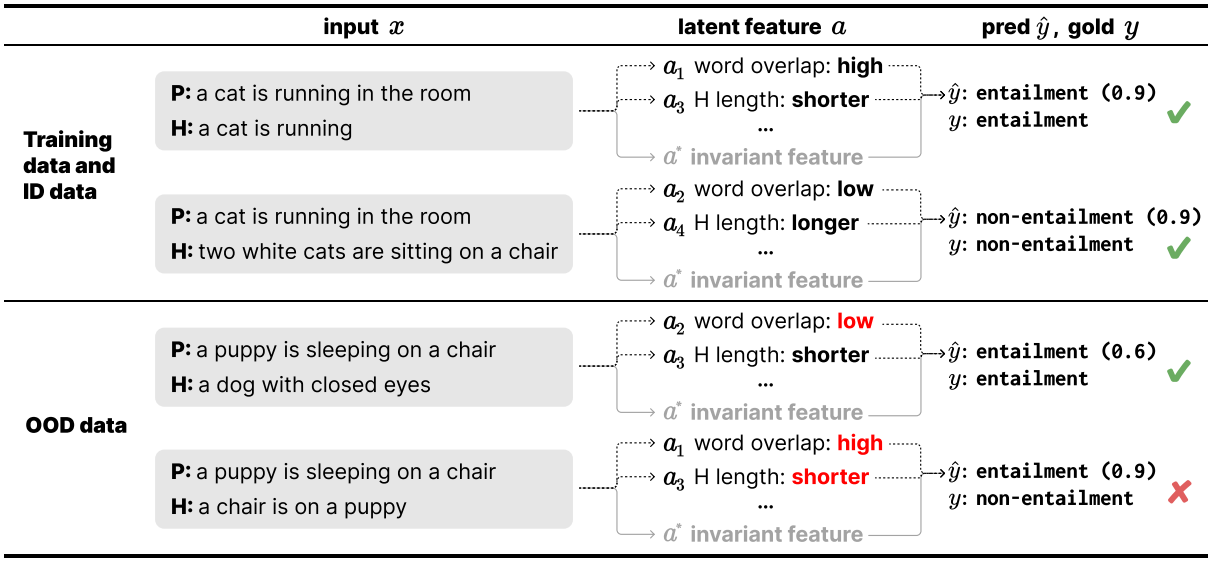


Figure 1: An illustrative example of shortcuts in the task of natural language inference. *P* and *H* denote the premise and hypothesis sentence, respectively. $\{a_i\}$ are latent features related to x. The value on the right-hand side of $\hat{y}$ shows the confidence $\in [0.0, 1.0]$ of the prediction. a_i is correctly predictive of label y in the training and ID data but not in the OOD data where the association between a_i and y changed. a^* is an ideal latent feature predictive of y across distributions. However, such a^* is generally difficult for models to rely on. This figure illustrates the common case where predictions are not based on a^*.

Even when using OOD validation data, its distribution is the same as that of test data. Thus, in other words, the approach requires knowing the test-time distribution to tune hyperparameters, which is impractical in testing OOD robustness.

In this paper, we opt not to pursue training to eliminate shortcuts. Instead, we propose to aggregate predictions of a mixture model during inference. The problem with shifts in shortcuts is that some latent features in the training data are no longer associated with the labels. We hypothesize that this OOD situation can be addressed by effectively aggregating predictions, assuming that the predictions are based on relatively different latent features. We propose a mixture model and its training strategy to encourage such modeling of latent features. At inference time, we perform theoretically grounded risk minimization strategies through post-hoc control for the predictions in the event of potential shifts in shortcuts.

The experimental results demonstrate that our method significantly enhances the model's robustness when faced with shifts in shortcuts. Moreover, our method shows two other practical benefits that address the problems of previous methods. First, the mixture weights of our model can be used to detect shifts in latent features during inference. This opens up the possibility of adaptive post-hoc control to address the performance trade-off between ID and OOD data. Second, hyperparameters can be tuned with ID data only, removing the need to tune hyperparameters with OOD data. We also analyze our mixture model and provide results supporting the assumption of modeling latent features.

2 Background

This section first overviews shortcuts. Then, we describe how previous approaches have addressed shortcuts and outline how we approach them. Below, $\mathcal{X}$ and $\mathcal{Y}$ denote the input instance space and the entire class of target labels, respectively.

2.1 Shortcuts in Detail

Shortcuts or spurious correlations arise when (1) some feature a related to input $x \in \mathcal{X}$ is predictive of label $y \in \mathcal{Y}$ in training data, (2) but this association between a and y changes under relevant distribution shifts (Makar et al., 2022; Feder et al., 2022). Often, those features are latent, that is, difficult to identify *a priori*. Among those latent features, shortcuts refer to those that

are easy to represent; sometimes, they refer to the exploitation of such latent features (Makar et al., 2022; Feder et al., 2022; Du et al., 2021; Meissner et al., 2022). Following Makar et al. (2022), we emphasize that the ease of modeling is an important characteristic of shortcuts. It enables models to capture and depend on the latent features, thereby posing a serious threat when the relevant distribution shifts.

Shortcuts are pervasive in NLU datasets due to the simple heuristics, preferences, etc., possessed by annotators (Gururangan et al., 2018; Geva et al., 2019). Shortcut-dependent models severely degrade performance on datasets collected with different heuristics and preferences (Geva et al., 2019; McCoy et al., 2019; Zhang et al., 2019; Schuster et al., 2019). Moreover, there is a more fundamental discussion that the compositional nature of natural language produces many simple features (e.g., words and phrases) that can robustly predict labels when the entire context is considered but are only spuriously correlated when considered individually (Gardner et al., 2021; Eisenstein, 2022). Figure 1 shows an illustrative example where simple word-overlap features and length features are associated with labels in training and ID data, but the association drastically changes in OOD data.

While pervasive, note that shortcuts are only part of the distribution-shift problem. For example, shortcuts can be viewed as a special case of domain shift (Feder et al., 2022) and can arise independently from the shift in label distribution $p(y)$ (Yang et al., 2023). Also, the problem of shortcuts is one of the consequences of underspecification, where distinct solutions can solve the problem equivalently well (D'Amour et al., 2022). Following Makar et al. (2022), **we address distribution shifts exclusively in terms of shortcuts**. Consequently, the OOD data we address involve shifts in the association between a and y.

2.2 Overview of Previous Approaches

To improve the OOD performance, previous studies have tried to remove the reliance on shortcuts. In the study of NLU, a widely used approach is **reweighting** (Clark et al., 2019; He et al., 2019; Mahabadi et al., 2020). This approach reweights instances to reduce learning on shortcut-inducing instances and increase learning on the others. The weights are computed based on how accurately shortcuts predict labels. During training, instances where shortcuts are predictive are down-weighted, and the others are up-weighted.

In the machine learning (ML) literature, training data is first partitioned into groups (also called environments) based on the spuriously correlated features. The training data is assumed to be a mixture of the groups divided by the features. Previous approaches avoid relying on shortcuts, that is, the group-specific spurious correlations. There are two principal approaches in the ML literature. Invariant risk minimization (**IRM**; Arjovsky et al., 2019) trains a classifier that is simultaneously optimal for all groups. Group distributionally robust optimization (**GroupDRO**; Sagawa et al., 2020) learns to minimize the worst-group risk by up-weighting the loss of the worst-case group.

2.3 Problems of Previous Approaches

These approaches share one common idea: training models while minimizing reliance on shortcuts to achieve robust predictions. In practice, however, daring to eliminate predictive features in the training data and its ID data causes deviations from the ID data, resulting in a performance trade-off between the ID and OOD data. See, for example, the aforementioned work on reweighting, IRM, and GroupDRO for empirical results. This trade-off raises the following practical problems.

(a) Overfitting to OOD Data. The degraded performance on ID data is a direct consequence of this trade-off (Utama et al., 2020a). Evaluating worst-case performance or performance on adversarial OOD data is essential for assessing generalization, and this study also aims to improve on these evaluations. However, such extreme distribution shifts do not always occur after model deployment, so it is desirable for practical purposes to be able to deal with ID data as well.

(b) Hyperparameter Tuning with OOD Data. An indirect but more fundamental problem is the need for OOD test or validation data to tune hyperparameters. This problem arises because the trade-off makes it difficult to predict performance on OOD data simply by looking at performance on ID data. Obtaining OOD test data (in the context of ML, worst-group) requires pre-identification of shortcuts and their test-time distribution. Even when using OOD validation data, its distribution needs to be the same as test data's. In testing OOD robustness, this requirement is clearly impractical.

Initial studies used training data where shortcuts are pre-identified, in addition to OOD test or validation data (Clark et al., 2019; Arjovsky et al., 2019; Sagawa et al., 2020, *inter alia*). Pre-identification of shortcuts is costly as it requires careful analysis of given data. Seeking more practical solutions, subsequent approaches followed that did not require pre-identification of shortcuts in training data (Clark et al., 2020a; Liu et al., 2021; Creager et al., 2021, *inter alia*). However, they still need OOD test or validation data related to the pre-identified shortcuts to tune hyperparameters. This requirement has been discussed as a serious limitation for practical use (Clark et al., 2019; Mahabadi et al., 2020; Clark et al., 2020a; Ghaddar et al., 2021; Liu et al., 2021; Creager et al., 2021; Yu et al., 2022). Moreover, the performance of those approaches on OOD data is considerably low without the hyperparameter tuning on OOD data (Yang et al., 2023).

2.4 Overview of Our Approach

In this study, we explore a different direction from previous approaches: We aggregate predictions of a mixture model. Our hypothesis is that effective aggregation of predictions enables addressing potential shifts in shortcuts, suppose the predictions are based on relatively different latent features.

Let a be a discrete random variable defined over the space $\mathcal{A} = \{a_1, \ldots, a_K\}$. As described in Section 2.1, NLU data are likely to have multiple latent features associated with labels. Considering the existence of those latent features, the conditional probability of y given x, $p(y|x)$, can be rewritten as a finite mixture model as follows:[3]

$$p(y|x) = \sum_{a \in \mathcal{A}} p(y, a|x), \tag{1}$$

$$= \sum_{a \in \mathcal{A}} p(y|a, x)\ p(a|x). \tag{2}$$

mixture weights

Therefore, the mixture model naturally aligns with the data structure. The *mixture weights* $p(a|x)$ are expected to estimate the distribution of the latent features, and $p(y|a, x)$ to predict based on each a. If this assumption holds, pessimistic aggregation of $p(y|a, x)$ can minimize the risk under OOD circumstances at inference time, where it is not known which latent features to rely on.

In addition, this approach allows us to address the problems described in Section 2.3 as follows. **(a′)** We have the flexibility to address both ID and OOD data scenarios through the adaptable application or omission of post-hoc control techniques.[4] This adaptability sets our approach apart from the existing methods, which typically rely on fitting a single model exclusively for either the ID or OOD case. **(b′)** Since our method focuses on fitting ID data during training, it does not require OOD data for training or tuning hyperparameters.

3 Methods

The proposed method consists of two parts. The first part is a training method using a mixture model. The second part is a test-time operation, which aggregates the mixture model's predictions to make robust predictions when facing distribution shifts. Figure 3 in Appendix A shows the overview of our method.

3.1 Training Phase: Mixture Model to Capture Latent Features

To model latent features, we employ a mixture model, as seen in Eq (3). A typical implementation of the mixture model is **mixture-of-experts (MoE)** (Jacobs et al., 1991). Shazeer et al. (2017) showed that MoE improves performance and efficiency in large-scale deep-learning models. Following these studies' success, we employ a variant of MoE in this study.

Our Implementation of MoE. MoE consists of K *expert networks (experts)* and a *router network (router)* responsible for assigning inputs to the expert networks. Hereafter, we use the terms mixture weights and output distribution of a router interchangeably because they have the same

[3] We assume that it is reasonable to consider the finite latent features. Generally speaking, model predictions are likely to depend on simple latent features strongly associated with labels, not evenly dependent on infinite possible latent features. In addition, it is empirically established that finite mixture models can approximate a wide variety of distributions, as long as a sufficient number of mixture components are included (Titterington et al., 1985; Walker and Ben-Akiva, 2011; Nguyen et al., 2020).

[4] While this adaptive use of post-hoc control necessitates determining whether the test data falls under the ID or OOD category, the results presented in Table 3 suggest that changes in the mixture weights can effectively discern this distinction at inference time. We revisit this point in Section 4.3.

function. The MoE given an input x is defined as follows:

$$\mathrm{MoE}(x) = \sum_{k=1}^{K} E^k(x)\ \pi_k(x), \tag{3}$$

($E^k(x)$: k-th expert; $\pi_k(x)$: router)

where $E^k(x)$ is the output of the k-th expert and $\pi_k(x)$ is the k-th element of the router output $\pi(x) = [\pi_1(x), \ldots, \pi_k(x)]^\top \in \mathcal{P}$, where $\mathcal{P}$ is a $(K-1)$-dimensional simplex. The structure of the expert and router networks can be arbitrarily determined. For example, the MoE modules can be stacked layer by layer while sparsifying the output of the router network (Shazeer et al., 2017).

Our goal in employing the mixture model is to enable the aggregation of predictions based on relatively different latent features. To this end, the parameters capturing latent features should be in one place. Therefore, we decided to employ the MoE module only in the final layer of classification. This is consistent with **mixture-of-softmax (MoS)**, a particular instantiation of MoE (Yang et al., 2018). Similar to MoE, MoS consists of the experts and the router. Both the expert and router receive the same encoded vector from an encoder. Then, the experts predict target labels, while the router determines the weights over the experts. Our implementation of MoS is defined as follows:

$$p_{\boldsymbol{\theta}}(y|x) = \sum_{k=1}^{K} p^k(y|x)\ \pi_k(x), \tag{4}$$

(E^k)

where we specify

$$p^k(y|x) = \frac{\exp\left(f^k(\mathbf{h})^\top \mathbf{w}_y^k + b_y^k\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(f^k(\mathbf{h})^\top \mathbf{w}_{y'}^k + b_{y'}^k\right)}, \tag{5}$$

$$\pi_k(x) = \frac{\exp\left(f^r(\mathbf{h})^\top \mathbf{v}_k\right)}{\sum_{j=1}^{K} \exp\left(f^r(\mathbf{h})^\top \mathbf{v}_j\right)}, \tag{6}$$

$$\mathbf{h} = g_{\boldsymbol{\phi}}(x). \tag{7}$$

Here, $g_{\boldsymbol{\phi}} : \mathcal{X} \to \mathbb{R}^d$ is the encoder and its d-dimensional output is denoted by $\mathbf{h}$. $\mathbf{w}_y^k$ and $\mathbf{v}_k$ are the $d \times 1$ weighting vector related to the k-th expert prediction for y and the k-th element of the router output, respectively. The functions f^k and f^r respectively transform the encoder outcome to $\mathbb{R}^d$ for the k-th expert and the router. Both functions have the same structure and size of parameters, but the parameters are initialized and updated separately. We employ BERT for $g_{\boldsymbol{\phi}}$ and set f^* to the prediction head of BERT (Devlin et al., 2019; Zhang et al., 2022): $f^*(\mathbf{h}) = \mathrm{LayerNorm} \circ \mathrm{ReLU} \circ \mathrm{Linear}(\mathbf{h})$, where $\circ$ represents composite functions that applied from right to left. $\boldsymbol{\theta}$ denotes the entire parameters above.

We use the cross-entropy loss to train the parameters. Given a mini-batch of M instances with one-hot encoding of labels, the loss is as follows:

$$\mathcal{L}_C(\boldsymbol{\theta}) = -\frac{1}{M} \sum_{m=1}^{M} \log p_{\boldsymbol{\theta}}(y_m|x_m). \tag{8}$$

Penalty Term for π: Different Experts for Different Latent Features. Comparing Eq. (2) and (4), we see that different experts are expected to capture different latent features that predict labels. However, this expectation does not hold when the mixture weights are consistently uniform or dominated by the same few experts across all the training instances. In those cases, all the experts or the few experts capture the latent features indistinguishably. To facilitate capturing the mixture of latent features at the mixture architecture, we propose a penalty term that constrains the router π. Intuitively, it encourages the router to assign different inputs to different experts, assuming that different inputs have differences in their latent features to some extent.

Given a mini-batch of size M, define a $K \times M$ matrix of the router outputs as follows:

$$\boldsymbol{\Pi} = \left[\pi(x_1), \pi(x_2), \ldots, \pi(x_M)\right]. \tag{9}$$

Our goal is to encourage the columns of $\boldsymbol{\Pi}$ to be distinct distributions from each other. Hinted by Lin et al. (2017), we accomplish this by minimizing the Frobenius norm of $\boldsymbol{\Pi}^\top\boldsymbol{\Pi}$, where the (m, m')-th element is the dot product $\pi(x_m)^\top \pi(x_{m'}) \in [0, 1]$ and represents the similarity of the two distributions. Each element of $\boldsymbol{\Pi}^\top\boldsymbol{\Pi}$ takes a maximum value of 1 when the two distributions are identical one-hot distributions and a minimum value of 0 when they have no overlap. Therefore, the Frobenius norm $\|\boldsymbol{\Pi}^\top\boldsymbol{\Pi}\|_F$ takes a large value when the similarity of the distributions in a mini-batch is high, whereas it takes a small value when the similarity is low. Using this property, Lin et al. (2017) proposed to minimize $\|\boldsymbol{\Pi}^\top\boldsymbol{\Pi} - \mathbf{I}\|_F$ as a penalty term to reduce the

Algorithm 1: Row-wise top-ℓ dropout

Input: Square matrix $\mathbf{M} \in \mathbb{R}^{M \times M}$ and ℓ
Output: Square matrix $\mathbf{M} \in \mathbb{R}^{M \times M}$

```
for i = 1, ..., M do
    m ← M_i   ▷ M_i ∈ R^M is the i-th row vector of M
    Sort(m)   ▷ Sort in descending order
    m[: ℓ] ← 0   ▷ 0 ∈ R^ℓ, drop out top-ℓ
    M_i ← m
end
return M
```

similarity of self-attention maps. $\mathbf{I} \in \mathbb{R}^{M \times M}$ is an identity matrix to encourage $\pi(x_m)$ to be one-hot.

We use this penalty term with the following modifications. First, the penalty cannot be minimized to zero when the mini-batch size M exceeds the number of experts K. During joint minimization with the classification loss $\mathcal{L}_C$, forcing the minimization of the never-zero penalty could lead models too far away from the optimal solution for $\mathcal{L}_C$. To avoid this, we consider that in the m-th row of $\mathbf{\Pi}^\top \mathbf{\Pi} - \mathbf{I}$, the corresponding expert for x_m captures the same latent features among the top-ℓ elements (instances). We then exclude those elements to allow such multi-instance assignments. Let $d_\ell : \mathbb{R}^{M \times M} \to \mathbb{R}^{M \times M}$ be a function to drop out the elements with the top-ℓ values in each row to 0 (Algorithm 1). We minimize $\|d_\ell(\mathbf{\Pi}^\top \mathbf{\Pi} - \mathbf{I})\|_F$.

Second, the penalty varies highly depending on the batch size M, as the Frobenius norm takes the sum, not the mean, of the squares of the matrix elements. To search for weighting hyperparameters robust to changes in M, we normalize the penalty into [0, 1]. We divide the penalty by $\|d_\ell(\mathbf{J} - \mathbf{I})\|_F$, where $\mathbf{J} \in \mathbb{R}^{M \times M}$ is a matrix of ones.

Taking all of these together, we define our penalty term as follows:

$$\mathcal{L}_R(\boldsymbol{\theta}) = \frac{\|d_\ell(\mathbf{\Pi}^\top \mathbf{\Pi} - \mathbf{I})\|_F}{\|d_\ell(\mathbf{J} - \mathbf{I})\|_F}. \tag{10}$$

The final loss is defined using the weighting hyperparameters λ as follows:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_C(\boldsymbol{\theta}) + \lambda \mathcal{L}_R(\boldsymbol{\theta}). \tag{11}$$

Previous studies on MoE observed that assignment was concentrated on the same few experts and proposed penalty terms to balance the assignment among experts (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022b,a). However, these penalty terms were proposed for text generation models and cannot be applied directly to the classification model in consideration. Notably, the penalty terms encourage balanced, uniform assignments but do not encourage diverse assignments that vary among groups of instances.

3.2 Inference Phase: Post-Hoc Control for Risk Minimization under Uncertainty

The problem of shortcuts emerges upon the distribution shifts where some latent features are no longer associated with labels. In this subsection, we consider controlling the mixture weights to minimize the risk under the OOD circumstances where we do not know which latent features to rely on during inference. For this control, we suppose that different experts capture those latent features with small overlaps, as encouraged in the training (see Section 3.1: penalty term). However, *note that this is not a strict requirement*, and moderate differences may be sufficient. The point is that not all experts depend on the same latent features. We introduce two post-hoc operations on π to ensure predictions remain robust to such shifts. The operations replace the estimated π with π^* according to the theory of risk minimization under uncertainty.

Uniform Weighting. The simplest way to obtain robustness to variation is to use a uniform distribution. Assuming that all experts are equally good, a simple way to obtain robustness to the unknown shifts is to consider the expert's predictions equally. We replace the estimated π with a uniform distribution as follows:

$$y^* = \arg\max_{y \in \mathcal{Y}} \underbrace{K^{-1} \sum_{k=1}^{K} p^k(y|x)}_{\pi^*}. \tag{12}$$

This operation is equivalent to taking the mean of $p^k(y|x)$ across K experts.

Argmin Weighting. In the worst-case scenario, the assumption that all experts are equally good does not hold. An alternative approach to minimize the risk of erroneous predictions in this case

is to determine the mixture weights by considering the expert model's predictions, as follows:

$$y^* = \arg\max_{y \in \mathcal{Y}} \sum_{k=1}^{K} p^k(y|x)\ \mathbb{1}\{k^*(x) = k\}, \quad (13)$$

$$k^*(x) = \arg\min_{k \in \mathcal{K}} p^k(y|x), \quad (14)$$

where $\mathcal{K} = \{1, 2, \ldots, K\}$ and $\mathbb{1}\{\cdot\}$ is the indicator function. This operation first selects the expert that minimizes the probability $p^k(y|x)$ over a set of K experts for each label, and then chooses the label that maximizes the resulting probability. See Figure 4 in Appendix A for an example.

Derivation of the Operations. The remainder of this subsection further explains the principles behind the prediction rules introduced earlier, with a focus on risk minimization. This perspective is rooted in the classical statistical decision-making framework (see Wald, 1950; Berger, 1985).

We consider a prediction function $\delta : \mathcal{X} \to \mathcal{Y}$ and define $\mathcal{D}$ as a collection of such measurable prediction functions. To evaluate the prediction, we employ the 0–1 loss $L : \mathcal{Y} \times \mathcal{Y} \to \{0, 1\}$, which measures classification error as follows:

$$L\left(y, \delta(x)\right) = 1 - \mathbb{1}\{y = \delta(x)\}. \quad (15)$$

Since x and y are random variables, the value of the loss is a random quantity.

We consider the expected value of the loss $L\left(y, \delta(x)\right)$ and call it the risk. Given that the mixture weights π are the essential elements for our analysis, we explicitly state the dependency of the risk on the weights. We denote the risk function $R : \mathcal{P} \times \mathcal{D} \to \mathbb{R}$, given by

$$R(\pi, \delta) = 1 - \mathbb{E}_x \left[\sum_{k=1}^{K} p^k\left(\delta(x)|x\right) \pi_k(x) \right], \quad (16)$$

where $\mathbb{E}_x[\cdot]$ represents the expectation operator over the outcomes of the random variable x.

If the mixture weights π are known, then it can be easily shown that the risk $R(\pi, \delta)$ is minimized by

$$\delta^*_\pi(x) = \arg\max_{y \in \mathcal{Y}} \sum_{k=1}^{K} p^k\left(y|x\right) \pi_k(x). \quad (17)$$

That is, the optimal prediction selects an element of $\mathcal{Y}$ corresponding to the maximum conditional probability. When the training and evaluation data share the same joint distributions of y and x, we can apply the above classification rule with the estimated mixture weights. However, this does not hold for OOD data.

We propose two approaches to deal with the case of unknown mixture weights. First, we assume that while we do not know the correct mixture weights, all experts are equally good. Then, we can take uniform weights for the mixture weights and minimize the risk by

$$\delta^*_u(x) = \arg\max_{y \in \mathcal{Y}} \frac{1}{K} \sum_{k=1}^{K} p^k\left(y|x\right). \quad (18)$$

Another approach takes a strategy of prudence, focusing on the worst-case scenario to guarantee the most favorable outcome among these least desirable possibilities. Consequently, the prediction performs uniformly well across the mixture set $\mathcal{P}$, aligning with the minimax principle. More precisely, the minimax problem is written as

$$\min_{\delta \in \mathcal{D}} \max_{\pi \in \mathcal{P}} R(\pi, \delta) \quad (19)$$

which is solved by the following prediction rule

$$\delta^*(x) = \arg\max_{y \in \mathcal{Y}} \min_{k \in \mathcal{K}} p^k\left(y|x\right). \quad (20)$$

The above prediction rule is a maximin criterion where one selects the element of $\mathcal{Y}$ that maximizes the minimum conditional probability.

4 Experiments

Our goal is to achieve predictions robust to distribution shifts related to shortcuts. In this section, we test whether the proposed post-hoc control improves performance on those OOD tests and analyze the mechanism based on our assumption.

4.1 Setup

This subsection describes the experimental setup. Please refer to Appendix B for further details.

Datasets. In accordance with previous research on shortcut mitigation, we experimented with three NLU datasets. These are popular datasets but are all reported to induce shortcuts, and OOD test data were later created that cannot be correctly classified by the shortcuts. Each dataset consists of

Fix	*Search*	MNLI			QQP			FEVER		
		$\mathcal{L}_C$	$\mathcal{L}_R$	$\mathcal{L}_C + \mathcal{L}_R$	$\mathcal{L}_C$	$\mathcal{L}_R$	$\mathcal{L}_C + \mathcal{L}_R$	$\mathcal{L}_C$	$\mathcal{L}_R$	$\mathcal{L}_C + \mathcal{L}_R$
1st : $\lambda = 0.0$	$K = 5$	0.706	0.379	1.084	0.378	0.299	0.677	0.440	0.674	1.114
	$K = 10$	0.433	0.127	**0.560**	0.415	0.505	0.920	0.392	0.417	**0.809**
	$K = 15$	0.501	0.301	0.802	0.253	0.245	**0.498**	0.466	0.367	0.832
		$\boldsymbol{K^* = 10}$			$\boldsymbol{K^* = 15}$			$\boldsymbol{K^* = 10}$		
2nd : $K = K^*$	$\lambda = 0.0$	0.433	0.127	0.560	0.253	0.245	0.498	0.392	0.417	0.809
	$\lambda = 0.5$	0.437	0.022	**0.459**	0.333	0.010	0.343	0.523	0.128	0.651
	$\lambda = 1.0$	0.666	0.005	0.671	0.332	0.003	**0.335**	0.416	0.132	**0.548**
		$\boldsymbol{\lambda^* = 0.5}$			$\boldsymbol{\lambda^* = 1.0}$			$\boldsymbol{\lambda^* = 1.0}$		

Table 1: Results of the two-stage hyperparameter search for the number of experts K and the loss-weighting value λ. K^* and λ^* are the optimal values that minimize $\mathcal{L}_C + \mathcal{L}_R$ on ID dev. All results are the average of two runs with different seeds.

training data, validation data drawn from the same distribution as the training data (**ID dev**), and test data where the correlation between some latent features and labels changed adversarially (**OOD test**). Following previous studies in comparison, we evaluate the accuracy.

MNLI (Williams et al., 2018) is a dataset for natural language inference (NLI) across multiple genres. Given a pair of premise and hypothesis sentences, the task is to classify the relationship between the two sentences into one of three labels: *entailment*, *contradiction*, or *neutral*. In MNLI, a shortcut arises from a spurious correlation between the word overlap of input sentences and target labels. We used its matched development set as ID dev and **HANS** (McCoy et al., 2019) as OOD test. **QQP** is a dataset for paraphrase identification. The task is to classify whether two sentences are paraphrases or not. A shortcut also arises from a spurious correlation in the word overlap of input sentences. We used its development set as ID dev and **PAWS** (Zhang et al., 2019) as OOD test. **FEVER** (Thorne et al., 2018) is a dataset for fact verification. Given two sentences of claim and evidence, the task is to classify the relation of the evidence toward the claim into either *Supports*, *Refutes*, or *Not-enough-info*. Some negative phrases in the claim sentences spuriously correlate with target labels, causing a shortcut that allows classification using only the claim sentences. We used its development set as ID dev and **FEVER Symmetric v1 and v2** (Schuster et al., 2019) as OOD tests.

Baseline and Principal Methods. We used **BERT** (`bert-base-uncased`) (Devlin et al., 2019) as the baseline and backbone for a fair comparison with previous studies. We used the last layer in the position of `[CLS]` for $\mathbf{h}$ in Eq. (7).

To compare in *a practical setting where only ID data are available for training and tuning (Yang et al., 2023)*, *we reran principal methods in that setting* using their publicly available code.

Conf-reg ♠ self-debias (Utama et al., 2020b) and **JTT** (Liu et al., 2021) use heuristics that weak models are likely to exploit shortcuts. Conf-reg ♠ self-debias reweights the loss according to predictions of a weak model while balancing the weights using predictions of a teacher model. JTT up-weights the loss of training instances that a weak model misclassified. **RISK** (Wu and Gui, 2022) considers shortcuts to be redundant features and applies feature reduction. **EIIL** (Creager et al., 2021) first estimates the groups of training instances where some shortcuts are in common and then applies IRM (see Section 2.2) using the estimated groups. **BAI** (Yu et al., 2022) extends EIIL to estimate multiple levels of groups and apply IRM multiple times accordingly. **GroupDRO$_{\text{label-group}}$** (Sagawa et al., 2020) and **ReWeightCRT** (Kang et al., 2020) are reported to perform well on OOD data when the label distribution $p(y)$ is imbalanced in ID data but is uniform in OOD data (Yang et al., 2023), while they do not aimed at addressing the shift in shortcuts. GroupDRO$_{\text{label-group}}$ minimizes the loss on the worst-case class label given groups divided only by class labels, and ReWeightCRT reweights the loss with the relative frequency of class labels.

Hyperparameters. As Eq. (11) shows, the optimal model for the proposed method is one that

Main Results: Tuning with ID Dev							
	MNLI		**QQP**		**FEVER**		
	ID	OOD	ID	OOD	ID	OOD	
	Dev	**HANS**	**Dev**	**PAWS**	**Dev**	**Symm. v1**	**Symm. v2**
BERT	$\underline{84.4}_{\pm 0.2}$	$55.2_{\pm 4.2}$	$\underline{91.5}_{\pm 0.1}$	$36.7_{\pm 3.1}$	$86.7_{\pm 0.2}$	$58.5_{\pm 1.4}$	$65.1_{\pm 1.5}$
+ MoS	$\underline{84.4}_{\pm 0.1}$	$59.4_{\pm 5.5}$	$91.4_{\pm 0.1}$	$34.9_{\pm 1.6}$	$87.0_{\pm 0.5}$	$58.9_{\pm 1.2}$	$65.5_{\pm 1.0}$
$\rightarrow$ Uniform	$83.0_{\pm 1.0}$	$63.6_{\pm 5.7}$	$89.1_{\pm 2.4}$	$47.0_{\pm 8.6}$	$\underline{87.6}_{\pm 1.2}$	$\underline{\mathbf{62.2}}_{\pm 0.7}$	$\underline{\mathbf{68.2}}_{\pm 1.0}$
$\rightarrow$ Argmin	$81.0_{\pm 3.1}$	$\underline{\mathbf{67.2}}_{\pm 4.6}$	$83.8_{\pm 7.3}$	$\underline{\mathbf{55.7}}_{\pm 8.5}$	$85.3_{\pm 6.8}$	$61.8_{\pm 1.2}$	$67.4_{\pm 2.2}$
Conf-reg ♠ $_{\text{self-debias}}$	$84.5_{\pm 0.2}$	$63.7_{\pm 2.4}$	$90.5_{\pm 0.2}$	$31.0_{\pm 1.7}$	$87.1_{\pm 0.7}$	$59.7_{\pm 1.3}$	$66.5_{\pm 1.1}$
Conf-reg ♠ $^{\text{last}}_{\text{self-debias}}$	$84.5_{\pm 0.2}$	$63.7_{\pm 2.4}$	$90.5_{\pm 0.2}$	$31.0_{\pm 1.7}$	$86.7_{\pm 0.4}$	$59.3_{\pm 1.2}$	$65.9_{\pm 1.1}$
JTT	$80.7_{\pm 0.3}$	$57.3_{\pm 2.2}$	$89.4_{\pm 0.2}$	$36.0_{\pm 0.6}$	$82.7_{\pm 1.1}$	$53.0_{\pm 2.6}$	$60.3_{\pm 2.6}$
RISK	$83.9_{\pm 0.3}$	$56.3_{\pm 4.2}$	$90.5_{\pm 0.1}$	$34.8_{\pm 3.2}$	$87.6_{\pm 0.8}$	$58.9_{\pm 2.6}$	$65.9_{\pm 1.6}$
EIIL	$83.9_{\pm 0.2}$	$61.5_{\pm 2.4}$	$91.1_{\pm 0.2}$	$31.0_{\pm 0.6}$	$86.8_{\pm 1.1}$	$56.2_{\pm 1.9}$	$63.8_{\pm 1.7}$
+ BAI	$83.7_{\pm 0.2}$	$62.0_{\pm 2.1}$	$91.2_{\pm 0.2}$	$31.2_{\pm 0.3}$	$86.3_{\pm 1.2}$	$56.0_{\pm 2.1}$	$63.6_{\pm 1.9}$
GroupDRO$_{\text{label-group}}$	$84.3_{\pm 0.3}$	$57.7_{\pm 2.9}$	$\mathbf{91.6}_{\pm 0.1}$	$34.6_{\pm 3.7}$	$\mathbf{89.3}_{\pm 0.2}$	$62.1_{\pm 1.1}$	$67.9_{\pm 1.3}$
ReWeightCRT	$\mathbf{84.6}_{\pm 0.1}$	$55.8_{\pm 0.3}$	$91.5_{\pm 0.0}$	$32.0_{\pm 0.3}$	$88.5_{\pm 0.0}$	$61.3_{\pm 0.4}$	$66.9_{\pm 0.2}$

Table 2: Main results. The results of our method are colored in the background. All the scores are shown in the mean and standard deviation of five runs with different seeds. The highest *mean* scores are shown in **bold**, and the highest *mean* scores within the baseline and our method are underlined.

can accurately classify x and output diverse $\pi(x)$. Therefore, we define the optimal hyperparameters for the proposed method as those that minimize the sum of the two losses ($\mathcal{L}_C + \mathcal{L}_R$) on **ID dev**.

In the proposed method, the number of experts K in Eq. (4), the number of row-wise dropouts ℓ in Eq. (10), and the loss-weighting value λ in Eq. (11) are model-specific hyperparameters. We explored the values of $K \in \{5, 10, 15\}$ and $\lambda \in \{0.0, 0.5, 1.0\}$. For an efficient search, we conducted a two-stage search. At the first stage, we fixed $\lambda = 0$ and determined K^* that naturally fit the data. Then, we searched for the optimal balance of losses λ under K^*. Table 1 shows the results of the hyperparameter search. Across settings, the value of ℓ was set to be the smallest value in 2^n that satisfies $\min(K) \cdot \ell \geq M$. This ensures that $\mathcal{L}_R$ in each mini-batch of size M can be zero in all settings when π is maximally diverse: when a different expert is allocated to every ℓ instances with probability one. We used parallel processing of two mini-batches of $M = 32$ each and $\min(K) = 5$, so we set $\ell = 8$ to satisfy the condition. Regarding epochs, we set the training epoch to 10 and the learning rate to 2e-5 for all datasets and select the best epoch on ID dev scores without applying post-hoc control.

When rerunning the comparison methods, we set all hyperparameters to the values specified in the papers or the official implementation, except for an annealing hyperparameter α of Conf-reg ♠ $_{\text{self-debias}}$, as it was tuned on OOD tests. We took the best epoch on ID dev for all the methods and the best α on ID dev for Conf-reg ♠ $_{\text{self-debias}}$.

4.2 Results

As the main results, we demonstrate that our post-hoc control over the experts achieves robust predictions on OOD test data. Table 2 shows the results in the setting where no shortcut is pre-identified. **BERT** is the baseline, **+ MoS** is our mixture model, and $\rightarrow$ **Uniform / Argmin** performs the post-hoc control on the mixture model. Since scores on the OOD tests have been reported to have high variance, all the results are shown in the mean and standard deviation of five runs with different seeds in accordance with previous studies. We observe that in all datasets, our post-hoc control significantly improves performance on the OOD tests from the baseline and MoS.

The comparison methods do not improve performance on the OOD tests much when tuned solely with ID data,[5] which is consistent with the

[5]Conf-reg ♠ $_{\text{self-debias}}$ reported taking the last epoch of arbitrarily determined epochs rather than ID dev best epoch. We also reported the performance of the last-epoch models (Conf-reg ♠ $^{\text{last}}_{\text{self-debias}}$), but we found that this practice did not work well when its annealing hyperparameter α (see Section 4.1) was tuned solely with ID data.

observation in Yang et al. (2023). As an exception, GroupDRO$_{\text{label-group}}$ and ReWeightCRT perform well on FEVER, where the difference from our method is marginal considering the standard deviation. This is because the label distribution of FEVER shifts as these methods suppose,[6] which is also consistent with the observation in Yang et al. (2023). However, they do not improve the OOD performance on MNLI and QQP, which have no such label distribution shift. In contrast, our method does not exploit assumptions on label distribution shifts but consistently improves the OOD performance across all the datasets.

4.3 Analyses

Now, we turn to the mechanism behind our method's robust performance and analyze the mixture model based on our assumption.

Analysis 1: Penalty Term $\mathcal{L}_R$ in ID and OOD Data. We first analyze the penalty term $\mathcal{L}_R$, the essential statistic of our mixture model. Recall that $\mathcal{L}_R$ encourages the router to assign different inputs to different experts, assuming different inputs have some difference in their latent features.[7] In other words, $\mathcal{L}_R$ measures the sensitivity to the difference in inputs. Drawing an inference from this, we expect that the value of $\mathcal{L}_R$ differs in the shifts related to latent features, that is, shifts between ID and OOD data we address.

Table 3 shows the value of $\mathcal{L}_R$ on the ID and OOD datasets.[8] In all the datasets, the values of $\mathcal{L}_R$ differ significantly between ID and OOD data, indicating that $\mathcal{L}_R$ is sensitive to the distribution shifts in these data.

From a practical perspective, this sensitivity may provide an advantage. We can compute $\mathcal{L}_R$ during inference since its computation does not require annotated labels either in training or inference. Therefore, during inference, we can determine which data to perform the post-hoc control on by looking at how different $\mathcal{L}_R$ is from that on ID data. While the post-hoc control decreases the ID dev scores, MoS performs the same as the baseline on the ID dev, regardless of the training with the penalty term (Table 2). Thus, adaptively applying the post-hoc control enables handling both ID and OOD data. This adaptive use is an advantage over previous methods, which only obtain a single model fitted to either OOD or ID data. However, note that it is limited to when involving a major shift in the distribution of latent features. Since we do not precisely know the threshold for how much difference should be regarded as a threatening shift, it may be difficult to determine in data such as FEVER, where the difference is significant but relatively small.

Interestingly, FEVER differs from the others in how $\mathcal{L}_R$ changes between ID and OOD data. While the others have lower $\mathcal{L}_R$ on ID data and higher $\mathcal{L}_R$ on OOD data, the opposite is true on FEVER. This suggests that the mixture model does not model latent features well on FEVER, and in fact, the performance improvement by performing the post-hoc control is relatively small on FEVER ($\boldsymbol{\Delta_{\text{MoS}\rightarrow\text{Argmin}}}$). Shortcuts in FEVER depend on very local patterns: particular phrases contained only in claim sentences. Our method uses the highly abstracted final-layer features of BERT

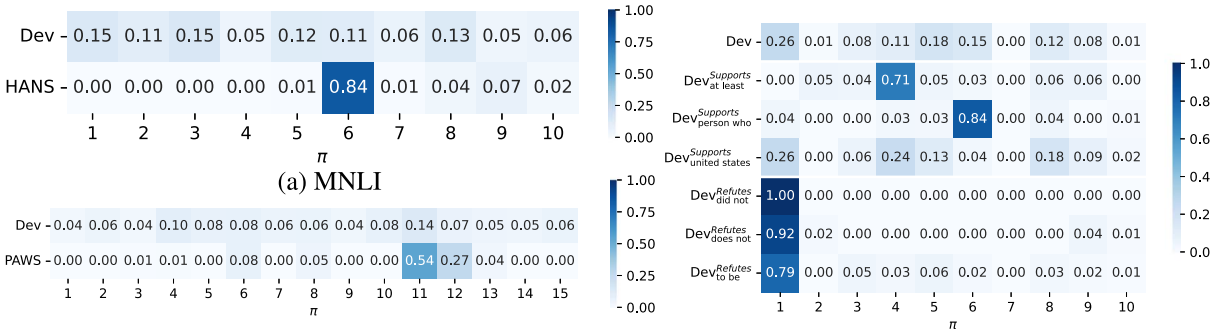
	$\mathcal{L}_R$	$\Delta_{\text{MoS}\rightarrow\text{Argmin}}$
MNLI		
Train	$0.020_{\pm 0.014}$	–
Dev	$0.017_{\pm 0.009}$	-3.4 $_{(84.4\rightarrow 81.1)}$
HANS	$\mathbf{0.633}_{\pm 0.075}$	$\mathbf{+7.8}$ $_{(59.4\rightarrow 67.2)}$
QQP		
Train	$0.002_{\pm 0.001}$	–
Dev	$0.003_{\pm 0.001}$	-7.6 $_{(91.4\rightarrow 83.8)}$
PAWS	$\mathbf{0.329}_{\pm 0.133}$	$\mathbf{+20.8}$ $_{(34.9\rightarrow 55.7)}$
FEVER		
Train	$0.010_{\pm 0.001}$	–
Dev	$\mathbf{0.136}_{\pm 0.027}$	-1.7 $_{(87.0\rightarrow 85.3)}$
Symm. v1	$0.082_{\pm 0.016}$	$\mathbf{+2.9}$ $_{(58.9\rightarrow 61.8)}$
Symm. v2	$0.087_{\pm 0.028}$	$+1.9$ $_{(65.5\rightarrow 67.4)}$

Table 3: The value of $\mathcal{L}_R$ and difference in before and after performing the post-hoc control ($\Delta_{\text{MoS}\rightarrow\text{Min}}$). The scores were obtained with five runs of different seeds. We **bold** the worst *mean* score of $\mathcal{L}_R$ and the best gain in $\Delta_{\text{MoS}\rightarrow\text{Min}}$.

[6]The label distribution of FEVER is approximately Supports:Refutes:Not-enough-info = 2:1:2 in the training data but Supports:Refutes:Not-enough-info = 1:1:0 in both ID dev and OOD test. Thus, supposing the flat label distribution for Supports and Refutes improves the OOD performance even without addressing shortcuts.

[7]While not an inevitable consequence of this objective nor a requirement for our method, we analyzed how different experts output different predictions. Figure 5 in Appendix A shows a significant variance between experts' predictions.

[8]HANS is sorted by the type of shortcuts, so we shuffled the order before computing $\mathcal{L}_R$. We did not observe this kind of sorted pattern in the other datasets.



(a) MNLI

(b) QQP

(c) FEVER

Figure 2: The mixture weights averaged on each split of the datasets. Each split, excluding the ID dev (Dev), has its own dominant feature. For FEVER, $\text{Dev}^{label}_{\text{bigram}}$ consists of the instances in ID dev that contain the bigram reported to strongly correlate with the label (Schuster et al., 2019). Here, no post-hoc control is performed on the mixture weights.

and may not be good at successfully isolating the effects of the local patterns. The features $\mathbf{h}$ in Eq. (7) can be modified arbitrarily, so we leave more effective encoding methods for future work.

Analysis 2: Captured Latent Features and their Interpretability. Our post-hoc control supposes that different experts capture different latent features to some extent. The robust performance of our post-hoc control supports this assumption but not directly. Toward direct validation, we analyze which experts a particular feature is assigned to.

To this end, we use some data splits in which a specific feature, known as a shortcut, is dominant. For MNLI and QQP, high word overlap is a dominant feature in HANS and PAWS, while their creation process is different. The sentence pairs in HANS were created by replacing or partially deleting some parts of premise sentences, while those in PAWS by word swapping, back translation, and human post-processing. However, FEVER has no such split where a single feature is dominant. To obtain such splits in FEVER, we extracted instances that contain frequent bigrams that are reported to strongly correlate with labels. The bigrams ‘‘at least’’, ‘‘person who’’, and ‘‘united states’’ strongly correlate with the *Supports* label and ‘‘did not’’, ‘‘does not’’, and ‘‘to be’’ with the *Refutes* label (Schuster et al., 2019).[9] We created six splits from FEVER ID dev. All the features above are known to be shortcuts.

Figure 2 shows the mixture weights averaged on each split of the datasets. Note that no post-hoc control is performed on the mixture weights here. Overall, a single or a few experts dominate the mixture weights in the splits where specific features are dominant: HANS, PAWS, and the newly created FEVER splits. We also observe that the dominant experts differ among the FEVER splits. Although the features under analysis are limited, this behavior of the mixture weights aligns with our assumption that different experts capture different latent features. As an exception, there are no dominant experts in the data split of ‘‘united states’’, and the same expert is dominant in the data splits of ‘‘did not’’, ‘‘does not’’, and ‘‘to be.’’ However, note that the assumption does not need to hold completely (see Section 3.2), and such local bigram features may be exceptionally difficult to capture in the current encoding (see Section 4.3). Taking these points into account, we consider that the results as a whole support the assumption.

Figure 2 also suggests that the mixture weights provide some degree of interpretability: Instances assigned to the same expert are likely to have the same features in common. Although our mixture model does not specify captured features by itself, the suggested interpretability may allow us to discover new prominent features in data by analyzing the commonalities of the instances assigned to the same expert. The discovery of commonalities in each expert will serve to support the assumption further. We leave this direction as our future work.

Analysis 3: Ablation Study on Mixture Model. We analyzed the contribution to the performance with respect to the hyperparameters of our mixture model: the number of experts K, the number of row-wise dropouts ℓ, and the loss-weighting value

[9] We omitted *Not-Enough-Info* labels since the FEVER ID dev and OOD tests have no instances with this gold label.

	MoS		→ Uniform	→ Argmin
	Dev	**HANS**	**HANS**	**HANS**
$\boldsymbol{K = 10}$	$\mathbf{84.4}_{\pm 0.1}$	$59.4_{\pm 5.5}$	$63.6_{\pm 5.7}$	$\mathbf{67.2}_{\pm 4.6}$
$K = 5$	$84.3_{\pm 0.2}$	$58.8_{\pm 5.9}$	$59.7_{\pm 5.9}$	$60.9_{\pm 4.3}$
$K = 15$	$\mathbf{84.4}_{\pm 0.2}$	$\mathbf{61.1}_{\pm 5.9}$	$\mathbf{64.5}_{\pm 10.0}$	$65.2_{\pm 6.4}$
$\boldsymbol{\lambda = 0.5}$	$84.4_{\pm 0.1}$	$59.4_{\pm 5.5}$	$\mathbf{63.6}_{\pm 5.7}$	$\mathbf{67.2}_{\pm 4.6}$
$\lambda = 0.0$	$\mathbf{84.5}_{\pm 0.0}$	$57.6_{\pm 4.8}$	$60.0_{\pm 4.2}$	$60.7_{\pm 4.4}$
$\lambda = 1.0$	$84.1_{\pm 0.3}$	$\mathbf{60.3}_{\pm 4.1}$	$57.0_{\pm 4.0}$	$65.0_{\pm 3.9}$
$\boldsymbol{\ell = 8}$	$84.4_{\pm 0.1}$	$59.4_{\pm 5.5}$	$\mathbf{63.6}_{\pm 5.7}$	$\mathbf{67.2}_{\pm 4.6}$
$\ell = 0$	$\mathbf{84.5}_{\pm 0.1}$	$58.1_{\pm 5.2}$	$57.6_{\pm 4.7}$	$58.1_{\pm 4.6}$
$\ell = 16$	$\mathbf{84.5}_{\pm 0.1}$	$\mathbf{59.8}_{\pm 2.7}$	$61.0_{\pm 2.5}$	$61.4_{\pm 3.5}$
DeBERTa$_{\text{v3-large}}$	$91.8_{\pm 0.0}$	$66.3_{\pm 1.8}$	$60.8_{\pm 10.3}$	$74.4_{\pm 8.4}$

Table 4: Ablation study on MNLI. The results of the best hyperparameters on the ID dev are colored in the background. The scores were obtained with five runs of different seeds. The highest *mean* scores are shown in **bold**.

λ. Table 4 shows the ablation study on MNLI. This table shows how performance changes by varying one of the hyperparameters from the values determined to be optimal on the ID dev. There is little to no difference in performance on the ID dev for any given value, but the OOD performance with post-hoc control is best for nearly all the values determined to be best on the ID dev. It is also worth noting that using our $\mathcal{L}_R$ and top-ℓ dropout consistently improves OOD performance better than without using them (when λ or ℓ is zero). These results indicate the effectiveness of the proposed training strategy and hyperparameter search.

We also tested DeBERTa$_{\text{v3-large}}$ (He et al., 2023) for the encoder g_ϕ. It has around three times larger parameters than the BERT we used and performs better than BERT$_{\text{large}}$, RoBERTa$_{\text{large}}$ (Liu et al., 2019), XLNet$_{\text{large}}$ (Yang et al., 2019), ELECTRA$_{\text{large}}$ (Clark et al., 2020b), etc., on MNLI (He et al., 2023). We conducted the same hyperparameter search for DeBERTa$_{\text{v3-large}}$ and found the best hyperparameters were exactly the same as BERT's. The results show that the larger model significantly improves not only ID performance but also OOD performance. However, there is still a gap between ID and OOD performance, and applying the proposed method further improves OOD performance. These results indicate that even for large models, our method is effective in improving OOD performance.

Analysis 4: Identifiability of Finite Mixture. The empirical results clearly demonstrate the effectiveness of our approach. Nevertheless, another limitation of this work is that we do not provide a theoretical guarantee for our mixture model to capture latent features within the data. This issue has previously been studied in the statistical literature and is referred to as the identification problem of finite mixtures. See Huang and Yao (2012), Compiani and Kitamura (2016), and Xiang et al. (2019) for the recent development of finite mixture models. As explained by Compiani and Kitamura (2016) among others, the identification of a finite mixture model is accomplished when predictors have a distinct influence on both the outcome prediction and mixture weights. Consistent with this, our penalty term $\mathcal{L}_R$ is designed to ensure the experts and router play distinct roles in determining the conditional outcome probabilities and the mixture weights. This approach allows our model to effectively capture and reflect the significant variations found within the data. From our empirical Analyses 1 and 3, the penalty term $\mathcal{L}_R$ is indeed understood as an important source of identifying mixture weights. Since our main focus is the excellent performance of our approach in NLU applications, we plan to leave the theoretical analysis of identification for future work.

5 Related Work

As seen in Section 2.1, datasets for NLU tasks are known to have multiple shortcuts due to the simple heuristics, preferences, etc., possessed by annotators (Gururangan et al., 2018; Geva et al., 2019), or more fundamentally, the compositional nature of natural language (Gardner et al., 2021). A number of studies have addressed the problem of shortcuts in NLU, but their primary difference lies in prior knowledge of shortcuts.

Known Shortcut Setting. This setting allows models to know the existence and details of shortcuts in advance. Previous studies used this prior knowledge to mitigate the identified shortcuts.

Reweighting is the basic strategy of previous methods. They used shortcut-dependent models that only take shortcut features as input, e.g., word overlap (Clark et al., 2019; He et al., 2019; Mahabadi et al., 2020). These shortcut-dependent models let main models know which training instances cannot be predicted correctly via shortcuts and thus should be up-weighted. Xiong et al. (2021) showed that the performance of these methods was further enhanced by calibrating

the uncertainty of the shortcut-dependent models. Utama et al. (2020a) additionally employed a teacher model to adjust the weights so that a main model would not deviate too much from the distribution of training data. Belinkov et al. (2019) and Stacey et al. (2020) trained a main model adversarially to a shortcut-dependent classifier.

Izmailov et al. (2022) and Kirichenko et al. (2023) first trained a model on ID data and then re-trained its last classification layer on a small amount of OOD data, showing that this small parameter update for the ID-fitted model is enough to improve OOD performance.

Several approaches used the counterfactual framework of causal inference. To make counterfactual predictions unaffected by shortcuts, Tian et al. (2022) and Niu et al. (2021) combined predictions of a main model and a shortcut-dependent model. Wang and Culotta (2020) classified features into genuine or spurious and selected genuine features for predictions. Others utilized identified spurious features to train model predictions to be invariant to interventions on the spurious features (Veitch et al., 2021; Makar et al., 2022; Puli et al., 2022b).

The above methods effectively mitigate shortcuts but require the significant cost of careful analysis to achieve the prior knowledge of shortcuts.

Unknown Shortcut Setting. The existence and details of shortcuts are generally unknown. Another line of studies has sought a way to mitigate shortcuts without the cost of manual identification.

The basic strategy is reweighting, just as in the known shortcut setting. To estimate the weights, previous methods have utilized the heuristics that weak models are likely to exploit shortcuts. The weak models include models with limited capacity (Clark et al., 2020a; Sanh et al., 2021), models trained with a limited number of data (Utama et al., 2020b), a single epoch (Du et al., 2021), or shallow layers (Ghaddar et al., 2021; Wang et al., 2022). While these methods used continuous weights, Yaghoobzadeh et al. (2021) used binary weights that take the value of 1 only for training instances that a weak model misclassified. Other approaches applied reweighting when learning to prune fully trained models (Meissner et al., 2022; Du et al., 2023; Liu et al., 2022).

Some studies addressed shortcuts in the feature space by removing redundancy (Wu and Gui, 2022) or correlations (Dou et al., 2022; Gao et al., 2022) in the space. These studies reported the best scores at different epochs for each of the ID validation data and OOD test data, so their results are not directly comparable to the other studies.

The above methods still require OOD test data related to pre-identified shortcuts to tune hyperparameters, as described in Section 2.3. Our method is different from them in that the training and tuning can be conducted solely on ID data. In Section 4.2, we demonstrated that in the setting of *fully unknown shortcuts* where only ID data are available, our method improves the performance on OOD data significantly better than the previous methods.

Additional Data. Other studies make use of additional data to mitigate shortcuts. Counterfactual data augmentation is one such study. Counterfactual data were generated using manual annotation (Kaushik et al., 2021), known shortcuts (Wu et al., 2022), or large language models (Wen et al., 2022; Chen et al., 2023). Other studies used human explanation (Stacey et al., 2022a,b) or human gaze signals (Ren and Xiong, 2023) as additional supervision to guide models during training. Although effective, collecting these external data is cost-intensive and requires additional training.

Literature Outside of NLU. Outside of NLU, shortcuts have been addressed in the ML literature as one of the broader OOD problems (Krueger et al., 2021; Yang et al., 2023). Still, many methods used in ML and NLU tasks have the same concepts in common, such as reweighting (Nam et al., 2020; Liu et al., 2021; Clark et al., 2020a; Utama et al., 2020b), IRM (Creager et al., 2021; Yu et al., 2022), counterfactual invariance (Veitch et al., 2021; Makar et al., 2022; Puli et al., 2022b), and data augmentation (Yao et al., 2022; Puli et al., 2022a; Wu et al., 2022). As described in Section 2.2, IRM (Arjovsky et al., 2019) and GroupDRO (Sagawa et al., 2020) are the two principal approaches. These approaches considered the known shortcut setting, and similar to NLU literature, their follow-up approaches have sought to address shortcuts in the unknown shortcut setting (Nam et al., 2020; Liu et al., 2021; Creager et al., 2021; Yao et al., 2022; Puli et al., 2022a; Izmailov et al., 2022; Kirichenko et al., 2023). However, also similar to NLU literature, those follow-up approaches still require the shortcuts

that shift in test data to be pre-identified in validation data (Yang et al., 2023).

6 Conclusion

This study proposed a conceptually novel approach to address the shortcuts problem by pessimistically aggregating the mixture model's predictions at inference time. We introduced the MoE-based model, a penalty term to encourage different experts to capture different latent features, and post-hoc control for the mixture weights that is theoretically grounded in risk minimization. The experimental results show that our method not only significantly enhances the model's robustness to shifts in shortcuts but also provides additional benefits to address the previous methods' problems: the performance trade-off between ID and OOD data and the need for OOD test or validation data to tune hyperparameters.

Our analyses provided results supporting the assumption: Different experts capture different latent features to some extent. However, we also noted the limitations in the encoding method (Analysis 1), the tested features and interpretability (Analysis 2), and the theoretical guarantee of identifiability (Analysis 4). Future work includes improving the encoding method to capture latent features more accurately, analyzing the instances assigned to the same expert to interpret what it captures and further support the assumption, and theoretically accounting for how the penalty term enhances identifiability. While the focus of this study is on shortcuts, another future direction is extending our method to address a broader range of OOD problems (see Section 2.1). We believe these are interesting future research departing from this study.

Acknowledgments

We thank Jacob Eisenstein, who served as our TACL action editor, and the anonymous reviewers for their insightful comments.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893v3*.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1084

James O. Berger. 1985. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York. https://doi.org/10.1007/978-1-4757-4286-2

Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. 2023. DISCO: Distilling counterfactuals with large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5514–5528, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.1353/dis.2023.a923671

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1418

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020a. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031–3045, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.272

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020b. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Giovanni Compiani and Yuichi Kitamura. 2016. Using mixtures in econometric models: A brief review and some new results. *The Econometrics Journal*, 19(3):C95–C127. https://doi.org/10.1111/ectj.12068

Elliot Creager, Joern-Henrik Jacobsen, and Richard Zemel. 2021. Environment inference for invariant learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 2189–2200. PMLR.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. 2022. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Shihan Dou, Rui Zheng, Ting Wu, SongYang Gao, Junjie Shan, Qi Zhang, Yueming Wu, and Xuanjing Huang. 2022. Decorrelate irrelevant, purify relevant: Overcome textual spurious correlations from a feature perspective. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2278–2287, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of NLU models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.71

Mengnan Du, Subhabrata Mukherjee, Yu Cheng, Milad Shokouhi, Xia Hu, and Ahmed Hassan Awadallah. 2023. Robustness challenges in model distillation and pruning for natural language understanding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1766–1778, Dubrovnik, Croatia. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.eacl-main.129

Jacob Eisenstein. 2022. Informativeness and invariance: Two perspectives on spurious correlations in natural language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4326–4331, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.321

Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. 2022. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Transactions of the Association for Computational Linguistics*, 10:1138–1158. https://doi.org/10.1162/tacl_a_00511

William Fedus, Jeff Dean, and Barret Zoph. 2022a. A review of sparse expert models in deep learning. *arXiv preprint arXiv:2209.01667v1*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022b. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

SongYang Gao, Shihan Dou, Qi Zhang, and Xuanjing Huang. 2022. Kernel-whitening: Overcome dataset bias with isotropic sentence embedding. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4112–4122, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.275

Matt Gardner, William Merrill, Jesse Dodge, Matthew Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.135

Mor Geva, Yoav Goldberg, and Jonathan Berant. 2019. Are we modeling the task or the annotator? An investigation of annotator bias in natural language understanding datasets. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1161–1166, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1107

Abbas Ghaddar, Phillippe Langlais, Mehdi Rezagholizadeh, and Ahmad Rashid. 2021. End-to-end self-debiasing framework for robust NLU training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1923–1929, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.168

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-2017

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-6115

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Mian Huang and Weixin Yao. 2012. Mixture of regression models with varying mixing proportions: A semiparametric approach. *Journal of the American Statistical Association*, 107(498):711–724. https://doi.org/10.1080/01621459.2012.682541

Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew G. Wilson. 2022. On feature learning in the presence of spurious correlations. In *Advances in Neural Information Processing Systems*, volume 35, pages 38516–38532. Curran Associates, Inc.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87. https://doi.org/10.1162/neco.1991.3.1.79, PubMed: 31141872

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. 2020. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*.

Divyansh Kaushik, Amrith Setlur, Eduard H. Hovy, and Zachary Chase Lipton. 2021. Explaining the efficacy of counterfactually augmented data. In *International Conference on Learning Representations*.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. 2023. Last layer re-training

is sufficient for robustness to spurious correlations. In *The Eleventh International Conference on Learning Representations*.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. 2021. Out-of-distribution generalization via risk extrapolation (rex). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5815–5826. PMLR.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *International Conference on Learning Representations*.

Evan Z. Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. 2021. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR.

Yuanxin Liu, Fandong Meng, Zheng Lin, Jiangnan Li, Peng Fu, Yanan Cao, Weiping Wang, and Jie Zhou. 2022. A win-win deal: Towards sparse and robust pre-trained language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 19189–19202. Curran Associates, Inc.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692v1*.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.769

Maggie Makar, Ben Packer, Dan Moldovan, Davis Blalock, Yoni Halpern, and Alexander D'Amour. 2022. Causally motivated shortcut removal using auxiliary labels. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 739–766. PMLR.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1334

Johannes Mario Meissner, Saku Sugawara, and Akiko Aizawa. 2022. Debiasing masks: A new framework for shortcut mitigation in NLU. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7607–7613, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.517

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. 2020. Learning from failure: De-biasing classifier from biased classifier. In *Advances in Neural Information Processing Systems*, volume 33, pages 20673–20684. Curran Associates, Inc.

T. Tin Nguyen, Hien D. Nguyen, Faicel Chamroukhi, and Geoffrey J. McLachlan. 2020. Approximation by finite mixtures of continuous density functions that vanish at infinity. *Cogent Mathematics & Statistics*, 7(1):1750861. https://doi.org/10.1080/25742558.2020.1750861

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2021. Counterfactual VQA: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer*

Vision and Pattern Recognition (CVPR), pages 12700–12710. https://doi.org/10.1109/CVPR46437.2021.01251

Aahlad Puli, Nitish Joshi, He He, and Rajesh Ranganath. 2022a. Nuisances via negativa: Adjusting for spurious correlations via data augmentation. *arXiv preprint arXiv:2210.01302v2*.

Aahlad Manas Puli, Lily H. Zhang, Eric Karl Oermann, and Rajesh Ranganath. 2022b. Out-of-distribution generalization in the presence of nuisance-induced spurious correlations. In *International Conference on Learning Representations*.

Yuqi Ren and Deyi Xiong. 2023. HuaSLIM: Human attention motivated shortcut learning identification and mitigation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12350–12365, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.781

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks. In *International Conference on Learning Representations*.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M. Rush. 2021. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *International Conference on Learning Representations*.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425, Hong Kong, China. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1341

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Joe Stacey, Yonatan Belinkov, and Marek Rei. 2022a. Supervising model attention with human explanations for robust natural language inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11349–11357. https://doi.org/10.1609/aaai.v36i10.21386

Joe Stacey, Pasquale Minervini, Haim Dubossarsky, and Marek Rei. 2022b. Logical reasoning with span-level predictions for interpretable and robust NLI models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3809–3823, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.251

Joe Stacey, Pasquale Minervini, Haim Dubossarsky, Sebastian Riedel, and Tim Rocktäschel. 2020. Avoiding the hypothesis-only bias in natural language inference via ensemble adversarial training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8281–8291, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.665

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: A large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1074

Bing Tian, Yixin Cao, Yong Zhang, and Chunxiao Xing. 2022. Debiasing NLU models via causal intervention and counterfactual reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11376–11384. https://doi.org/10.1609/aaai.v36i10.21389

Donald M. Titterington, Adrian F. M. Smith, and Ehud Makov. 1985. *Statistical Analysis of Finite Mixture Distributions*. Applied section. Wiley.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020a. Mind the trade-off: Debiasing NLU models without degrading the in-distribution performance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8717–8729, Online. Association for Computational Linguistics.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020b. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.613

Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. 2021. Counterfactual invariance to spurious correlations in text classification. In *Advances in Neural Information Processing Systems*, volume 34, pages 16196–16208. Curran Associates, Inc.

Abraham Wald. 1950. *Statistical Decision Functions*. Wiley: New York.

Joan L. Walker and Moshe Ben-Akiva. 2011. *Advances in Discrete Choice: Mixture Models*, chapter 8. Edward Elgar Publishing, Cheltenham, UK. https://doi.org/10.4337/9780857930873.00015

Fei Wang, Zhewei Xu, Pedro Szekely, and Muhao Chen. 2022. Robust (controlled) table-to-text generation with structure-aware equivariance learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5037–5048, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.371

Zhao Wang and Aron Culotta. 2020. Identifying spurious correlations for robust text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3431–3440, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.308

Jiaxin Wen, Yeshuang Zhu, Jinchao Zhang, Jie Zhou, and Minlie Huang. 2022. AutoCAD: Automatically generate counterfactuals for mitigating shortcut learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2302–2317, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-emnlp.170

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1101

Ting Wu and Tao Gui. 2022. Less is better: Recovering intended-feature subspace to robustify NLU models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1666–1676, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yuxiang Wu, Matt Gardner, Pontus Stenetorp, and Pradeep Dasigi. 2022. Generating data to mitigate spurious correlations in natural language inference datasets. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2660–2676, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.190

Sijia Xiang, Weixin Yao, and Guangren Yang. 2019. An overview of semiparametric extensions of finite mixture models. *Statistical Science*, 34(3):391–404. https://doi.org/10.1214/19-STS698

Ruibin Xiong, Yimeng Chen, Liang Pang, Xueqi Cheng, Zhi-Ming Ma, and Yanyan Lan. 2021. Uncertainty calibration for ensemble-based debiasing methods. In *Advances in Neural Information Processing Systems*, volume 34, pages 13657–13669. Curran Associates, Inc.

Yadollah Yaghoobzadeh, Soroush Mehri, Remi Tachet des Combes, T. J. Hazen, and

Alessandro Sordoni. 2021. Increasing robustness to spurious correlations using forgettable examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3319–3332, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.291

Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. 2023. Change is hard: A closer look at subpopulation shift. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39584–39622. PMLR.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. 2022. Improving out-of-distribution robustness via selective augmentation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 25407–25437. PMLR.

Sicheng Yu, Jing Jiang, Hao Zhang, Yulei Niu, Qianru Sun, and Lidong Bing. 2022. Interventional training for out-of-distribution natural language understanding. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11627–11638, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.799

Chen Zhang, Lei Ren, Jingang Wang, Wei Wu, and Dawei Song. 2022. Making pretrained language models good long-tailed learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3298–3312, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.217

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

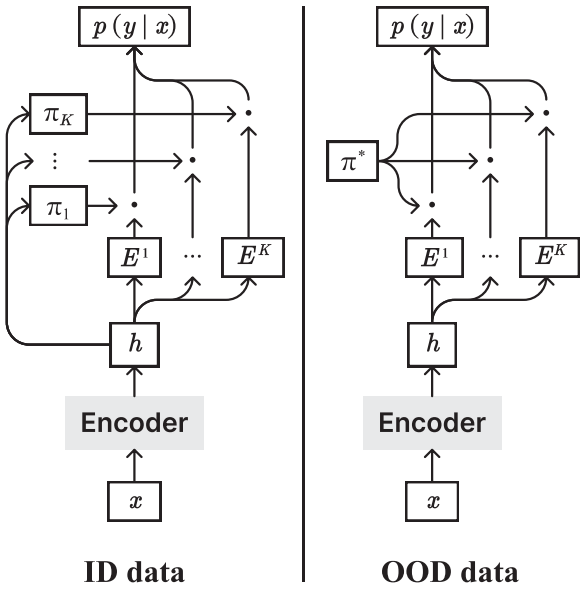


Figure 3: Overview of our method. We fit training data using a mixture model consisting of K expert networks $\{E^k\}_{k=1}^K$ and a router network π (Section 3.1). During inference, the model is used as is for ID data, and π is replaced with π^* for OOD data (Section 3.2).

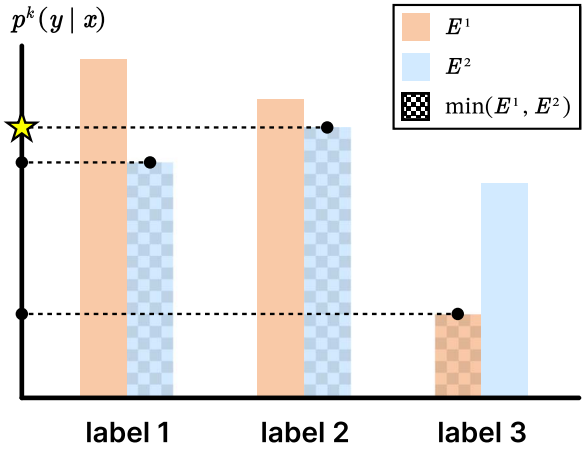


Figure 4: An example of decision-making with argmin weighting, where $K = 2$ and $|\mathcal{Y}| = 3$. After performing argmin weighting, label 2 achieves the highest score (starred) and is thus chosen as the answer.

A Additional Figures

Figure 3 shows an overview of our method (Section 3), and Figure 4 illustrates an example of our Argmin weighting (Section 3.2).

Figure 5 shows the average prediction of each expert, calculated across MNLI ID dev (Section 4.3). We observe a significant variance between experts' predictions, which indicates that different experts tend to make different predictions.

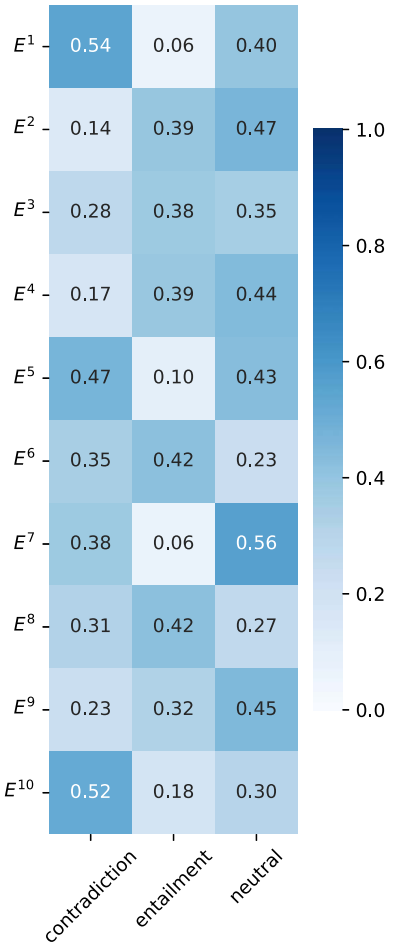


Figure 5: The average prediction of each expert E^k across MNLI ID dev.

B Further Setup Details

Table 5 specifies the URLs of the datasets, pre-trained models, and code of the previous methods we introduced in Section 4.1.

Following the fine-tuning hyperparameters of DeBERTa$_{\text{v3-large}}$ (He et al., 2023), we set the learning rate to 5e-6 and used gradient clipping with the maximum gradient norm of 1.0 in the ablation study with DeBERTa$_{\text{v3-large}}$ (Section 4.3).

Datasets	
MNLI	https://cims.nyu.edu/~sbowman/multinli/
HANS	https://github.com/tommccoy1/hans
QQP and PAWS	https://github.com/google-research-datasets/paws
FEVER and Symm. v1/v2	https://github.com/TalSchuster/FeverSymmetric
Pre-Trained Models	
BERT	https://huggingface.co/bert-base-uncased
$\text{DeBERTa}_{\text{v3-large}}$	https://huggingface.co/microsoft/deberta-v3-large
Code	
Conf-reg ♠ $^{*}_{\text{self-debias}}$	https://github.com/UKPLab/emnlp2020-debiasing-unknown
JTT*	https://github.com/YyzHarry/SubpopBench
RISK	https://github.com/CuteyThyme/RISK
EIIL	https://github.com/PluviophileYU/BAI
BAI	https://github.com/PluviophileYU/BAI
$\text{GroupDRO}^{*}_{\text{label-group}}$	https://github.com/YyzHarry/SubpopBench
ReWeightCRT*	https://github.com/YyzHarry/SubpopBench

Table 5: URLs of the datasets, pre-trained models, and code of the previous methods we used in the experiments. The methods with * needed modification on the codes to cover all the datasets we used.